HAL
open science

Decentralized MPC for UAVs Formation Deployment and Reconfiguration with Multiple Outgoing Agents

Thomas Chevet, Cristina Vlad, Cristina Stoica Maniu, Youmin Zhang

To cite this version:

Thomas Chevet, Cristina Vlad, Cristina Stoica Maniu, Youmin Zhang. Decentralized MPC for UAVs Formation Deployment and Reconfiguration with Multiple Outgoing Agents. Journal of Intelligent and Robotic Systems, 2020, 97, pp.155-170. 10.1007/s10846-019-01025-x . hal-02100517

HAL Id: hal-02100517

https://hal.science/hal-02100517v1

Submitted on 5 Mar 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Journal of Intelligent & Robotic Systems manuscript No.
(will be inserted by the editor)

Decentralized MPC for UAVs Formation Deployment and Reconfiguration with Multiple Outgoing Agents

Thomas Chevet · Cristina Vlad · Cristina Stoica Maniu · Youmin Zhang

Received: date / Accepted: date

Abstract This paper presents a new decentralized algorithm for the deployment and reconfiguration of a multi-agent formation in a convex bounded polygonal area when considering several outgoing agents. The system is deployed over a two-dimensional convex bounded area, each agent being driven by its own linear model predictive controller. At each time instant, the area is partitioned into Voronoi cells associated with each agent. Due to the movement of the agents, this partition is time-varying. The objective of the proposed algorithm is to drive the agents into a static configuration based on the Chebyshev center of each Voronoi cell. When some agents present a non-cooperating behavior (e.g. agents required for a different mission, faulty agents, etc.), they have to leave the formation by tracking a reference outside the system's workspace. The outgoing agents and their objective positions partition the convex bounded polygonal area into working regions. Each remaining agent will track a new objective point allowing it to avoid the trajectory of the outgoing agents. The computation of this objective position is based on the agent's safety region (i.e. the intersection of the contracted Voronoi cell and the contracted working region). When the outgoing agents have left the workspace, the remaining agents resume their deployment objective. Simulation results on a formation of a team of unmanned aerial vehicles are finally presented to validate the algorithm proposed in this paper when several agents leave the formation.

Keywords Multi-agent systems · Unmanned aerial vehicles · Decentralized model predictive control · Voronoi tessellation · Formation reconfiguration

This work was supported by the CNRS' LIA on *Information, Learning and Control* and the Natural Sciences and Engineering Research Council of Canada.

T. Chevet
Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec-CNRS-Université Paris-Sud, Université Paris-Saclay 3, rue Joliot Curie, 91192 Gif-sur-Yvette, France
ORCID: 0000-0001-8599-7038
E-mail: thomas.chevet@l2s.centralesupelec.fr

C. Vlad
Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec-CNRS-Université Paris-Sud, Université Paris-Saclay 3, rue Joliot Curie, 91192 Gif-sur-Yvette, France
ORCID: 0000-0003-0250-5785
E-mail: cristina.vlad@l2s.centralesupelec.fr

C. Stoica Maniu
Laboratoire des Signaux et Systèmes (L2S), CentraleSupélec-CNRS-Université Paris-Sud, Université Paris-Saclay 3, rue Joliot Curie, 91192 Gif-sur-Yvette, France
ORCID: 0000-0002-5838-599X
E-mail: cristina.stoica@l2s.centralesupelec.fr

Y.M. Zhang
Department of Mechanical, Industrial and Aerospace Engineering, Concordia University, 1455 Blvd. de Maisonneuve Montréal, QC H3G 1M8, Canada
ORCID: 0000-0002-9731-5943
E-mail: ymzhang@encs.concordia.ca

1 Introduction

Given their reduced cost, light weight, ease of maneuverability and deployment, unmanned aerial vehicles (UAVs) have been the source of a constant interest for both research and industry for several years. With all these advantages, such vehicles are used for a large scope of applications such as forest fire surveillance and detection [2, 7, 18], ground and resource monitoring [15, 25] or 3D mapping and modeling [21, 33]. Control of UAVs is then an active research topic [16]. For several of these applications, such as monitoring or mapping missions, the use of a multi-agent system is beneficial and sometimes required since it distributes the work load. Each agent has then to be able to autonomously track a given reference determined by the needs of the mission

and/or maintain a given formation. All of this has to be done while avoiding collisions with other agents from the system.

To carry out such missions, an area coverage problem is often considered. Such problems have been vastly studied [9, 26, 30, 32]. Some of the considered approaches [20, 24] rely on set-theoretic methods [4], with different applications such as system deployment over a bounded region. For multi-agent system deployment and formation control, various strategies have been adopted such as proportional integral control [11], linear quadratic control [28], robust control [36], sliding mode control [34] or model predictive control (MPC) [23, 27]. When an agent becomes non-cooperating because it ends up being faulty, both a fault detection [37] and a fault tolerant control [39] algorithm are needed.

