2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A game theoretic approach for safe and distributed control of unmanned aerial vehicles

Allan Andre do Nascimento, Antonis Papachristodoulou, Kostas Margellos

***Abstract*— This paper presents a distributed methodology to produce collision-free control laws for an Unmanned Aerial Vehicle (UAV) fleet. We use a game theoretic framework, where UAVs accommodate for individual and fleet goals, while respecting safety requirements. The method combines Control Barrier Functions (CBFs) and a primal-dual algorithm for Nash equilibrium (NE) seeking in generalized games. Feedback is introduced by Model Predictive Control (MPC) and we analyze its stability properties. The combination of these tools allows for a distributed, collision-free pointwise equilibrium solution, despite the agents' coupling, due to common target tracking and the collision avoidance constraints. Our algorithmic results are supported theoretically and our method's efficacy is demonstrated via extensive numerical simulations.**

I. Introduction

This work addresses safe deployment of Unmanned Aerial Vehicle (UAV) fleets while taking into account the conflicting objectives of individual and common goal tracking by use of game theoretic approaches. Our method covers a wide class of multi-agent games, and designs a safe by construction, high-performance algorithm, capitalizing on the widespread applications of Control Barrier Functions (CBFs). CBFs provide a way to enforce safety through invariant set analysis [3] and have been recently applied to various problems in continuous and discrete time settings, such as adaptive cruise control [4], safe teleoperation of UAVs [18], bipedal robot navigation [1], and learning [21]. The rising number of CBF applications stem from their computational viability: when the system dynamics are affine in the input and the cost to be minimized is quadratic, a family of Quadratic Programs (QPs) parameterized by the system state needs to be solved [5], [11], [16].

Furthermore, to enhance the real-time deployment potential, and effective peer-to-peer communication, we resort to recent advancements in distributed algorithms for Nash Equilibrium (NE) computation. Distributed safety requirements produce coupled constraints, giving rise to generalized games: typically, primal-dual iterative methods are then used, in which the NE of the game is reached asymptotically. In this work we rely on a recently developed technique based on gradient tracking [8], that exhibits superior convergence properties compared to similar methodologies [6], [10]. This is a discrete-time algorithm motivated by cooperative counterparts [13]; for some conceptually similar continuous-time methods we refer to [9]. Finally, feedback is introduced by Model Predictive Control (MPC) [12], [2], which has only recently been exploited in conjunction with CBF constraints [17], [20].

Blending CBFs, MPC and the primal-dual algorithm for Nash equilibrium seeking in generalized games, that up to now have been mainly used separately, is a distinct feature of our work which inherits the benefits of these methodologies. Merging these tools allows for a distributed, collision-free equilibrium solution for the multi-UAV game, despite coupled constraints and common target, which generates an aggregative term in the cost. We propose sufficient MPC stability conditions, accounting for this extra term and illustrate the method's efficacy via numerical simulations.

This paper is organized as follows: Section II introduces the UAV system, its states and the adopted gaming setup. Section III presents the ingredients of the game for the specific multi-UAV setting. Section IV presents a distributed equilibrium seeking algorithm for the coordination problem and discusses a sufficient condition for the "aggregative MPC" stability. Section V presents a four UAV position swapping problem and a sensitivity analysis for the parameters encoding safety and prediction horizon. Section VI concludes this work and points out future research directions.

Notation. Let $\mathbb{R}, \mathbb{R}_+, \mathbb{N}$ be the set of real, non-negative real and natural numbers. $\mathscr{G} = (I,E)$ denotes an undirected graph with I being the node set and $E \subset I \times I$ the set of edges. Agent i can receive information from j if the edge $(j,i) \in E$ and *vice versa*. The set of neighbours of i is denoted by $\mathscr{N}_i := j \in I : (j,i) \in E$ and we consider that $i \in \mathscr{N}_i$. $\mathscr{W} \in \mathbb{R}^{n \times n}$ denotes the weighted adjacency matrix of graph $\mathscr{G}$, with its entries satisfying $w_{ij} = w_{ji} > 0$ if $(j,i) \in E$ and $w_{ij} = 0$ otherwise. A continuous function $f : [0,a) \to [0,\infty)$ for $a > 0$ is of class $\mathscr{K}$ if it is strictly increasing and $f(0) = 0$ [19].

II. Problem formulation

A. UAV model

Let $I = \{1,\dots,N\}$ index a fleet of N UAVs, where $p_{xi}(t), p_{yi}(t)$ denote the i^{th}-UAV's (x,y) centre of gravity coordinates at time t. Similarly, let $v_{xi}(t), v_{yi}(t), a_{xi}(t), a_{yi}(t)$ denote its velocity and acceleration components. For simplicity, we assume in this initial study that all UAVs perform level flights and share the same double integrator dynamics[1]: $\ddot{p}_{xi}(t) = a_{xi}(t)$, $\ddot{p}_{yi}(t) = a_{yi}(t)$. Discretizing with sample time

Research support by MathWorks is gratefully acknowledged; the authors are particularly grateful to Dr Rory Adams and Dr Roberto Valenti for their useful insights.

