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 safe by construction methodology to produce collision-free control laws for a population of Unmanned Aerial Vehicles (UAVs) that can be computed in a distributed manner. We view this problem under a game theoretic lens, with UAVs seeking to reach individual locations, while being coupled by safety requirements to avoid collision as well as due to the presence of an aggregate objective encoding the population's common goal to keep track of a common target. To achieve this, we bring together Control Barrier Functions (CBFs) and a recently developed primal-dual algorithm for Nash equilibrium seeking in generalized games. Moreover, we introduce feedback by means of a receding horizon Model Predictive Control (MPC) strategy and analyze this in terms of its stability properties. The combination of these tools allows for a distributed, collision-free equilibrium solution, despite agents being coupled due to the common target they seek to track and the collision avoidance constraints. Our algorithmic results are supported theoretically and their efficacy is demonstrated via extensive numerical simulations, illustrating convergence of the proposed scheme to a safe solution, while meeting the UAVs' individual objectives.**

I. INTRODUCTION

The ongoing embedded systems reduction in cost and increase in capacity has sparked a huge interest in deploying teams of robots in real world scenarios. Such endeavour has opened up new directions in the area of Unmanned Aerial Vehicles (UAVs). However, there are still several challenges to be addressed to facilitate real time implementation, requiring optimal and at the same time safe operation in dynamic environments. Motivated by this, we concentrate on constructing collision free control laws for a population of Unmanned Aerial Vehicles (UAVs). We view this problem under a game theoretic lens, with UAVs/agents seeking to reach individual locations, while being coupled by safety requirements to avoid collision (which induces safety constraints) as well due to the presence of an aggregate objective encoding the population's common goal to keep track of a common target. These requirements capture a wide class of multi-agent games subject to safety requirements, that go beyond the UAV case study under consideration. At the same time, conflicting objectives and information privacy concerns prevent UAVs from sharing information considered as private, thus calling for distributed equilibrium seeking solutions.

To address these challenges we aim at designing a safe by construction, high-performance and distributed algorithm, to reach Nash equilibria in multi-UAV games. To achieve this we capitalize on the widespread applications of Control Barrier Functions (CBFs), that provide a theoretically sound framework to enforce a system to remain inside a forward invariant set, offering a safe by construction design [3]. In particular, CBFs have recently been applied to various problems both in a continuous and discrete time setting, ranging from adaptive cruise control [4] to safe teleoperation of UAVs [18], and bipedal robot navigation, with the construction of discrete CBFs provided in [1]. The increasing number of CBF applications stem from their computational viability, as in the case where the system dynamics are affine in the input and the cost criterion to be minimized is quadratic, e.g., see [12], [5], [13], which involves solving a family of Quadratic Programs (QPs) parameterized by the system state.

To allow for distributed solution strategies thus enhancing the potential for real-time deployment, while accounting for peer-to-peer communication, we resort to recent advancements in distributed algorithms for Nash equilibrium computation. Hard safety requirements couple the UAVs in the constraints thus giving rise to the so called generalized games. To deal with such games, typically primal-dual iterative methodologies are employed so that the underlying Nash equilibrium of the underlying game is reached asymptotically. In this work we rely on a recently developed technique based on gradient tracking [9], that exhibits superior convergence properties compared to algorithmically similar methodologies [6], [11]. This is a discrete-time algorithm motivated by cooperative counterparts [15]; for some conceptually similar continuous-time methods we refer to [10]. Moreover, to boost performance and introduce feedback we suggest a receding horizon implementation based on Model Predictive Control (MPC) [14], [8], [2]. It should be noted that MPC has only recently been exploited in conjunction with CBF constraints [20].

Blending CBFs, MPC and the recently developed primal-dual algorithm for Nash equilibrium seeking in generalized games is a distinct feature of our work, as this allows inheriting benefits from each of these methodologies that up to now have been mainly treated separately. The combination of these tools allow for a distributed, collision-free equilibrium solution for the multi-UAV game, despite agents being coupled due to a common target they seek to track and the collision avoidance constraints. Moreover, we propose a sufficient condition for stability of the MPC problem accounting for the presence of aggregative terms due to the agents' coupling. Our algorithmic results are supported theoretically and their efficacy is demonstrated via

Research support by MathWorks is gratefully acknowledged.

All authors are with the Department of Engineering Science, University of Oxford, Parks Road, Oxford OX13PJ, United Kingdom. {allan.andredonascimento, antonis, kostas.margellos}@eng.ox.ac.uk

Manuscript 517 submitted to 2023 62nd IEEE Conference on Decision and Control (CDC) . Received March 29, 2023.

extensive numerical simulations, illustrating convergence of the proposed scheme to a safe solution, while meeting the UAVs' individual objectives.