The study of area coverage problems mainly focuses on centroidal Voronoi configurations [9, 19, 30–32]. Such a configuration frequently relies on the centers of mass of a Voronoi tessellation [38]. The center of mass of a Voronoi cell is defined as the integral of the points of the cell weighted by a density function [22]. This point can then be laborious to obtain, depending on the considered density function. A different approach considers the Chebyshev centers of the Voronoi tessellation of the aforementioned region [8, 24]. The computation of such centers is of reduced complexity compared to the computation of the center of mass, a Chebyshev center being defined as the solution of a linear optimization problem.

This paper considers a multi-agent system with a time-varying number of agents, which are able either to leave the system when they end up being non-cooperating, or to join it. These agents are distributed over a convex two-dimensional *bounded deployment area* partitioned into a Voronoi tessellation. While deploying, the agents track the Chebyshev center of the cell they belong to, in order to converge towards a Chebyshev static configuration [8, 24]. The reconfiguration strategy for non-cooperating agents in [8] considers only *one outgoing agent* (i.e. an agent leaving the formation). The barycentric-based approach from [8] is not easily scalable for the case of *several outgoing agents* at the same time. The main reason is that if a remaining agent is located inside the area defined by the trajectories of two outgoing agents, its barycenter might end up close to one of these trajectories, leading to a possible collision with an outgoing agent. To overcome this inconvenience for the case of several outgoing agents, the main contribution of the present paper covers the design of a decentralized MPC technique based on safety regions of the remaining agents, allowing us to steer each agent towards a new objective position in order to achieve a Chebyshev static configuration. A 3-step procedure is proposed to compute these *safety regions* and these *new objectives*, which are not crossed by the outgoing agents' trajectories. (1) The hyperplanes defined by the *current position* and the *objective position* of each of the outgoing agents partition the *bounded deployment area* into *working regions* for the remaining agents. (2) The *safety region* is a convex set defined by the intersection of the *contracted Voronoi cell* and the *contracted working region* of the considered remaining agent. (3) The *new objective position* for each remaining agent is the point, belonging to its *safety region*, that minimizes the sum of the weighted distances to the agent's neighbors. This reconfiguration approach guarantees that the outgoing agents leave the bounded area without crossing the safety regions of the remaining agents. The remaining agents track the new objective positions computed at each time instant and ultimately resume their deployment objective towards a Chebyshev static configuration. In this paper, a multi-agent system composed of quadrotor UAVs is considered. In some applications such as forest fire monitoring or surveillance, a constant altitude is required to avoid damages. These applications justify a two-dimensional approach which can then be applied to unmanned ground or surface vehicles.

Section 2 describes the model of the global multi-agent system and the model of the individual UAV agents. Section 3 presents the mathematical tools needed for the deployment and reconfiguration algorithms as well as a discussion on the reconfiguration approach from [8]. Section 4 formulates the main result: the control strategy when several agents leave the formation. Finally, Section 5 details two simulation scenarios for UAV agents and the associated results. Concluding remarks and perspectives are drawn in Section 6.

General notation. In the following, $\mathbb{R}$ is the set of the real numbers and $\mathbb{N}$ is the set of the positive integers. The matrices $\mathbf{0}_{n\times m}$, $\mathbf{1}_{n\times m}$ and $\boldsymbol{I}_{n\times m}$ are of size $n \times m$, with $n, m \in \mathbb{N}$, and contain respectively only zeros, only ones and ones on the main diagonal. If $n = m$, the notations become $\mathbf{0}_n$, $\mathbf{1}_n$ and $\boldsymbol{I}_n$. The transpose of the matrix $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^\top$. A matrix denoted by $\mathrm{diag}\,(a_1, \ldots, a_n)$ is a diagonal matrix with diagonal elements $a_1, \ldots, a_n$. The norm $\|\mathbf{x}\|_2$ is the Euclidean norm of the vector $\mathbf{x}$ such that $\|\mathbf{x}\|_2^2 = \mathbf{x}^\top\mathbf{x}$. The quadratic form $\|\mathbf{x}\|_{\boldsymbol{Q}}$ is defined such that $\|\mathbf{x}\|_{\boldsymbol{Q}}^2 = \mathbf{x}^\top\boldsymbol{Q}\mathbf{x}$, where $\boldsymbol{Q} = \boldsymbol{Q}^\top$ is a strictly positive definite matrix, denoted by $\boldsymbol{Q} \succ 0$. The set of all integers from m to n, with $m \leq n$, is denoted by $\{m, \ldots, n\}$. The cardinality of a set $\mathscr{S}$ is denoted by $|\mathscr{S}|$. The Minkowski sum of two sets $\mathscr{A}$ and $\mathscr{B}$ is the set $\mathscr{A} \oplus \mathscr{B} = \{a + b | a \in \mathscr{A}, b \in \mathscr{B}\}$.

Multi-agent system notation. A multi-agent system is denoted by Σ and evolves in the state-space[1] $\mathbb{X}$. The subsystem $\Sigma_{\mathbb{W}}$ contains the part of the multi-agent system Σ which is steered into formation inside the convex bounded workspace $\mathbb{W} \subset \mathbb{X}$. The subystem $\Lambda \in \Sigma_{\mathbb{W}}$ contains the agents leaving the workspace. A set indexed by α is related to the agent $\alpha \in \Sigma$. A set $\mathbb{V}_\alpha^i$, $\mathscr{W}_\alpha^i$ or $\mathscr{S}_\alpha^i$ with a superscript $i \in \{\lambda, \lambda_{\mathbb{V}}, \lambda_{\mathscr{W}}\}$ is a contracted version of the corresponding

[1] In the context of multi-agent systems, the state-space is also called environment.

set $\mathbb{V}_\alpha$, $\mathscr{W}_\alpha$ or $\mathscr{S}_\alpha$ centered on its Chebyshev center with a contraction factor i.