Department of Engineering Science, University of Oxford, Parks Road, Oxford OX13PJ, U.K. {allan.adn, antonis, kostas.margellos}@eng.ox.ac.uk

[1]This will be replaced by a 6-DOF nonlinear UAV model in future studies.

h leads to

$$x_i[k+1] = \begin{bmatrix} 1 & 0 & h & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x_i[k] + \begin{bmatrix} \frac{h^2}{2} & 0 \\ 0 & \frac{h^2}{2} \\ h & 0 \\ 0 & h \end{bmatrix} a_i[k], \quad (1)$$

where k denotes the time index, $x_i[k] = [p_{x_i}[k] \; p_{y_i}[k] \; v_{x_i}[k] \; v_{y_i}[k]]^T \in \mathbb{R}^4$ is the state and $a_i[k] = [a_{x_i}[k] \; a_{y_i}[k]]^T \in \mathbb{R}^2$ the control input. Let $p_d = [p^d_{x_1} \; p^d_{y_1} \dots p^d_{x_N} \; p^d_{y_N}]^T \in \mathbb{R}^{2N}$ collect each UAV target position coordinates. Using the error dynamics $e_i[k] = x_i[k] - [p^d_{x_i} \, p^d_{y_i} \, 0 \, 0]^T$, a compact depiction of (1) is

$$e_i[k+1] = A_{d_i} e_i[k] + B_{d_i} a_i[k]. \quad (2)$$

B. Game setup

In this non-cooperative multi-agent game setting, agent $i \in I$ aims at solving a finite horizon optimization problem, with horizon length H. For each $i, j \in I, i \neq j$, let $u_i = [a_i[0] \dots a_i[H-1]]_i^T$ be the decision vector of agent i over the time horizon, where $u_i \in U_i \subseteq \mathbb{R}^{2H}$ accounts for the presence of local constraints (to be defined in the sequel). Each agent $i \in I$ seeks to solve:

$$\begin{aligned} &\min_{u_i \in U_i} \; J_i(u_i, \sigma(u)) \\ &\text{subject to } A_i u_i + \sum_{j \in I \setminus \{i\}} A_j u_j \leq \sum_{i \in I} b_i \end{aligned} \quad (3)$$

where $\sigma(u) := \frac{1}{N} \sum_{i \in I} \phi_i(u_i)$ is an aggregative vector, formed by the aggregate function ϕ_i. $J_i(u_i, \sigma(u))$ is the finite horizon cost each agent $i \in I$ seeks to minimize, subject to the local constraint set U_i and coupled constraints, defined by matrices A_i and vectors b_i of appropriate dimension. The game is described by the tuple $\mathscr{G}_a = (N, \{J_i\}_{i \in I}, \{U_i\}_{i \in I}, \{A_i, b_i\}_{i \in I})$ and referred to as generalized aggregative game due to the coupling in the objective and constraints. The solution concept for such a game class is a Generalized Nash Equilibrium (GNE), summarized below.

Definition 1: A collection of strategies u^*, formed by u_i^* (optimal u_i), is a GNE for $\mathscr{G}_a$ if for all $i \in I$, we have

$$J_i(u_i^*, \sigma(u^*)) \leq \min_{u_i \in C_i(u^*_{-i})} J_i(u_i, \sigma(u^*)), \quad (4)$$

where $C_i(u^*_{-i}) = \{u_i \in U_i : \; A_i u_i + \sum_{j \in I \setminus \{i\}} A_j u_j^* \leq \sum_{i \in I} b_i\}$, and u^*_{-i} consists of $u_j^*, \forall j \in I, \, j \neq i$.

III. Ingredients of the Game

A. Coupling constraints

Following [1], for $i, j \in I$ with $i \neq j$, consider $\delta p_{i,j}[k] = [p_{x_i}[k] \, p_{y_i}[k]]^T - [p_{x_j}[k] \, p_{y_j}[k]]^T$ and a set S, denoted to be the superlevel set of a function $h_i^j : \mathscr{P} \subset \mathbb{R}^2 \to \mathbb{R}$

$$S = \{\delta p_{i,j}[k] \in \mathscr{P} \subset \mathbb{R}^n : h_i^j(\delta p_{i,j}[k]) \geq 0\}.$$

Set S is the safe set within which agents' trajectories should be confined to avoid collision. If the conditions below hold

$$\begin{aligned} &(i) \; h_i^j(\delta p_{i,j}[0]) \geq 0, \\ &(ii) \; \exists u_i[k], u_j[k] \text{ such that } \forall k \in \mathbb{N} \cup \{0\}, \\ &\quad h_i^j(\delta p_{i,j}[k+1]) - h_i^j(\delta p_{i,j}[k]) \geq -\gamma_{\text{cbf}}(h_i^j(\delta p_{i,j}[k])). \end{aligned} \quad (5)$$

then the function $h_i^j : \mathscr{P} \to \mathbb{R}$ is said to be a discrete-time exponential CBF and the set S is invariant along the trajectories of (2) governed by $u_i[k], u_j[k]$ in (5), $\forall k \in \mathbb{N} \cup \{0\}$ [1]. In (5), γ_{cbf} is a class $\mathscr{K}$ function satisfying $0 < \gamma_{\text{cbf}}(h_i^j(\delta p_{i,j}[k])) \leq h_i^j(\delta p_{i,j}[k])$ [20]; here we restrict our analysis to $0 < \gamma_{\text{cbf}} \leq 1$. CBF is preferred over a simple distance constraint, being justified in an MPC framework due to higher safety. [20] shows that a CBF with smaller γ_{cbf} produces obstacle avoidance earlier[2]. Comparatively, distance constraints demand larger prediction horizon H, not viable on real-time systems, while lower values may lead to infeasibility. Finally, CBF designs the inequality's right hand-side "adaptively", using an invariance framework, unlike distance constraints. As (3) involves affine coupling constraints, the chosen CBF is