The rest of this paper is organized as follows: Section II introduces the UAV system under consideration and states 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 UAV team coordination problem and discusses a sufficient condition for stability of the constructed MPC controller. Section V presents a detailed numerical study for a four UAV position swapping problem, including a sensitivity analysis for the tunable parameters encoding safety. Finally, Section VI draws conclusions on the presented method and points out future research directions.

Notation. Let $\mathbb{R}, \mathbb{R}_+, \mathbb{N}$ denote the set of real, non-negative real and natural numbers respectively. $\mathscr{G} = (I, E)$ 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 and vice versa if the edge $(j,i) \in E$. The set of neighbors of i is denoted by $\mathscr{N}_i := j \in I : (j,i) \in E$ and we consider that $i \in \mathscr{N}_i$. $\mathscr{W} \in \mathbb{R}^{n \times n}$ is used to denote 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 some $a > 0$ is said to belong to class $\mathscr{K}$ if it is strictly increasing and $f(0) = 0$ [19].

II. Problem formulation

A. UAV model

Consider a set $I = \{1,\dots,N\}$ indexing a fleet of N UAVs/agents. For each $i \in I$, let $p_{xi}(t), p_{yi}(t)$, denote the $x-y$ position coordinates of the centre of gravity of UAV i at time t. Similarly, let $v_{xi}(t), v_{yi}(t), a_{xi}(t), a_{yi}(t)$ denote its velocity and acceleration components, respectively. All UAVs are assumed to be described by the same dynamics, performing level flights thus neglecting for the purpose of this analysis their vertical motion. Thus, the dynamics of each UAV can be described by the simplified 2-DOF double integrator dynamics as

$$\ddot{p}_{xi}(t) = a_{xi}(t),$$
$$\ddot{p}_{yi}(t) = a_{yi}(t).$$

Discretizing these dynamics with a sample time h and using a zero-order-hold, we obtain the following discrete time double integrator dynamics,

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

where k denotes time index, $x_i[k] = (p_{x_i}[k], p_{y_i}[k], v_{x_i}[k], v_{y_i}[k])^T \in \mathbb{R}^4$ denotes the state and $a_i[k] = (a_{x_i}[k], a_{y_i}[k])^T \in \mathbb{R}^2$ the control input of the discrete time system. Let also the vector $p_d = (p^d_{x_1}, p^d_{y_1}, \dots, p^d_{x_N}, p^d_{y_N})^T \in \mathbb{R}^N$ contain the target position coordinates for each UAV. Considering the error dynamics $e_i[k] = x_i[k] - (p^d_{x_i}, p^d_{y_i}, 0, 0)^T$, we represent (1) in compact form as

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

B. Game setup

We consider a non-cooperative setting with the UAVs acting as players in a multi-agent game. Agent $i \in I$ aims at solving a finite horizon optimization problem, with horizon length H. To this end, for each $i, j \in I, i \neq j$, let $u_i = (u_i[0], \dots, u_i[H-1])^T$ be the decision vector of agent i over the time horizon, where $u_i[k]$ includes the acceleration inputs $a_i[k]$ we seek to determine, as well as additional auxiliary variables $t_{i,j}[k]$ to be defined in the sequel.

Given the decision variables of all other agents, each agent $i \in I$ seek to solve the following problem.

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

where $\sigma(u) := \frac{1}{N} \sum_{i \in I} \phi_i(u_i)$ is an aggregative vector, and ϕ_i is a mapping encoding the contribution of variable u_i to the aggregate. For each $i \in I$, $J_i(u_i, \sigma(u))$ is the finite horizon cost agent i seeks to minimize, given the aggregate $\sigma(u)$, and U_i is a local constraint set. Matrices A_i and vectors b_i are of appropriate dimension and dictate affine constraints coupling agents decisions. Their particular representation is to facilitate the distributed algorithm presented in the sequel.

The game is thus described by the tuple $\mathscr{G}_a = (N, \{J_i\}_{i \in I}, \{U_i\}_{i \in I}, \{A_i, b_i\}_{i \in I})$. Games of this form where agents' decisions are coupled both in the objective through the agents aggregate σ, and in the constraints, are referred to as generalized aggregative games. We adopt the generalized Nash equilibrium as the solution concept for such games. This is summarized in the following definition.

Definition 1: A collection of strategies u^* is a generalized Nash equilibrium 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} denotes the decision variables of all agents but for the i-th one.

Prior to demonstrating how the distributed algorithm of [9] can be employed to reach a Nash equilibrium of such game as per Definition 1, we define in the next section each element of the tuple characterizing our game for the specific UAV set-up.

III. Ingredients of the game

A. Coupling constraints

In this subsection we discuss how A_i and b_i, $i \in I$, that give rise to the coupling constraints in (3) are constructed for our multi-UAV game. To this end, we show how control barrier functions can be used to avoid collisions that in turn give rise to such coupling constraints. We consider a more strict definition of the CBF in the sense that we demand exponential convergence of the CBF.