2 Description of the System

This paper focuses on two aspects related to formation control: steering a team of UAVs in a desired formation and reconfiguring the formation in the case of outgoing non-cooperating agents. This section further presents both the mathematical models of the multi-agent system and of the individual UAV agents used in simulation.

2.1 General dynamics of the agents

Let Σ be a heterogeneous[2] multi-agent system composed of a time-varying number of agents $N(t)$ evolving in an environment $\mathbb{X}$. The agent $\alpha \in \Sigma$ is characterized by its state and input vectors $\mathbf{x}_\alpha \in \mathbb{X} \subseteq \mathbb{R}^n$ and $\mathbf{u}_\alpha \in \mathbb{U} \subseteq \mathbb{R}^m$. Each agent obeys the continuous-time nonlinear dynamics:

$$\begin{aligned} \dot{\mathbf{x}}_\alpha(t) &= \boldsymbol{f}_\alpha\left(\mathbf{x}_\alpha(t), \mathbf{u}_\alpha(t)\right) \\ \mathbf{y}_\alpha(t) &= \boldsymbol{C}_\alpha \mathbf{x}_\alpha(t) \end{aligned} \tag{1}$$

with $\boldsymbol{C}_\alpha \in \mathbb{R}^{p\times n}$ the output matrix, $\boldsymbol{f}_\alpha : \mathbb{X} \times \mathbb{U} \to \mathbb{Y}$ the evolution function and $\mathbf{y}_\alpha \in \mathbb{Y} \subseteq \mathbb{R}^p$ the measurement vector of agent α.

The dynamics described by (1) are linearized around an equilibrium point $(\bar{\mathbf{x}}_\alpha, \bar{\mathbf{u}}_\alpha)$, yielding $\bar{\mathbf{y}}_\alpha = \boldsymbol{C}_\alpha \bar{\mathbf{x}}_\alpha$. Thus the continuous-time linear time invariant dynamics are:

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}_\alpha(t) &= \boldsymbol{A}_\alpha \underbrace{\left(\mathbf{x}_\alpha(t) - \bar{\mathbf{x}}_\alpha\right)}_{\tilde{\mathbf{x}}_\alpha(t)} + \boldsymbol{B}_\alpha \underbrace{\left(\mathbf{u}_\alpha(t) - \bar{\mathbf{u}}_\alpha\right)}_{\tilde{\mathbf{u}}_\alpha(t)} \\ \underbrace{\mathbf{y}_\alpha(t) - \bar{\mathbf{y}}_\alpha}_{\tilde{\mathbf{y}}_\alpha(t)} &= \boldsymbol{C}_\alpha \left(\mathbf{x}_\alpha(t) - \bar{\mathbf{x}}_\alpha\right) \end{aligned} \tag{2}$$

where $\boldsymbol{A}_\alpha \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B}_\alpha \in \mathbb{R}^{n\times m}$ with the pair $(\boldsymbol{A}_\alpha, \boldsymbol{B}_\alpha)$ controllable and the pair $(\boldsymbol{C}_\alpha, \boldsymbol{A}_\alpha)$ observable. In the following, the variables with a tilde are the variations around the equilibrium values.

To simplify the reading, the time dependency is dropped for the remaining of the paper.

Assumption 1. *The formation is composed of a time-varying subsystem $\Sigma_{\mathbb{W}}$ of Σ evolving in a convex bounded workspace $\mathbb{W} \subset \mathbb{X}$.*

Assumption 2. *The communication between the agents is perfect with no package loss or communication delay.*

Assumption 3. *At $t = 0$ s, the agents of Σ are all participating in the formation and are not subject to any fault.*

[2] The proposed strategy holds for a heterogeneous multi-agent system. However to simplify the presentation of the results, the simulations are performed on a homogeneous multi-agent system.

2.2 Dynamical model of one UAV agent

Let Σ be composed of a set of quadrotor UAVs. These UAVs evolve in the Earth's frame $(\mathrm{O}, \mathbf{x}_{\text{Earth}}, \mathbf{y}_{\text{Earth}}, \mathbf{z}_{\text{Earth}})$. A mobile frame $(\mathrm{O}, \mathbf{x}_{\text{UAV}}, \mathbf{y}_{\text{UAV}}, \mathbf{z}_{\text{UAV}})$ is attached to each UAV. Both frames are presented in Fig. 1. This type of UAV is actuated by four propellers driven by brushless direct current (BLDC) motors. Each propeller generates a thrust $\mathbf{t}_i = t_i \mathbf{z}_{\text{UAV}}$, with $i \in \{1, \ldots, 4\}$, their sum being $\mathbf{f}_t = f_t \mathbf{z}_{\text{UAV}}$. The torques generated by the difference of propeller thrusts are $\mathbf{m}_x = \tau_x \mathbf{x}_{\text{UAV}}$, $\mathbf{m}_y = \tau_y \mathbf{y}_{\text{UAV}}$ and $\mathbf{m}_z = \tau_z \mathbf{z}_{\text{UAV}}$. The four elements f_t, τ_x, τ_y and τ_z are linked to the propellers' thrusts t_i, with $i \in \{1, \ldots, 4\}$, as detailed in [14]:

$$\begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ L & -L & 0 & 0 \\ 0 & 0 & L & -L \\ -C & -C & C & C \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \\ t_4 \end{bmatrix} \tag{3}$$

where L is the length of an arm from the origin O to the axis of a propeller as shown in Fig. 1 and C is the thrust to moment ratio.