$$\begin{aligned} h_i^j(\delta p_{i,j}[k]) &= \frac{|p_{x_i}[k] - p_{x_j}[k]|}{r_1} + \frac{|p_{y_i}[k] - p_{y_j}[k]|}{r_2} - 1 \\ &= \frac{|e_i^x[k] - e_j^x[k] + \delta p^d_{x_{i,j}}|}{r_1} + \frac{|e_i^y[k] - e_j^y[k] + \delta p^d_{y_{i,j}}|}{r_2} - 1, \end{aligned} \quad (6)$$

where $h_i^j(\delta p_{i,j}[k]) \geq 0$ guarantees no collision between UAVs i and j at time k. Parameters $r_1, r_2 > 0$ are the norm-1 unsafe radii, delimiting the UAV's body in the x and y directions. The second equality in (6) is obtained by using (2), where e_i^x, e_i^y denote the first two components of e_i, and constant target positions $p^d_{x_{i,j}}$, producing: $\delta p^d_{x_{i,j}} = p^d_{x_i} - p^d_{x_j}$ and $\delta p^d_{y_{i,j}} = p^d_{y_i} - p^d_{y_j}$. The choice of h_i^j in (6) ensures that condition (i) in (5) is convex. To ensure that (ii) in (5) is also convex we follow a linearization procedure similar to [15]:

$$\begin{aligned} &h_i^j(\delta p_{i,j}[k+1]) - h_i^j(\delta p_{i,j}[k]) \\ &\quad \geq (\nabla h_i^j(\delta p_{i,j}[k]))^T (\delta p_{i,j}[k+1] - \delta p_{i,j}[k]) \\ &\quad \geq -\gamma_{\text{cbf}} h(\delta p_{i,j}[k]), \end{aligned} \quad (7)$$

where the first inequality is due to convexity of the CBF. Satisfying (7), offers a sufficient condition for satisfaction of (ii) in (5). For all $i, j \in I$ we have:

$$\begin{aligned} \delta v_{l_{i,j}}[k] &= v_{l_i}[k] - v_{l_j}[k], \text{ for } l = x, y, \\ \Gamma_{i,j}[k] &= \frac{2}{h^2} \gamma_{\text{cbf}} h_i^j(\delta p_{i,j}[k]) \\ &\quad + \frac{2}{h} \Big(\frac{1}{r_1} \text{sgn}(\delta p_{x_{i,j}}[k]) \delta v_{x_{i,j}}[k] \\ &\quad + \frac{1}{r_2} \text{sgn}(\delta p_{y_{i,j}}[k]) \delta v_{y_{i,j}}[k] \Big). \end{aligned}$$

[2]If the system is constrained over the prediction horizon - explored in future studies.

We now rewrite relation (7) in the form of (3) resulting in

$$\begin{bmatrix}-v_c & v_c\end{bmatrix}\begin{bmatrix}u_i\\ u_j\end{bmatrix} \leq \Gamma_{i,j}[k],$$

$$v_c = \begin{bmatrix}\frac{\operatorname{sgn}(\delta p_{x_{i,j}}[k])}{r_1} & \frac{\operatorname{sgn}(\delta p_{y_{i,j}}[k])}{r_2} & 0_{1\times 2(H-1)}\end{bmatrix}. \tag{8}$$

Equation (8) imposes a safety constraint for the UAV pair i,j and similar constraints have to be instantiated in case of static obstacles.

To determine an equilibrium solution for the game under consideration we present in the next section a distributed methodology proposed in [8]. However, the latter requires that matrix A formed by the horizontal concatenation of A_i, A_j, $\forall i,j \in I, i \neq j$ must be full row rank (see Assumption IV.2 in [8]). Since we aim at avoiding constraints with every other UAV, rows of A scale as $N(N-1)/2$, whereas its columns scale as $2\times NH$. To ensure that A is full row rank, we need to ensure that $H \geq (N-1)/4$. We also assume complete knowledge by UAV i of both its own position and velocity and the corresponding information from other UAVs, (coupled in the same constraint) at each time instance.

B. Local constraints

Each agent is also subject to local constraints. These refer to $a_{i_{max}} \in \mathbb{R}^2$, $i \in I$, the limiting acceleration capacity of each UAV. The local constraint set of agent $i \in I$ is given by

$$U_i = \left\{u_i\,:\, -a_{i_{max}} \leq a_i[k] \leq a_{i_{max}},\ \text{for all } k = 0,\dots,H-1\right\}. \tag{9}$$

C. Objective functions

The objective function of each UAV $i \in I$ is expressed as

$$\begin{aligned} J_i(u_i,\sigma(u)) &= \sum_{l=0}^{H-1} e_i[l]^T Q_i e_i[l] + u_i[l]^T R_i u_i[l] \\ &+ \beta(e_i[H]^T Q_{H_i} e_i[H]) \\ &+ \frac{p_a}{N}\left(\frac{1}{N}\sum_{z=1}^{N} p_{x_z}[H] - p_{o_x}\right)^2 + \frac{p_a}{N}\left(\frac{1}{N}\sum_{z=1}^{N} p_{y_z}[H] - p_{o_y}\right)^2, \end{aligned} \tag{10}$$