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 $h_i^j : \mathscr{P} \subset \mathbb{R}^2 \to \mathbb{R}$

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

The set S will play the role of the safe set within which agents' trajectories should be confined to avoid collision. If the conditions below hold

(i) $h_i^j(\delta p_{i,j}[0]) \geq 0$,
(ii) $\exists u_i[k], u_j[k]$ such that $\forall k \in \mathbb{N} \cup \{0\}$,

$$h_i^j(\delta p_{i,j}[k+1]) - h_i^j(\delta p_{i,j}[k]) \geq -\gamma_{\text{cbf}} h_i^j(\delta p_{i,j}[k]). \tag{5}$$

then the function $h_i^j : \mathscr{P} \to \mathbb{R}$ is said to be a discrete-time exponential control barrier function and the set S is invariant along the trajectories of the discrete time control system governed by $u_i[k], u_j[k]$ in (5), $\forall k \in \mathbb{N} \cup \{0\}$ [1]. In (5), γ_{cbf} should be 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]; in this work we restrict attention at designing controllers satisfying $0 < \gamma_{\text{cbf}} \leq 1$.

Motivated in part by the fact that (3) involves affine coupling constraints so that it is amenable to the distributed methodology that will be presented in the sequel, we make the following choice for the CBF function

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

where $h_i^j(\delta p_{i,j}[k]) \geq 0$ implies that there is no collision between UAVs i and j at the time k. Parameters r_1 and r_2 are positive, defining the first norm safety radii in the x and y direction, respectively. Notice that the second equality follows from the first one by considering the error dynamics in (2) with e_i^x, e_i^y denoting the first two components of e_i, and the fact that the desired target position is constant at each step. We also set $\delta p_{x_{i,j}}^d = p_{x_i}^d - p_{x_j}^d$ and $\delta p_{y_{i,j}}^d = p_{y_i}^d - p_{y_j}^d$.

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 [17]. In fact, we have that

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

where the first inequality is due to convexity of the CBF. Notice that satisfying (7), offers a sufficient condition for satisfaction of (ii) in (5).

For any $i, j \in I$, set $t_{i,j}[k] = \Gamma_{i,j}[k]$, where

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

Note that $t_{i,j}$ are auxiliary decision variables that are introduced to ensure that the matrix formed by the horizontal concatenation of A_i, A_j, $\forall i, j \in I, i \neq j$ is full row rank (see Assumption IV.2 in [9]). We equivalently rewrite conditions (i) and (ii) that ensure collision avoidance (convexified via the aforementioned procedure) in compact form as

$$\begin{aligned} &\Big[\tfrac{-\text{sgn}(\delta p_{x_{i,j}}[k])}{r_1} \;\; \tfrac{-\text{sgn}(\delta p_{y_{i,j}}[k])}{r_2} \;\; -1 \;\; 0_{1\times(n_p-1)} \;\; 0_{1\times(2+n_p)(H-1)}\Big] u_i \\ &+\Big[\tfrac{\text{sgn}(\delta p_{x_{i,j}}[k])}{r_1} \;\; \tfrac{\text{sgn}(\delta p_{y_{i,j}}[k])}{r_2} \;\; 0_{1\times(n_p)} \;\; 0_{1\times(2+n_p)(H-1)}\Big] u_j \leq 0, \end{aligned} \tag{8}$$

where for all $i \in I$, for all $k = 0, \ldots, H-1$

$$u_i[k] = (a_{xi}[k], a_{yi}[k]\,(t_{i,j}[k])_{j\in I, j\neq i})^T,$$

denotes the decision vector of agent i, that comprises the associated acceleration inputs as well as n_p auxiliary variables associated to its coupling with the other agents.

Inequality (8) is then in the format of (3). In particular, each row constraint in (8) imposes a safety constraint for each UAV pair (similar constraints would have to be instantiated in case of static obstacles). Columns corresponding to inputs from UAV i reflect how UAV i, through its acceleration decision variables, can influence the collision avoidance strategy with respect to UAV j, $j \in I \setminus \{i\}$. In such a formulation, at each time instance we are assuming complete knowledge by UAV i of both its own position and velocity; to obtain the corresponding information from other UAVs that are coupled in the same constraint while preventing information sharing with all UAVs in the fleet, we will be employing in the next section a distributed mechanism that is iterative, and at each iteration involves communication only with a subset of the agents considered as neighbours.