The state-space model of the UAV is written by considering the state vector $\mathbf{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^\top$, where x, y and z are the UAV's position in the Earth's frame, ϕ, θ and ψ the roll, pitch and yaw angles defining the orientation of the UAV in the Earth's frame and v_x, v_y, v_z, ω_x, ω_y and ω_z their respective time derivatives. The state-space model itself is derived from Lagrangian mechanics [5] and can be found in Appendix A along with the numerical values of the parameters.

Moreover, the thrust generated by a propeller is linked to the duty cycle of the pulse width modulation voltage going through its corresponding BLDC motor by the first order relation [1]:

$$\dot{t}_i + \omega_m t_i = K \omega_m u_i, \text{ with } i \in \{1, \ldots, 4\}, \tag{4}$$

where K is the motor gain, ω_m its bandwidth, and u_i the duty cycle of its supply voltage.

Assumption 4. *The time constant of the DC motor ω_m^{-1} is negligible compared to that of the UAV [35] and to the sampling period.*

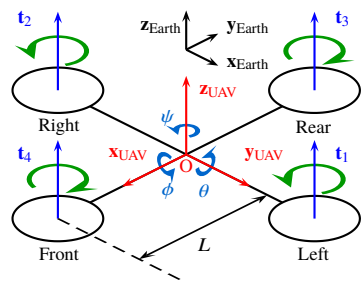

Fig. 1 Schematic representation of a quadrotor UAV.

Based on Assumption 4 and for simulation purpose, the equation (4) is simplified to $t_i = Ku_i$, with $i \in \{1, \dots, 4\}$, the vector $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^\top$ being the input signal of the model. All the numerical values for the model's parameters are gathered in Table 1 in Appendix A.

Finally, the UAV is considered to be equipped with a set of sensors providing measurements such that $\boldsymbol{C} = \boldsymbol{I}_{12}$. The ouptput space $\mathbb{Y}$ and the state-space $\mathbb{X}$ are then identical in (1) and (2).

2.3 Model linearization

For linearization purpose, the considered equilibrium point corresponds to the hovering state of the UAV. At steady state, the state vector is defined as $\bar{\mathbf{x}} = \begin{bmatrix} \bar{x} & \bar{y} & \bar{z} & \mathbf{0}_{1\times 2} & \bar{\psi} & \mathbf{0}_{1\times 6} \end{bmatrix}^\top$, where $\bar{x}$, $\bar{y}$ and $\bar{z}$ represent the equilibrium position of the UAV and $\bar{\psi}$ its equilibrium orientation. Thus, the total thrust of the propellers is $\bar{f}_t = mg$, which is equally distributed over all the propellers to have $\bar{\tau}_x = \bar{\tau}_y = \bar{\tau}_z = 0$, and, equivalently, $\bar{u}_i = \frac{mg}{4K}$, with $i \in \{1, \dots, 4\}$.

The nonlinear model from the equations in Appendix A is linearized around the equilibrium point $(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ leading to a linear state-space model around this point:

$$\dot{\tilde{\mathbf{x}}} = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \boldsymbol{I}_3 \\ \mathbf{0}_3 & \boldsymbol{A}_1 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tilde{\mathbf{x}} + K \begin{bmatrix} \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 1} & \mathbf{0}_{8\times 1} \\ \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} \\ \frac{L}{I_x} & -\frac{L}{I_x} & 0 & 0 \\ 0 & 0 & \frac{L}{I_y} & -\frac{L}{I_y} \\ -\frac{C}{I_z} & -\frac{C}{I_z} & \frac{C}{I_z} & \frac{C}{I_z} \end{bmatrix} \tilde{\mathbf{u}} \tag{5}$$

where m is the UAV's mass, I_x, I_y and I_z are the moments of inertia with respect to the axes $\mathbf{x}_{\text{UAV}}$, $\mathbf{y}_{\text{UAV}}$ and $\mathbf{z}_{\text{UAV}}$ illustrated in Fig. 1 and g is the gravitational acceleration, with $\boldsymbol{A}_1 = \begin{bmatrix} g\sin\bar{\psi} & g\cos\bar{\psi} & 0 \\ -g\cos\bar{\psi} & g\sin\bar{\psi} & 0 \\ 0 & 0 & 0 \end{bmatrix}$. The observation equation is omitted since $\boldsymbol{C} = \boldsymbol{I}_{12}$. The numerical values of these parameters are given in Appendix A, Table 1.