where the first line of (10) is the running cost, with matrices $Q_i \succ 0$ (needed for strong monotonicity; for the stability analysis only $Q_i \succeq 0$ is necessary) and $R_i \succ 0$ of appropriate dimension, penalize the state (in error dynamics) and control input, respectively. The second line of (10) is a terminal cost with weighting matrix $Q_{H_i} \succ 0$ and a scaling factor $\beta > 0$. The last line of (10) penalizes the agents' aggregate (this implicitly defines functions ϕ_i and σ in (3)) with respect to a common fixed target $p_o = (p_{o_x}, p_{o_y})^T$, to be tracked by all UAVs. Here we require the aggregate to be close to the target at the terminal time H. The term $\frac{p_a}{N} > 0$ is introduced to ensure that each J_i gets exactly $(1/N)$-th of the aggregative term and we account (by scaling p_a) for the relative importance of the individual and common objectives. To relate terms in (10) with $u_i[k]$, we use Eq. (2) recursively. This objective satisfies strong monotonicity and Lipschitz continuity with respect to agents' decisions (e.g., see standing Assumptions II.2 and II.3 in [8]).

IV. Solution methodology and analysis

A. Primal-Dual TRADES algorithm

We assume that each agent knows its own information and that it can exchange information with its neighbours $\mathcal{N}_i$. We use a similar communication structure as in [8], which relies on undirected and strongly connected graphs.

The Primal-Dual TRADES (Algorithm 1) [8] solves (3) and uses variables to track the coupling constraint and the aggregative term with the goal of solving the problem in a distributed manner.

Algorithm 1 Primal-Dual TRADES (Agent i)

1: **Initialization:** $u_i^0 \in U_i$, $\lambda_i^0 \in \mathbb{R}_+^m$, $z_i^0 = 0$, $y_i^0 = 0$
2: **Repeat until convergence**
3: $u_i^{p+1} = u_i^p + \delta_{tr}([u_i^p - \gamma_{tr}\tilde{F}_i(u_i^p, \phi_i(u_i^p) + z_i^p) - \gamma_{tr} G_{u,i}(N(A_i u_i^p - b_i) + y_i^p, \lambda_i^p] - u_i^p)$
4: $\lambda_i^{p+1} = \sum_{j\in\mathcal{N}_i} w_{ij}\lambda_j^p + \delta_{tr}\gamma_{tr} G_{\lambda,i}(N(A_i u_i^p - b_i) + y_i^p, \lambda_i^p)$
5: $z_i^{p+1} = \sum_{j\in\mathcal{N}_i} w_{ij} z_j^p + \sum_{j\in\mathcal{N}_i} w_{ij}\phi_j(u_j^p) - \phi_i(u_i^p)$
6: $y_i^{p+1} = \sum_{j\in\mathcal{N}_i} w_{ij} y_j^p + \sum_{j\in\mathcal{N}_i} w_{ij} N(A_j u_j^p - b_j) - N(A_i u_i^p - b_i)$
7: **return** u_i^*

In Algorithm 1, line 1 initializes the decision variables u_i, the Lagrange multiplier λ_i, and tracking variables z_i and y_i for the cost J_i and the coupled constraints in (3). The superscript $p \in \mathbb{N}\cup\{0\}$ denotes the iteration index. Line 2 updates the variables until convergence, in line 3 $\tilde{F}_i$ and $G_{u,i}$ are the pseudo gradient of the cost and constraints in (3), with respect to the primal variable, i.e.,

$$\tilde{F}_i(x_i,s) = \nabla_1 J_i(x_i,s) + \frac{\nabla\phi_i(x_i)}{N}\nabla_2 J_i(x_i,s), \tag{11}$$

$$G_{u,i}(s_1,s_2) = \sum_{l=1}^{m} \max\{\rho[s_1]_l + [s_2]_l, 0\}[A_i]_l^T, \tag{12}$$

where $\nabla_1 J_i$ and $\nabla_2 J_i$ are the gradients of J_i with respect to its first and second arguments. $[A_i]_l$ and $[b_i]_l$ indicate the l-th row of matrix A_i and vector b_i in (3), and the constant $\rho > \frac{\delta_{tr}\gamma_{tr}}{w_{ii}} > 0$ (see (33) in [8]). The argument within brackets in line 3 acts as a gradient descent method with $\gamma_{tr} \in \mathbb{R}$ as the step size. Parameter $\delta_{tr} \in (0,1)$ performs a convex combination between the current value and the new estimate. Line 4 updates the dual variable also using gradient descent, with $G_{\lambda,i}$ (13) as the pseudo gradient of the constraints in (3) with respect to the dual variable, i.e.,

$$G_{\lambda,i}(s_1,s_2) = \frac{1}{\rho}\sum_{l=1}^{m}(\max\{\rho[s_1]_l + [s_2]_l, 0\} - [s_2]_l)e_{b_l}, \tag{13}$$

Lines 5 and 6 have update laws for the aggregative cost and the coupling constraint tracking variables. Parameter w_{ij} is the i,j element of the network adjacency matrix, introduced for consensus among agents estimations. Note that we do not enforce local constraints in a hard manner in Algorithm 1; this stems from the fact that for generalized games the algorithm in [8] does not allow for local constraints. The control input generated by TRADES will be saturated to ensure it is within the desired limits (see Algorithm 2); numerical evidence shows that convergence is achieved even if local constraints are enforced in a hard manner.

The presented setup satisfies all technical assumptions introduced in [8], which are omitted here in the interest of space. The subsequent theorem summarizes the convergence behaviour of Algorithm 1.