B. Local constraints

Each agent's decisions are subject to local constraints as well. These refer to limits on the acceleration of each UAV that is restricted in magnitude to $a_{i_{max}} \in \mathbb{R}^2$, $i \in I$. Moreover, for all $k = 0, \ldots, H-1$, for all $i \in I$, we have the equality constraints $t_{i,j}[k] = \Gamma_{i,j}[k]$, $j \in I, j \neq i$, where the auxiliary variables of agent i are involved. Overall, the local constraint set of agent $i \in I$ is given by

$$\begin{aligned} U_i = \Big\{u_i :\ &\text{for all } k = 0, \ldots, H-1, -a_{i_{max}} \leq a_i[k] \leq a_{i_{max}}, \\ &t_{i,j}[k] = \Gamma_{i,j}[k], \text{ for all } j \in I, j \neq i\Big\}. \end{aligned} \tag{9}$$

C. Objective functions

The finite horizon cost function for each UAV $i \in I$ is expressed as

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

where the first term is the running cost, with matrices $Q_i \succeq 0 and \text{R˙i} \succ 0$ of appropriate dimension, penalizes the state (in error dynamics) and control input, respectively, at each time instance k. The second term is a terminal cost with weighting matrix $Q_{H_i} \succ 0$. The third and fourth terms penalize agents aggregate (this implicitly defines functions ϕ_i and σ in (3)) from a common fixed target with coordinates $p_o = (p_{o_x}, p_{o_y})^T$, to be tracked by all UAVs. Here we require that the aggregate is close to the target at the terminal time H. Moreover, note that the division by N before the squared quantity in the last two terms is introduced to ensure that each J_i gets exactly $(1/N)$-th of the aggregative term included. The relative importance of the individual objectives (first two terms in (10)) compared to the common target (last two terms in (10)) is captured by the weighting coefficient $p_a \geq 0$.

It should be noted that this formulation of agents' objective functions satisfies standard assumptions in the game theoretic literature, such as strong monotonicity and lipschitz continuity with respect to agents' decisions (e.g., see standing Assumptions II.2 and II.3 in [9]).

IV. Solution methodology and analysis

In the first part of this section we introduce the network, its communication protocol and introduce the Primal-Dual Trades algorithm. This algorithm has been chosen due to its capacity to solve our general problem setting in a fast and distributed manner. Subsequently, we propose an aggregative distributed optimization framework in a receding horizon fashion, using Control Barrier Function (CBF) for inter drone collision avoidance. Finally, we analyse modified stability conditions for MPC, given that our formulation includes an aggregative term.

A. Primal-Dual TRADES algorithm

In this work we assume that each agent has knowledge about its local information: u_i, U_i, ϕ_i, J_i, A_i, b_i and can also acquire knowledge from its neighbours $\mathcal{N}_i$ by exchanging information with them. Here we assume a similar communication structure as in [9] and that graphs considered here are strongly connected.

The Primal-Dual Trades (Algorithm 1), based on [9] solves (3) in a distributed way by introducing tracking variables for the coupling constraint and the aggregative term. In

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: $u_i^{p+1} = u_i^p + \delta_{tr}(P_{U_i}[u_i^p - \gamma_{tr}\tilde{F}_i(u_i^p, \phi_i(u_i^p) + z_i^p) - \gamma_{tr} G_{u,i}(N(A_i u_i^p - b_i) + y_i^p, \lambda_i^p] - u_i^p)$
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^*

Algorithm 1, line 1 initializes the variables u_i which are the decision variables, λ_i the Lagrange multiplier associated with the coupling constraints, z_i and y_i variables used to track the aggregative term in the cost J_i and the coupled constraints in (3) respectively. The superscript $p \in \mathbb{N} \cup \{0\}$ denotes the iteration index. Line 2 loops through the variables which will be updated until convergence. In line 3, $\tilde{F}_i$ (11) is the pseudo gradient of J_i in (3) and $G_{u,i}$ (12) is the pseudo gradient of the constraints in (3), both with respect to the primal variable, i.e.,

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

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

where $\nabla_1 J_i$ is the gradient of J_i with respect to its first argument and $\nabla_2 J_i$ is the gradient of J_i with respect to its second argument. $[A_i]_l$ and $[b_i]_l$ indicate the l-th row of matrix A_i and vector b_i in (3), respectively and $\rho > 0$ is a constant satisfying $\rho > \frac{\delta_{tr}\gamma_{tr}}{w_{ii}}$ (see (33) in [9]).

The argument within brackets in line 3 acts as a gradient descent method with $\gamma_{tr} \in \mathbb{R}$ playing the role of the step size. In order to make sure this solution fit the solution space of each agent, $P_{U_i}[v]$ projects the argument v within the brackets onto the local feasible set U_i in (9). Parameter $\delta_{tr} \in (0,1)$ is used to perform a convex combination between the current value and the newly found estimate, acting as a way to trade how much to trust this new value. Line 4, updates the dual variable also in a gradient descent fashion, in which $G_{\lambda,i}$ (13) is the pseudo gradient of the constraints in (3), with respect to the dual variable, i.e.,

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

where e_l is the l-th vector of the canonical basis. Lines 5 and 6 have update laws for the aggregative cost tracking and the coupling constraint tracking variables respectively. Parameter w_{ij} is the i,j element of the network adjacency matrix, introduced for consensus among agents estimations.