Assumption 5. *The yaw angle ψ and the altitude z are constant during the entire flight and equal to* 0 *and $\bar{z}$ respectively. With small variations of the pitch and roll angles θ and ϕ, the model is linear time invariant [1].*

The subspace of $\mathbb{X}$ whose elements are of the form[3] $\mathbf{x} = \begin{bmatrix} x & y & \bar{z} & \mathbf{0}_{1\times 9} \end{bmatrix}^\top$ is denoted in the following by $\mathscr{X}$. To simplify the reading, all the fixed coordinates of the elements of $\mathscr{X}$ will be dropped and $\mathscr{X}$ will be considered as a subset of $\mathbb{R}^2$, $\mathscr{X} \subseteq \mathbb{R}^2$. Thus, if $\mathbf{x} \in \mathscr{X}$, then $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^\top$.

[3] This means that all the elements of $\mathscr{X}$ have a constant altitude and null angles, speed and angular speed.

In line with Assumption 5, the following bounds are imposed [1]:

$$|\phi|, |\theta| \leq \frac{\pi}{12} \text{ rad} \qquad \text{and} \qquad |\psi| \leq 0.01 \text{ rad} \tag{6}$$

$$|v_x|, |v_y| \leq 5 \text{ m} \cdot \text{s}^{-1} \qquad \text{and} \qquad |v_z| \leq 0.1 \text{ m} \cdot \text{s}^{-1}. \tag{7}$$

If these inequalities hold, it is considered that the linearized model holds during the entire flight and this model will be used to design a controller in Section 4.1.

3 Problem Formulation

In Section 2.1, the multi-agent system Σ is assumed to evolve in an environment $\mathbb{X} \subset \mathbb{R}^n$, a part of it, $\Sigma_{\mathbb{W}} \subseteq \Sigma$, being steered into formation in a convex workspace $\mathbb{W} \subset \mathbb{X}$. This section presents mathematical objects constructed on this workspace $\mathbb{W}$, the Voronoi tessellation and the Chebyshev center of a bounded set. It finally presents the main goal of the global multi-agent system Σ based on these objects.

3.1 Voronoi tessellation

Assumption 6. *The workspace $\mathbb{W}$ is a convex strict subset of the state-space $\mathbb{X}$. The projection of $\mathbb{W}$ on $\mathscr{X}$ is denoted by $\mathscr{W}$ and it is a bounded polygon[4].*

With Assumption 6, the set $\mathscr{W} \subset \mathbb{R}^2$ is a convex bounded polygon, $\mathbb{R}^2$ being equipped with the Euclidean norm $\|\cdot\|_2$. If $|\Sigma_{\mathbb{W}}|$ agents are considered, it can be partitioned into $|\Sigma_{\mathbb{W}}|$ Voronoi cells such that:

$$\mathbb{V}_\alpha = \left\{ \mathbf{w} \in \mathscr{W} \middle| \left\| \mathbf{x}_\alpha^w - \mathbf{w} \right\|_2 \leq \left\| \mathbf{x}_\beta^w - \mathbf{w} \right\|_2, \forall \beta \neq \alpha \right\} \tag{8}$$

with $\mathbb{V}_\alpha \cap \mathbb{V}_\beta = \emptyset, \forall \alpha, \beta \in \Sigma_{\mathbb{W}}$, and $\mathscr{W} = \bigcup_{\alpha \in \Sigma_{\mathbb{W}}} \mathbb{V}_\alpha$, where $\mathbf{x}_\alpha^w$ and $\mathbf{x}_\beta^w$ are the projections of the state vectors $\mathbf{x}_\alpha, \mathbf{x}_\beta \in \mathbb{W}$ of agents α and β on $\mathscr{W}$, with α, $\beta \in \Sigma_{\mathbb{W}}$. In addition to their convexity, the cells $\mathbb{V}_\alpha$ are bounded polygons due to the boundedness of $\mathscr{W}$.

In the following, the $\mathscr{H}$-representation [29] of a polytope $\mathscr{P}$ in $\mathbb{R}^n$ is such that:

$$\mathscr{P} = \{ \mathbf{x} \in \mathbb{R}^n | \boldsymbol{H}_{\mathscr{P}} \mathbf{x} \leq \boldsymbol{\Theta}_{\mathscr{P}} \}$$

where $\boldsymbol{H}_{\mathscr{P}} \in \mathbb{R}^{m\times n}$ and $\boldsymbol{\Theta}_{\mathscr{P}} \in \mathbb{R}^m$.

With Assumption 2, each agent receives full state information from its neighbors and is then able to compute its own Voronoi cell. Moreover, considering Assumption 1 and due to the agents' movement inside $\mathscr{W}$, the Voronoi tessellation is time-varying.

The Voronoi cells partitioning the set $\mathscr{W}$ being defined, a remarkable point of these cells is needed for formation

[4] The set $\mathbb{W}$ will be used as a set of constraints in the MPC optimization problem (12). The set $\mathscr{W}$ will be used when only two-dimensional computations are needed.

purpose. However, several points can qualify, e.g. the center of mass (defined as the integral of the points of the cell weighted by a density function), the barycenter (defined as the weighted average of the vertices of the cell), the Chebyshev center etc. In this paper, the Chebyshev center is chosen as the remarkable point of a Voronoi cell and it is presented in the following section.