Theorem 1: [8] Consider Algorithm 1, with the tracking variables z_i^0 and y_i^0 being initialized as zero. Then, Algorithm 1 converges linearly to a NE of the game $\mathscr{G}_a$.

B. Safe aggregative receding horizon control

We now modify Algorithm 1 by adopting MPC to produce the enhanced safe distributed Algorithm 2.

Algorithm 2 Safe distributed receding horizon TRADES (Agent i)

1: UAV $i \leftarrow p_d, p_o, \forall i \in I$
2: $e_i[0] = x_i[0] - (p^d_{x_i}, p^d_{y_i}, 0, 0)^T,\ e_i[1] = e_i[0]$
3: $u_i[0] = 0, \forall i \in I$
4: **for** $k = 1, 2, \dots$ **do**
5: UAV $i \leftarrow e_j[k], u_j[k-1], \forall j \in I,\ j \neq i$
6: Calculate $A_i u_i + \sum_{j \in I \setminus \{i\}} A_j u_j \leq \sum_{i \in I} b_i$ as in (8)
7: Initialize **Algorithm 1**: $u_i^0 = u_i[k-1] \in U_i$, $\lambda_i^0 \in \mathbb{R}^m_+$, $z_i^0 = 0_2$, $y_i^0 = 0_m$.
8: Run **Algorithm 1**
9: $u_i^* = \max(\min(u_i^*, a_{max}), -a_{max})$
10: $u_i[k] = u_i^*[0]$
11: $e_i[k+1] = A_{d_i} e_i[k] + B_{d_i} a_i[k]$
12: UAV $j \leftarrow e_i[k+1],\ u_i[k], \forall j \in I, j \neq i$
13: **end for**

Algorithm 2 runs in parallel for all agents. Lines 1-3 collect relevant initialization information for UAV i such as the initial and desired end state for all UAVs and the common point to be tracked. Line 6 updates at each iteration the collision avoidance constraints, line 7 initializes the Primal-Dual trades with the optimal u_i from the previous iteration. Lines 8-11 involve the main updates. At each iteration Algorithm 1 (line 8) returns the optimal decision vector u_i^*, which is saturated (line 9), so that $u_i \in U_i$. Then, u_i^* has its 1^{st} component parsed to $u_i[k]$ (line 10), and acts in the system dynamics (line 11). Both $u_i[k]$, and $e_i[k+1]$ must be shared with all other UAVs for the coupled constraints' calculation in the next iteration of Algorithm 2.

C. Aggregative MPC stability discussion

Consider the MPC formulation:

$$\begin{aligned} J_i^*(u_i, \sigma(u)) = \min_{u_i} \quad & J_i(u_i, \sigma(u)) \\ \text{s.t. } & e_i[l+1] = A_{d_i} e_i[l] + B_{d_i} a_i[l] \\ & u_i[l+p] \in U_i,\ p = 0 \dots H-1 \\ & e_i[l+H] \in \mathscr{E}_{iH} \\ & e_i[l] = e_0. \end{aligned} \tag{14}$$

In standard MPC (without $\sigma(u)$), (14) is solved repeatedly whenever a new e_0 is available [7], [20]. However, since $J_i(u_i, \sigma(u))$ in (14) contains a quadratic cost on $\sigma(u)$, a re-evaluation of the MPC's stability condition is necessary. Excluding the CBF, we analyze the aggregative term's effect in terms of stability following [14]; however, our analysis is distinct due to the presence of that term. This is our main theoretical result.

Proposition 1: For all $i \in I$, consider the terminal set $\mathscr{E}_{H_i} = \{e_i \in \mathbb{R}^n : e_i^T Q_{H_i} e_i \leq \alpha_{H_i}\}$, where $\alpha_{H_i} > 0$ (non-degenerate as $Q_{H_i} \succ 0$). Assume there exists $\bar{k}$ such that $\forall k > \bar{k}$, $\forall i \in I$, $e_i[k] \in \mathscr{E}_{H_i}$. Furthermore, $\forall i \in I$, $\forall e_i[k] \in \mathscr{E}_{H_i}$, we assume the following conditions:

1) $A_{d_i} e_i[k] + B_{d_i} a_i[k] \in \mathscr{E}_{H_i}$
2) $u_i[k] \in U_i$
3) $\beta((e_i[k+1])^T Q_{H_i}(e_i[k+1]) - (e_i[k])^T Q_{H_i}(e_i[k])) + \frac{p_a}{N}(g^T[k+1]g[k+1] - g^{*^T}[k]g^*[k]) - e_i[k-H]^T Q_i e_i[k-H]$ $\leq -(e_i[k])^T Q_i (e_i[k]) - (a_i[k])^T R_i (a_i[k])$.

where

$$\begin{aligned} g[k+H] &= C\begin{bmatrix} A_d^{H-1} B_d & 0_{4\times n_p}, \dots, B_d & 0_{4\times n_p} \end{bmatrix} u_i \\ &+ z_i[k] + \frac{1}{N}\sum_{i=1}^{N}\left(C A_d^H e_i[k] + \begin{bmatrix} p^d_{x_i} \\ p^d_{y_i} \end{bmatrix} \right) - p_o, \\ C &= \begin{bmatrix} I_{2\times 2} & 0_{2\times 2} \end{bmatrix}. \end{aligned}$$

Then, $\forall i \in I$ the MPC (14) with cost J_i (10) and state $e_i[k]$ is stable.