Manuscript 517 submitted to 2023 62nd IEEE Conference on Decision and Control (CDC) . Received March 29, 2023.

The presented set-up satisfies all technical assumptions introduced in [9], which are omitted here in the interest of space. The subsequent theorem summarizes the convergence behaviour of Algorithm 1.

Theorem 1: 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 Nash equilibrium of the game $\mathscr{G}_a$.

B. Safe aggregative receding horizon control

The stated problem so far, has been proposed in open loop, in the sense that it is solved only one time, and returns a solution with the properties stated in Theorem 1. We now propose it to be used with MPC to enhance the overall performance. Algorithm 1 can then be adjusted to produce a safe distributed solution as shown in 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**: $x_i^0 = u_i[k-1] \in U_i,\ \lambda_i^0 \in \mathbb{R}^m_+,\ z_i^0 = 0_2,\ y_i^0 = 0_m$.
8: Run **Algorithm 1**
9: $u_i[k] = u_i^*[0]$
10: $e_i[k+1] = A_{d_i} e_i[k] + B_{d_i} a_i[k]$
11: UAV $j \leftarrow e_i[k+1],\ u_i[k], \forall j \in I, j \neq i$
12: **end for**

Algorithm 2 runs in parallel for all agents. Lines 1 to 3 act as the initialization phase, collecting necessary information for UAV i from all agents such as their initial and desired end state (including its own) and the common point to be tracked. This could be restricted to neighbouring agents $\mathscr{N}_i$ only, in case we decide to avoid collision exclusively within $\mathscr{N}_i$ and the aggregative goal was solely shared by $\mathscr{N}_i$. Line 6 updates at each iteration the collision avoidance constraints. This is followed in line 7 by an initialization of 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 is invoked (line 8), returning the optimal decision vector u_i^*. The first component of that finite-horizon decision sequence is parsed to $u_i[k]$ (line 9), and then the dynamics are rolled (line 10). Both $u_i[k]$, and $e_i[k+1]$ which is calculated as a result of applying $a_i[k]$ to the system, must be shared with all other UAVs for the calculation of coupled constraints in the next iteration of Algorithm 2.

C. Aggregative MPC stability discussion

We now study the stability of the adopted MPC formulation that unlike typical MPC settings, contains an aggregative term in the cost. The MPC optimization program can be written as

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

In the standard MPC setting without the aggregative term $\sigma(u)$, one would solve (14) repeatedly whenever a new e_0 is available [7], [20]. To ensure stability and recursive feasibility of (14), several ways of designing the terminal penalty $p(e_i) = e_i^T Q_{H_i} e_i$ and the terminal set constraint $\mathscr{E}_{H_i}$ have been proposed. Nonetheless, since in our formulation we consider $J_i(u_i, \sigma(u))$, and the cost J_i includes a quadratic cost on $\sigma(u)$, a cross term containing pairwise acceleration combinations of different UAVs will appear. This coupled formulation in the cost demands a deeper look in this problem's stability conditions [16]. Prior to studying the stability behavior of this aggregative system, let $T_i[k] = (t_{i,j}[k], t_{i,p}[k], \dots)^T$, denote the collection of all auxiliary variables appearing in u_i. Notice that the cost defined in (10) has $T_i^{*^T}[l] R_{d_i} T_i^*[l]$ within the term $u_i^T[l] R_i u_i[l]$, where R_{d_i} is a diagonal positive definite matrix collecting the terms of R_i weighting $t_{i,j}$. The terms weighted by R_{d_i} are introduced to satisfy the strong monotonicity assumption required for convergence of Algorithm 1. As such, we can choose the elements of R_{d_i} to be arbitrarily small and positive. In the remainder of our analysis, we will this assume that $T_i^T[k+1] R_{d_i} T_i[k+1] - T_i^{*^T}[k] R_{d_i} T_i^*[k]$ is neglected due to its minor effect in the cost. We thus only analyze the effect of the aggregative term in the MPC and do not take into account the CBF[1].

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 $\alpha_{H_i} > 0$ (note this is non-degenerate as $Q_{H_i} \succ 0$). Assume that there exists $\bar{k}$ such that for all $k > \bar{k}$, for all $i \in I$, $e_i[k] \in \mathscr{E}_{H_i}$. Furthermore, for all $i \in I$, for all $e_i[k] \in \mathscr{E}_{H_i}$:

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

where

$$g[k+H] = C \begin{bmatrix} A_d^{H-1} B_d & 0_{4 \times n_p}, \dots, B_d & 0_{4 \times n_p} \end{bmatrix} u_i + z_i[k] + p_\sigma[k],$$

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

For all $i \in I$ we then have that the dynamics of the state $e_i[k]$ generated by the MPC algorithm encoded by (14) with cost J_i as in (10), is stable.