This choice is motivated by the simplicity of the linear optimization problem to be solved once the $\mathscr{H}$-representation of a cell is known. By construction, the Chebyshev center of a polytope is the point maximizing the distance to all the edges of the cell, thus reducing the collision risk between agents when tracking this point.

3.2 Chebyshev center

Let $\mathscr{P}$ be a convex bounded polytope in $\mathbb{R}^n$. Its Chebyshev center, denoted by $\mathbf{c}^c$, is the center of the largest Euclidean ball $\mathscr{B} = \{\mathbf{x} \in \mathbb{R}^n | \, \|\mathbf{x} - \mathbf{c}^c\|_2 \leq r\}$ lying in $\mathscr{P}$ where r is the radius of the ball $\mathscr{B}$. If $\boldsymbol{h}^i_{\mathscr{P}}$ and $\theta^i_{\mathscr{P}}$, with $i \in \{1, \ldots, m\}$, denote the lines of $\boldsymbol{H}_{\mathscr{P}}$ and $\boldsymbol{\Theta}_{\mathscr{P}}$ from the $\mathscr{H}$-representation of $\mathscr{P}$, the values of the Chebyshev center and the radius are obtained by solving the linear optimization problem [6]:

$$\begin{aligned} &\underset{\mathbf{c}^c, r}{\text{maximize}} \quad r \\ &\text{subject to} \\ &\quad r \geq 0, \\ &\theta^i_{\mathscr{P}} \geq \boldsymbol{h}^i_{\mathscr{P}} \mathbf{c}^c + \left\| \boldsymbol{h}^i_{\mathscr{P}}{}^\top \right\|_2 r, \forall i \in \{1, \ldots, m\}. \end{aligned} \tag{9}$$

In the following, the considered polytopes are the polygonal Voronoi cells $\mathbb{V}_\alpha$, with $\alpha \in \Sigma_{\mathbb{W}}$. Since each agent knows the $\mathscr{H}$-representation of its cell, they are able to compute the corresponding Chebyshev centers. Moreover, since the Voronoi tessellation is time-varying, the Chebyshev centers are also time-varying.

Assumption 7. *The set $\mathscr{W}$ and the number of agents in the formation are given such that the Chebyshev center of each cell $\mathbb{V}_\alpha$, with $\alpha \in \Sigma_{\mathbb{W}}$, is unique.*

Definition 1 (Static configuration [24])**.** *A static configuration for a formation is achieved whenever $\mathbf{u}_\alpha(k) = 0$ and $\mathbf{x}_\alpha(k) = \mathbf{x}_\alpha(k_0)$, $\forall \alpha \in \Sigma$, $\forall k \geq k_0$.*

Definition 2 (Chebyshev configuration [24])**.** *A Chebyshev configuration for a formation is achieved whenever an agent's position $\mathbf{x}^w_\alpha$ coincides with its associated Chebyshev center $\mathbf{c}^c_\alpha$.*

When Assumption 7 is verified and the agents are steered with a full state-feedback control law, it has been proven in [12, 24] that the formation converges towards a static configuration if and only if it is a Chebyshev configuration.

3.3 Discussion on the barycentric reconfiguration approach

In [8], three decentralized MPC based algorithms are presented for the deployment of the agents towards a static Chebyshev configuration inside $\mathbb{W}$ and the reconfiguration of the agents in the case of an agent entering or leaving the workspace. This last case occurs when an agent becomes non-cooperating because it is either not needed anymore inside the workspace or faulty.

The barycentric-based reconfiguration algorithm [8] considers only one outgoing agent. This section briefly presents the weighted barycenter computation for an agent remaining inside the workspace and explains the potential collision issues when several outgoing agents leave the formation at the same time.

When an agent $l \in \Sigma_{\mathbb{W}}$ becomes non-cooperating, it will start tracking a reachable point $\mathbf{x}^{\text{ref}}_l$ outside the workspace $\mathbb{W}$. It then broadcasts its position $\mathbf{x}_l$ and objective $\mathbf{x}^{\text{ref}}_l$ to the agents of $\Sigma_{\mathbb{W}} \setminus \{l\}$ remaining inside $\mathbb{W}$. Upon receiving these information, the agents of $\Sigma_{\mathbb{W}} \setminus \{l\}$ will change their objective to their neighbors' weighted barycenter.

Definition 3. *For an agent $\alpha \in \Sigma_{\mathbb{W}}$, a* neighbor *$\nu$ is either an agent having a Voronoi cell $\mathbb{V}_\nu$ contiguous to $\mathbb{V}_\alpha$ or a vertex of $\mathbb{V}_\alpha$ lying on the border of $\mathscr{W}$. The set of all the neighbors of α is denoted by N_α.*