Proof: Let $J_i(k)$ and $J_i^*(k)$ be a cost and the optimal cost of the i^{th} agent at time k as in (10), with input sequence $u_i^*[k] = [a_i^*[k] \dots a_i^*[k+H-1]]_i^T$. A sub-optimal cost of $J_i(k+1)$ can be obtained using $u_i[k+1] = [a_i^*[k+1] \dots a_i^*[k+H-1]\, a_i[k+H]]_i^T$. Due to the MPC formulation (14), we assume there exists a $\bar{k}$ such that $\forall k > \bar{k}, \forall i \in I, e_i[k+H] \in \mathscr{E}_{H_i}$. Using the stated conditions in Proposition 1 we conclude: Due to condition 2) control input constraints are satisfied, while by condition 1), $e_i[k+H+1] \in \mathscr{E}_{H_i}$ [14]. The cost of the sub-optimal sequence in $k+1$ is:

$$\begin{aligned} J_i(k+1) = &\sum_{l=k+1}^{k+H-1} e_i^{*^T}[l] Q_i e_i^*[l] + a_i^{*^T}[l] R_i a_i^*[l] \\ &+ e_i[k+H]^T Q_i e_i[k+H] + a_i[k+H]^T R_i a_i[k+H] \\ &+ \beta(e_i[k+H+1]^T Q_{H_i} e_i[k+H+1]) \\ &+ \frac{p_a}{N}(g[k+H+1]^T g[k+H+1]). \end{aligned}$$

Adding and subtracting $e_i[k]^T Q_i e_i[k]$, $a_i^*[k]^T R_i a_i^*[k]$, $\beta(e_i^*[k+H]^T Q_{H_i} e_i^*[k+H])$, $\frac{p_a}{N}(g_i^*[k+H]^T g_i^*[k+H])$ we get

$$\begin{aligned} J_i(k+1) = &J_i^*(k) - e_i[k]^T Q_i e_i[k] - a_i^{*^T}[k] R_i a_i^*[k] \\ &- \beta(e_i^*[k+H]^T Q_{H_i} e_i^*[k+H]) \\ &- \frac{p_a}{N} g_i^*[k+H]^T g_i^*[k+H] + e_i[k+H]^T Q_i e_i[k+H] \\ &+ a_i[k+H]^T R_i a_i[k+H] \\ &+ \beta(e_i[k+H+1]^T Q_{H_i} e_i[k+H+1]) \\ &+ \frac{p_a}{N}(g[k+H+1]^T g[k+H+1]). \end{aligned}$$

Now using condition 3):

$$\beta(e_i[k+H+1]^T Q_{H_i} e_i[k+H+1] - e_i^{*^T}[k+H] Q_{H_i} e_i^{*^T}[k+H]) + e_i[k+H]^T Q_i e_i[k+H] + a_i[k+H]^T R_i a_i[k+H] + \frac{p_a}{N}(g[k+H+1]^T g[k+H+1] - g_i^{*^T}[k+H] g_i^*[k+H]) - e_i[k]^T Q_i e_i[k] \leq 0,$$

resulting in: $J_i^*(k+1) \leq J_i(k+1) \leq J_i^*(k) - a_i^{*^T}[k] R_i a_i^*[k]$. The optimal cost iterates form a non-increasing sequence which is bounded below from zero as $R_i \succ 0$, $\forall i \in I$. As a result, the cost iterates form a convergent sequence, tending to some point $e_{i_{eq}} = [p_{x_{i_{eq}}} \; p_{y_{i_{eq}}} \; 0 \; 0]^T$, $i \in I$ where the acceleration inputs vanish and thus, stability is proven. The presence of the aggregative term prevents cost iterates from necessarily converging to zero, ensuring only stability. However, when the aggregative objective is the average of individual agents' target locations p^d, the cost exhibits a decoupled structure since $\sigma(u) = 0$ in (3), guaranteeing asymptotic stability. ■

Assumption 3 in Proposition 1 is a non-increase condition; future work concentrates towards establishing sufficient conditions for its satisfaction.

V. Numerical Example

We study a four UAVs position swapping scenario, described by a fully connected communication graph, in which agents start at $p_{0_1} = [0, 1]^T$, $p_{0_2} = [0, -1]^T$, $p_{0_3} = [1, 0]^T$ and $p_{0_4} = [-1, 0]^T$, and aim for positions $p_{d_1} = p_{0_2}$, $p_{d_2} = p_{0_1}$, $p_{d_3} = p_{0_4}$ and $p_{d_4} = p_{0_3}$. Average positions should be kept near the recharging station, located at $p_o = [0,0]^T$ ($p_a = 1$) and $p_o = [0.5, 0.5]^T$ ($p_a = 10^4$) in the 1st and 2nd runs at the predicted horizon's end. The sample time is $h = 0.2\,s$, and accelerations are limited to $2\,m/s^2$ in both x and y directions. Simulations use the following parameters for all UAVs: $H = 3$, $Q_i = \text{diag}(5,5,5,5)$, $R_i = \text{diag}(2,2,2,2)$, and Q_{H_i} is the solution of the algebraic Riccati equation of the associated unconstrained infinite horizon quadratic objective function J_i, without the aggregative term. $\beta = 1$, $\gamma_{\text{cbf}} = 0.1$, $\gamma_{\text{tr}} = \delta_{\text{tr}} = 0.1$, and $r_1 = 0.25m$ and $r_2 = 0.5m$ were arbitrarily chosen. The stopping criterion for Algorithm 1 is $\max(||u_i^{p+1} - u_i^p||, ||\lambda_i^{p+1} - \lambda_i^p||) \leq \Delta$, $\forall i \in I$ for a number of consecutive iterations equal to the graph diameter, where $\Delta > 0$ is a chosen tolerance.