Proof: Our proof follows [16]. To this end, let $J_i(k)$ to be the cost and $J_i^*(k)$ to be the optimal cost of the i^{th} agent at time index k as defined in (10), under the input

[1] This is the reason why in (14), with a slight abuse of notation, we consider $a_i \in U_i$ and not $u_i \in U_i$.

Manuscript 517 submitted to 2023 62nd IEEE Conference on Decision and Control (CDC) . Received March 29, 2023.

sequence $\mathbf{a}_i^*[k] = [a_i^*[k]\dots a_i^*[k+H-1]]$. A sub-optimal cost of $J_i(k+1)$ could be obtained by the sequence $[a_i^*[k+1]\dots a_i^*[k+H-1]a_i[k+H]]$. Due to the MPC formulation (14), we have that there exists a $\bar{k}$ such that for all $k>\bar{k}$, for all $i\in I, e_i[k+H]\in\mathscr{E}_{H_i}$. Due to the second assumption in Proposition 1 the control constraint will be satisfied, while by the first assumption $e_i[k+H+1]\in\mathscr{E}_{H_i}$ [16]. The cost of the sub-optimal sequence (recall that we are neglecting $T_i^T[k]R_{d_i}T_i[k]$) in $k+1$ is:

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

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

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

Based on the third assumption in Proposition 1, we then have that

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

which results in

$$J_i^*(k+1) \le J_i(k+1) \le J_i^*(k) - a_i^{*^T}[k]R_i a_i^*[k],$$

which shows that the optimal cost iterates form a non-increasing sequence (assuming that the initial condition for states and inputs is non-zero, otherwise we have the trivial case) which is bounded below from zero as $R_i \succ 0$, for all $i\in I$. As a result, the cost iterates form a convergent sequence; since the cost is quadratic in the state, the state iterates will converge as well to some point $e_{i_{eq}} = (p_{x_{i_{eq}}}, p_{y_{i_{eq}}}, 0, 0)^T$, $i\in I$, thus proving stability. At that point the acceleration input vanishes. Notice that due to the presence of the aggregative term the cost iterates do not necessarily converge to zero, thus we only have stability. In the specific case where the aggregative objective is chosen as the average of the individual agents' target locations p^d, then the cost exhibits a decoupled structure, and asymptotic stability can be ensured. ■

Establishing sufficient conditions for the satisfaction of the last assumption in Proposition 1 constitutes a topic of future work; preliminary investigations involve considering a newly formulated $e_i[k]$, which would take into account a convex combination of the desired point p_{d_i} and p_o instead of the current formulation.

V. Numerical example

We illustrate the performance via a detailed numerical simulation on a four vehicles position swapping case study. Agents start at initial positions. $p_{0_1} = [0,\ 1]^T$, $p_{0_2} = [0,\ -1]^T$, $p_{0_3} = [1,\ 0]^T$ and $p_{0_4} = [-1,\ 0]^T$. Each agent is expected to reach the desired position $p_{d_1} = [0,\ -1]^T$, $p_{d_2} = [0,\ 1]^T$, $p_{d_3} = [-1,\ 0]^T$ and $p_{d_4} = [1,\ 0]^T$, respectively, while maintaining their respective positions at the end of the predicted horizon in the vicinity of the geometrical centre $p_o = [0,\ 0]^T$ without colliding. This example was chosen as it illustrates the collision avoidance capabilities of our algorithm, since without intelligence guidance the agents would collide. We discretize the continuous time dynamics of each agent using a sample time $h = 0.2s$, and consider a maximum acceleration of $2m/s^2$ for both the x and y directions for all vehicles at all times. For our first and second simulations we use a prediction horizon of $H = 3$, while the agents' objective functions are parameterized by $Q_i = \mathrm{diag}(5,5,5,5)$, $R_i = \mathrm{diag}(2,2,2,2)$, $p_a = 1$ and Q_{H_i} is set as the solution of the algebraic Riccati equation associated to the unconstrained infinite horizon version of the quadratic objective function J_i. For this first simulation $\beta > 0$, a parameter weighting the terminal penalty (i.e. $\beta e_i^T Q_{H_i} e_i$) was set to $\beta = 1$. We set $\gamma_{\mathrm{tr}} = \delta_{\mathrm{tr}} = 0.1$ and the CBF boundaries were chosen for all UAVs as $r_1 = 0.25m$ and $r_2 = 0.5m$. As a stopping criterion for Algorithm 1 we consider $\max(||u_i^{p+1} - u_i^p||, ||\lambda_i^{p+1} - \lambda_i^p||) \le \Delta$, $\forall i \in I$ for a number of consecutive iterations equal to the graph diameter, where $\Delta > 0$ is a user defined tolerance. The resulting evolution of the vehicles' position at different time instances are illustrated in Figure 1. The value of γ_{cbf} for the CBF in this example was set to 0.1.