The projections of $\mathbf{x}_l$ and $\mathbf{x}^{\text{ref}}_l$ on $\mathscr{X}$ define a hyperplane $\mathscr{H}$ in $\mathscr{X}$. Let α be an agent of $\Sigma_{\mathbb{W}} \setminus \{l\}$ and $\nu \in \mathrm{N}_\alpha$ a neighbor of α. If their positions $\mathbf{x}^w_\alpha$ and $\mathbf{x}^w_\nu$ lie in the same half-space of $\mathscr{X}$ and $d\left(\mathbf{x}^w_\alpha, \mathscr{H}\right) < d\left(\mathbf{x}^w_\nu, \mathscr{H}\right)$, the weight of ν is set to $\kappa > 1$; else, the weight is set to 1. The neighbors' barycenter $\mathbf{c}^{\text{bar}}_\alpha$ is the average of the weighted positions $\mathbf{x}^w_\nu$:

$$\mathbf{c}^{\text{bar}}_\alpha = \left(\sum_{\nu \in \mathrm{N}_\alpha} \omega_\nu \mathbf{x}^w_\nu \right) / \left(\sum_{\nu \in \mathrm{N}_\alpha} \omega_\nu \right)$$

with $\omega_\nu \in \{1, \kappa\}$ the weight associated to the neighbor ν.

Example 1 (Construction of a neighbors' barycenter) In Fig. 2, the seven agents belonging to $\mathscr{W}$ are represented by dots. The neighbors of the agent 2 (i.e. the elements of N_2) are represented by red dots. In this scenario, the agent 5 leaves the workspace to join $\mathbf{x}^{\text{ref}}_5$, thus defining the hyperplane $\mathscr{H}$. In order to compute the neighbors' barycenter $\mathbf{c}^{\text{bar}}_2$ (in red) of agent 2 which is an element of $\Sigma_{\mathbb{W}} \setminus \{5\}$, one has first to identify the neighbors of agent 2. The agents 3, 5, 6 and 7 are agents having a Voronoi cell contiguous to $\mathbb{V}_2$ and neighbors 8 and 9 are vertices of $\mathbb{V}_2$ lying on the border of $\mathscr{W}$. By Definition 3, they are then the neighbors of agent 2. The dashed line is the set of points as far from $\mathscr{H}$ as agent 2, thus all the neighbors being below this line are farther from $\mathscr{H}$ than agent 2 and receive the weight κ. The neighbors' barycenter of agent 2 (the red asterisk) is:

$$\mathbf{c}^{\text{bar}}_2 = \frac{\kappa \left(\mathbf{x}^w_7 + \mathbf{x}^w_8 + \mathbf{x}^w_9 \right) + \mathbf{x}^w_3 + \mathbf{x}^w_5 + \mathbf{x}^w_6}{3\,(\kappa + 1)}. \qquad \square$$

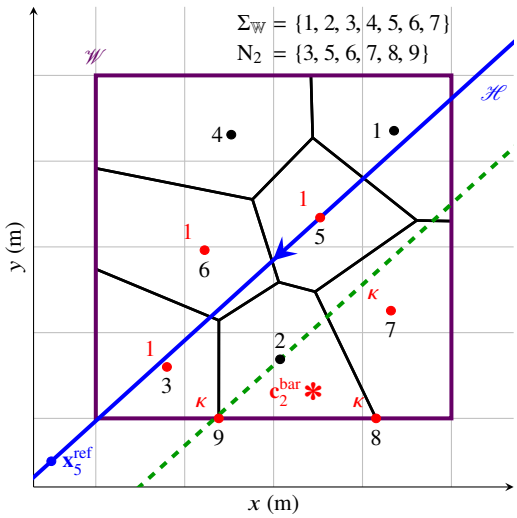

Fig. 2 Computation of agent 2's neighbors' barycenter.

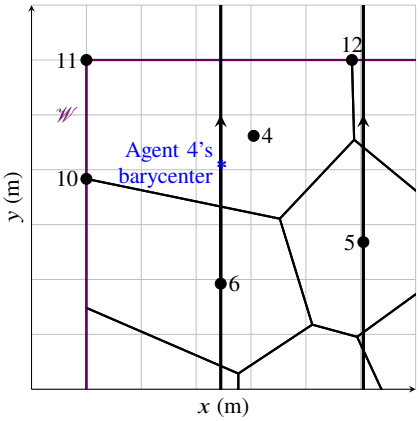

Fig. 3 Example of barycenter position in the case of several outgoing agents.

The method presented in [8] guarantees by construction that the agents remaining in the workspace will move away from the trajectory of the outgoing agent's trajectory. However, it can be difficult to deal with the situation of several agents leaving the workspace at the same time.

Example 2 (Illustration of a limit case of the barycentric-based approach) An example of such a situation is illustrated in Fig. 3. In this figure, agents 5 and 6 leave the workspace and agent 4 ends up between two hyperplanes. Thus, all the neighbors of agent 4 ($N_4 = \{5, 6, 10, 11, 12\}$) receive the same weight and the barycenter (the asterisk in Fig. 3) ends up lying on the trajectory of one of the outgoing agents. □

Some adjustments are then needed to guarantee that an agent remaining in the workspace will be driven away from the outgoing agents.