In Figure 1, triangles represent UAVs, parallelograms show the CBF boundary, 'o' and 'x' symbols depict start and target locations, and the square shows the common target/aggregate position. UAVs (shown at distinct times) avoid collisions by respecting unsafe boundaries and reach targets while limited by input constraints. The TRADES algorithm requires, when approaching the safety boundaries, a maximum of 112 iterations to solve the finite horizon problem with a numerical tolerance of 3.5×10^{-2}. With these parameters, up to 98.7% of Algorithm 2 iterations can run within $h = 0.2\,s$ (excluding communication overhead).

The parameter γ_{cbf} captures the maneuver's aggressiveness when agents are close to collision (e.g., see [20]). A smaller γ_{cbf} value results in a more cautious controller, while a higher

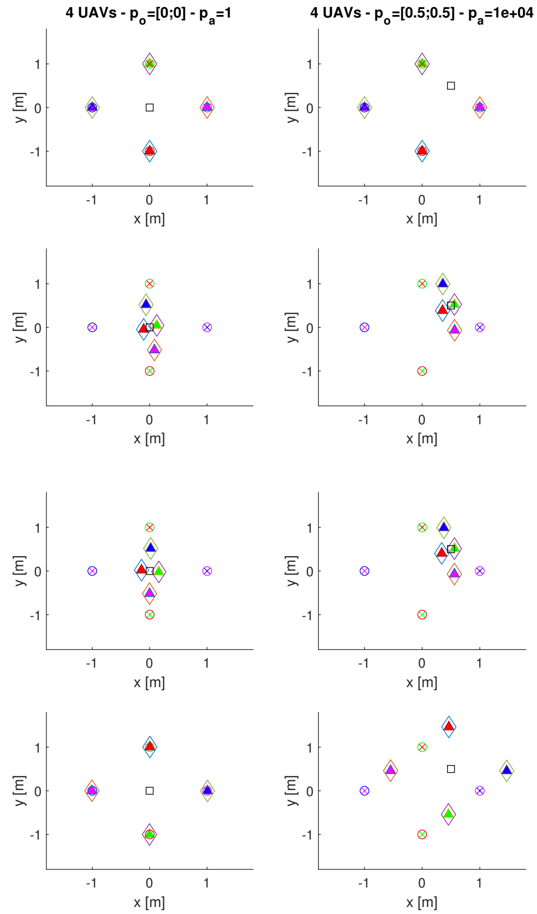

Fig. 1. Panels from top to bottom: UAVs at starting positions, coming close to neighbours' unsafe boundary, after collision is resolved, and at swapped positions in the end. Left column panels correspond to $p_o = [0,0]^T$ and $p_a = 1$, while the right one to $p_o = [0.5, 0.5]^T$ and $p_a = 10^4$.

value makes agents more aggressive, potentially causing feasibility issues. This is also noticed for small values of r_1 and r_2. In both feasibility issue cases, the UAVs see only the next step in terms of safety due to the CBFs in (8). This "allows" high accelerations, building up UAVs speed as they move towards its goals. Upon a sudden obstacle detection, UAVs have one instant to correct their path. Their high flying speeds demand strong deceleration, which is limited by the actuator capacity leading to infeasibility.

Figure 2 illustrates CBF value evolution for different γ_{cbf} values. Parameters used are: $h = 0.15\,\text{s}$, $\beta = 0.1$, $\Delta = 1 \times 10^{-3}$, and $a_{\max} = 4\,\text{m/s}^2$. Higher γ_{cbf} values reduce inter UAV CBF value compared to lower ones. The green curve with the highest γ_{cbf}, for instance, has the smallest minimum among all other curves, the aggressiveness can result in worse performance. Further predicted states of MPC in (5) and decreasing radii values r_1 and r_2 as more CBF constraints are introduced in (5) could address performance issues.

Under some circumstances, $H > 1$ enhance the performance of Algorithm 2 if compared to using $H = 1$, which acts as a DCLF-DCBF [20]. One case is when all other parameters remain the same and β is decreased. Another

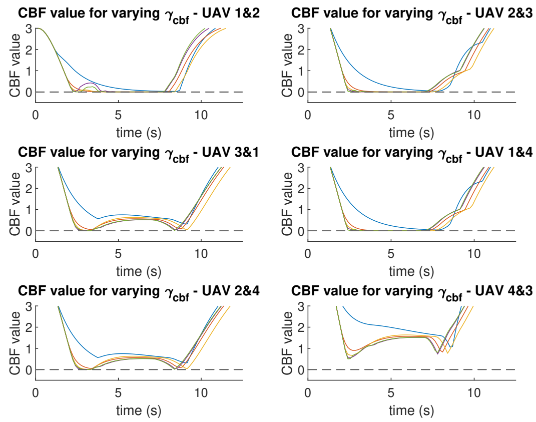

Fig. 2. CBF values between UAVs pairs for γ_{cbf} values: 0.1 (blue), 0.3 (red), 0.5 (yellow), 0.7 (purple), and 0.9 (green). Values above the dashed black line indicate safety.