In Figure 1 each triangle depicts a different UAV, the parallelogram around it shows the boundary of the safe region, i.e., the boundary of the CBF, while the 'o' and 'x' symbols show each agents' start and target locations, respectively. The square shows the common target at the vicinity of which the aggregate agents' position need to be maintained. The UAVs manage to avoid collision and reach to their respective target. The boundaries and input constraints are also respected. During the time instance when UAVs are approaching each others safety boundary, when the TRADES algorithm is invoked to solve the associated finite horizon problem, 54 iterations are required to converge to an equilibrium solution within a numerical tolerance 3.5×10^{-2}. A remark is that, with this choice of parameters we can run up to 99.3% of the iterations (disregarding the communication burden) of Algorithm 2 within the sample time duration of $h = 0.2s$.

We now quantify the effect of γ_{cbf} on the resulting trajectories, as this captures implicitly how aggressive the resolution maneuver will be when agents are close to collision (e.g., see [20]). In particular, the smaller the value of γ_{cbf}, the more cautious the resulting controller tends to be. Given

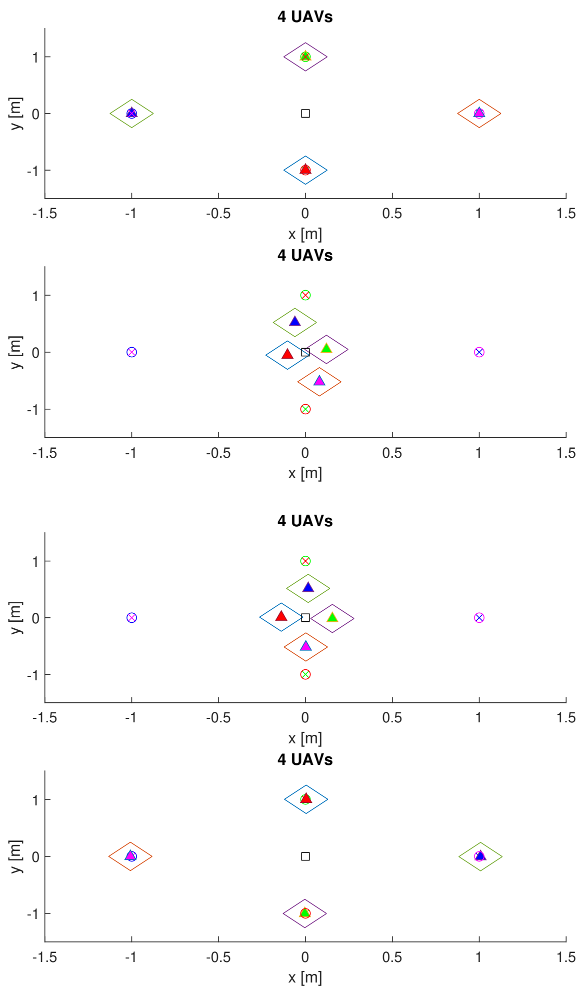

Fig. 1: UAVs' positions at four instances of the simulation. First panel: Starting positions. Second panel: At the instance coming the closest to each other's control barrier function boundary. Third panel: At the first instance after collision is resolved. Fourth panel: At the time instance when positions are swapped.

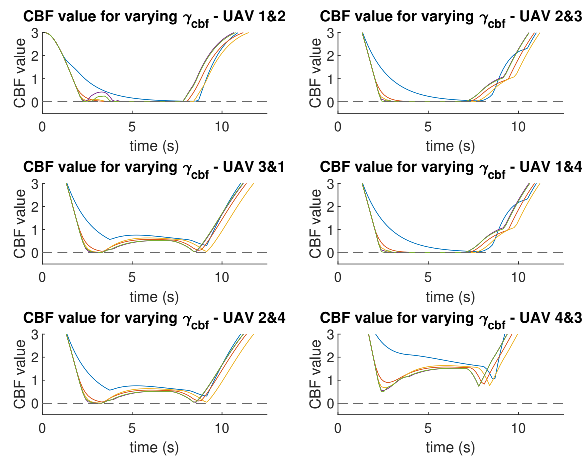

Fig. 2: CBF value between every pair of UAVs for varying γ_{cbf}. Color code: $\gamma_{\text{cbf}} = 0.1$ (blue), $\gamma_{\text{cbf}} = 0.3$ (red), $\gamma_{\text{cbf}} = 0.5$ (yellow), $\gamma_{\text{cbf}} = 0.7$ (purple), and $\gamma_{\text{cbf}} = 0.9$ (green). The dashed black line is the safe CBF value. Values above this line mean agents are safe.

that the CBF in (8) is taking into account only the very next step in terms of safety, a high value for γ_{cbf} make agents aggressive, while it has a shortsighted view in terms of safety. In addition, acceleration is limited, resulting in lack of sufficient control authority to correct initially inaccurate control actions early enough, which in turn may lead to feasibility issues.