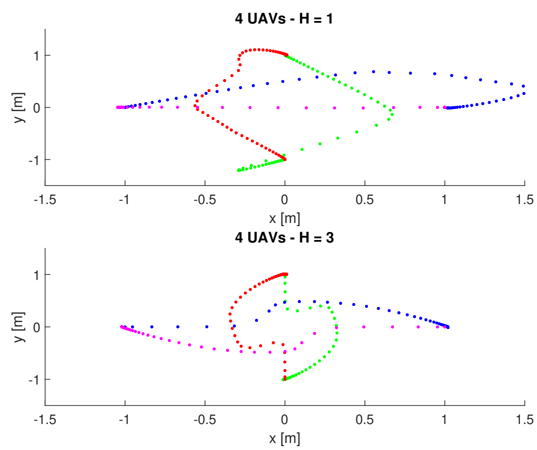

Fig. 3. Effect of distinct values of H in UAVs' trajectories.

situation is when we increase Δ, allowing Algorithm 1 within Algorithm 2 to terminate earlier (intensified if the UAVs have small control authority). This is a relevant case for real-time deployment as early termination of Algorithm 1 is certainly desired. A value $H > 1$ in such cases manage to avoid infeasibility, or to improve performance. Figure 3 illustrates this situation with $H = 1$ and $H = 3$, $\gamma_{\text{cbf}} = 0.5$ and convergence tolerance $\Delta = 5.5 \times 10^{-2}$. A value of $H = 1$ makes UAVs oscillate considerably across their trajectory to the target and overshoot around their final destinations, while $H = 3$ results in a smooth trajectory and arrival for UAVs.

VI. Concluding Remarks and Future Work

In this work, we proposed a distributed equilibrium seeking algorithm for multi-UAV control with safety guarantees using CBFs. A primal-dual mechanism determining the NE strategy in a distributed manner uses MPC to enhance its performance which is displayed by a multi-UAV position swapping problem. We studied the effect of CBF parameters and prediction horizon on the fleet performance. Future work includes exploring terminal cost/set pairs for stability, safety, and recursive feasibility and comparison to relevant methods.

References

[1] Ayush Agrawal and Koushil Sreenath. Discrete Control Barrier Functions for safety-critical control of discrete systems with application to bipedal robot navigation. In *Robotics: Science and Systems*, volume 13. Cambridge, MA, USA, 2017.

[2] Frank Allgöwer and Alex Zheng. *Nonlinear model predictive control*, volume 26. Birkhäuser, 2012.

[3] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *2019 18th European control conference (ECC)*, pages 3420–3431. IEEE, 2019.

[4] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE Conference on Decision and Control*, pages 6271–6278. IEEE, 2014.

[5] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[6] Giuseppe Belgioioso, Angelia Nedić, and Sergio Grammatico. Distributed generalized nash equilibrium seeking in aggregative games on time-varying networks. *IEEE Transactions on Automatic Control*, 66(5):2061–2075, 2020.

[7] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[8] Guido Carnevale, Filippo Fabiani, Filiberto Fele, Kostas Margellos, and Giuseppe Notarstefano. Tracking-based distributed equilibrium seeking for aggregative games. *arXiv preprint arXiv:2210.14547*, 2022.

[9] Claudio De Persis and Sergio Grammatico. Continuous-time integral dynamics for a class of aggregative games with coupling constraints. *IEEE Transactions on Automatic Control*, 65(5):2171–2176, 2019.

[10] Dian Gadjov and Lacra Pavel. Single-timescale distributed GNE seeking for aggregative games over networks via forward–backward operator splitting. *IEEE Transactions on Automatic Control*, 66(7):3259–3266, 2020.

[11] Kevin Galloway, Koushil Sreenath, Aaron D Ames, and Jessy W Grizzle. Torque saturation in bipedal robotic walking through control Lyapunov function-based quadratic programs. *IEEE Access*, 3:323–332, 2015.

[12] Basil Kouvaritakis and Mark Cannon. Model predictive control. *Switzerland: Springer International Publishing*, 38, 2016.

[13] Xiuxian Li, Lihua Xie, and Yiguang Hong. Distributed aggregative optimization over multi-agent networks. *IEEE Transactions on Automatic Control*, 67(6):3165–3171, 2021.

[14] Johan Löfberg. *Linear model predictive control: Stability and robustness*. PhD thesis, Linköping University Electronic Press, 2001.

[15] VP Varun, Abraham P Vinod, and Shishir Kolathaya. Motion planning with dynamic obstacles using convexified control barrier functions. In *2021 Seventh Indian Control Conference (ICC)*, pages 81–86. IEEE, 2021.

[16] Han Wang, Antonis Papachristodoulou, and Kostas Margellos. Distributed control design and safety verification for multi-agent systems. *arXiv preprint arXiv:2303.12610*, 2023.

[17] Zhe Wu and Panagiotis D Christofides. Control lyapunov-barrier function-based predictive control of nonlinear processes using machine learning modeling. *Computers & Chemical Engineering*, 134:106706, 2020.

[18] Bin Xu and Koushil Sreenath. Safe teleoperation of dynamic UAVs through control barrier functions. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7848–7855. IEEE, 2018.

[19] Xiangru Xu, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54–61, 2015.

[20] Jun Zeng, Bike Zhang, and Koushil Sreenath. Safety-critical model predictive control with discrete-time Control Barrier Function. In *2021 American Control Conference (ACC)*, pages 3882–3889. IEEE, 2021.

[21] Liqun Zhao, Konstantinos Gatsis, and Antonis Papachristodoulou. A barrier-lyapunov actor-critic reinforcement learning approach for safe and stable control. *arXiv preprint arXiv:2304.04066*, 2023.