Figure 2 shows the evolution of the CBF value for different values of γ_{cbf}. For these results we used a sample time of $h = 0.15s$, a $\beta = 0.1$, $\Delta = 1\times 10^{-3}$ and $a_{\max} = 4\,m/s^2$ while numerical values for all other parameters were set to the same values with the results associated to Figure 1. As expected, high values of γ_{cbf} tend to reduce the inter UAV CBF value when compared to lower ones. This can be seen by the minimum value of the green curve (corresponding to the highest value of γ_{cbf}) being the smallest among the minima of all other curves. The aggressive strategy tend to make each UAV less cautious and since they are moving towards each other's path, this results in UAVs bumping into each other's safe region much more often, making the team and their own performance worse. It is believed that this could be mitigated by introduction of a "predictive" component in the collision avoidance strategy. In other words, include further predicted states of the MPC in (5). Since we are using the procedure (7), decreasing radii values r_1 and r_2 as more CBF constraints are introduced for the predicted states in (5) could be beneficial to avoid infeasibility while giving some future safety awareness for the UAVs.

We also discuss the benefit of adopting an MPC framework along with Primal-Dual TRADES. Adopting MPC allows for more aggressive resolution maneuvers to be initiated closer to the point of collision, if compared to the use of Primal-Dual TRADES without an MPC algorithm. As the value of γ_{cbf} is increased, Algorithm 1 even with a short horizon outperforms Algorithm 2 with $H = 1$. In Figure 3 we show the trajectory of two simulations, one with $H = 1$ and another one with $H = 3$. For both simulations we set $\gamma_{\text{cbf}} = 0.5$ and the convergence tolerance to $\Delta = 10^{-4}$. All other parameters were set to the same values with the analysis of Figure 2.

At the upper panel of Figure 3 ($H = 1$) it can be observed that UAVs overshoot their final destination and oscillate around it substantially. On the lower panel ($H = 3$), the generated sequence of acceleration inputs result in agents arriving smoothly at their respective destinations.

VI. Concluding Remarks and Future Work

In this work we proposed a distributed, equilibrium seeking algorithm for a multi-UAV control problem. Safety constraints to account for collision avoidance regions were encoded by means of CBFs, while a recently developed primal-dual mechanism was employed to determine a Nash

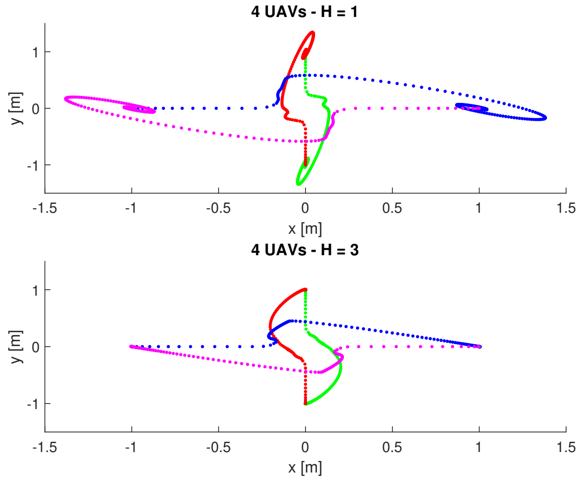

Fig. 3: UAVs' trajectories over a 250s simulation. Upper panel: Trajectories with prediction horizon set to $H = 1$. Lower panel: Trajectories with prediction horizon set to $H = 3$.

equilibrium strategy in a distributed manner. Moreover, we introduced a receding horizon implementation based on MPC and analyzed its stability properties. Our approach was first illustrated on a multi-UAV position swapping problem. We continued our experiments by conducting a parametric analysis on the effect of the CBF parameters on the efficacy of the resulting controller. We then finalized section V by displaying the benefits of adopting an MPC scheme through its performance comparison with a single horizon algorithm.

Current research concentrates towards investigating a terminal cost/set pair so that we simultaneously guarantee stability, safety, and recursive feasibility. Moreover, we wish to investigate whether a closed-loop Nash equilibrium solution can be determined exploiting the solution returned upon the application of MPC.

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] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer science & business media, 2013.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Kunal Garg and Dimitra Panagou. Control-Lyapunov and Control-Barrier Functions based quadratic program for spatio-temporal specifications. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1422–1429. IEEE, 2019.

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

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

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

[17] 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.

[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.