In order to overcome this situation, this paper presents a new way of driving the agents remaining in the workspace when one or several agents leave the workspace at the same time. The proposed approach is based on the computation of a new objective point $\mathbf{c}_\alpha^{\min}$ minimizing the norm of the vector[5]:

$$\sum_{\nu \in N_\alpha} \omega_\nu \left(\mathbf{c}_\alpha^{\min} - \mathbf{x}_\nu^w\right) \tag{10}$$

with N_α the set of the neighbors of agent α, under the condition that $\mathbf{c}_\alpha^{\min}$ belongs to a safety region avoiding the trajectories of the outgoing agents. The computation of this safety region along with the control technique necessary to lead the agent to its objective are presented in the next section.

4 Control Algorithm

This section presents the main result of this paper: a new strategy based on a decentralized linear MPC technique for the reconfiguration of a multi-agent system deployed in a bounded polygon when more than one of the agents leave the system.

4.1 Proposed model predictive control technique

The controller presented here is meant to deploy the multi-agent system $\Sigma_{\mathbb{W}}$ inside a bounded convex polygonal workspace and to reconfigure it when one or more agents have to leave the workspace. The continuous-time linear model (5) is discretized with a sampling period $T_s = 0.2$ s giving the discrete-time linear time-invariant dynamics:

$$\widetilde{\mathbf{x}}(k+1) = \boldsymbol{F}\widetilde{\mathbf{x}}(k) + \boldsymbol{G}\widetilde{\mathbf{u}}(k). \tag{11}$$

In the following, $\widetilde{\mathbf{x}}(k)$, $\widetilde{\mathbf{u}}(k)$, $\boldsymbol{F}$ and $\boldsymbol{G}$ are indexed by $\alpha \in \Sigma_{\mathbb{W}}$ to differentiate the agents.

The control signal $\widetilde{\mathbf{u}}_\alpha(k)$ for α is given by solving the convex optimization problem [8]:

$$\begin{aligned}\underset{\widetilde{\mathbf{u}}_\alpha}{\text{minimize}} \quad & \sum_{i=0}^{N_p-1} \Big(\left\|\widetilde{\mathbf{x}}_\alpha(k+i) - \widetilde{\mathbf{x}}_\alpha^{\text{obj}}(k)\right\|_{\boldsymbol{Q}}^2 \\ & + \left\|\widetilde{\mathbf{u}}_\alpha(k+i) - \widetilde{\mathbf{u}}_\alpha^{\text{obj}}(k)\right\|_{\boldsymbol{R}_1}^2 \\ & + \left\|\widetilde{\mathbf{u}}_\alpha(k+i) - \widetilde{\mathbf{u}}_\alpha(k+i-1)\right\|_{\boldsymbol{R}_2}^2 \Big) \\ & + \left\|\widetilde{\mathbf{x}}_\alpha(k+N_p) - \widetilde{\mathbf{x}}_\alpha^{\text{obj}}(k)\right\|_{\boldsymbol{P}_\alpha}^2\end{aligned} \tag{12a}$$

subject to

$$\widetilde{\mathbf{x}}_\alpha^{\text{obj}}(k) = \boldsymbol{F}_\alpha \widetilde{\mathbf{x}}_\alpha^{\text{obj}}(k) + \boldsymbol{G}_\alpha \widetilde{\mathbf{u}}_\alpha^{\text{obj}}(k), \tag{12b}$$

$$\widetilde{\mathbf{x}}_\alpha(k+i+1) = \boldsymbol{F}_\alpha \widetilde{\mathbf{x}}_\alpha(k+i) + \boldsymbol{G}_\alpha \widetilde{\mathbf{u}}_\alpha(k+i), \tag{12c}$$

$$\widetilde{\mathbf{x}}_\alpha(k+i+1) \in \mathbb{W}, \tag{12d}$$

$$\widetilde{\mathbf{u}}_\alpha(k+i) \in \mathbb{U}, \tag{12e}$$

$$\widetilde{\mathbf{x}}_\alpha(k+N_p) \in \mathbb{V}_\alpha^\lambda(k). \tag{12f}$$

[5] In the case of the neighbors' barycenter, this vector is the null vector.

Objective point. In the case of the multi-agent system deployment, the objective point $\widetilde{\mathbf{x}}_\alpha^{\text{obj}}$ of agent $\alpha \in \Sigma$ is the Chebyshev center $\widetilde{\mathbf{x}}_\alpha^c = \left[\mathbf{c}_\alpha^{c\top}\ \mathbf{0}_{1\times 10}\right]^\top$. When an agent leaves the workspace, its objective point is changed to a reachable point outside the workspace $\widetilde{\mathbf{x}}_\alpha^{\text{ref}}$. An agent remaining inside the workspace either keeps $\widetilde{\mathbf{x}}_\alpha^c$ as an objective or changes it to $\widetilde{\mathbf{x}}_\alpha^{\min} = \left[\mathbf{c}_\alpha^{\min\top}\ \mathbf{0}_{1\times 10}\right]^\top$ following the rules that will be described in Section 4.3, with $\mathbf{c}_\alpha^{\min}$ being described in Section 4.2.

Weights. In (12a), $\boldsymbol{Q} = \boldsymbol{Q}^\top$, $\boldsymbol{R}_1 = \boldsymbol{R}_1^\top$, $\boldsymbol{R}_2 = \boldsymbol{R}_2^\top \succ 0$ are diagonal matrices. The matrix $\boldsymbol{P}_\alpha$ is the solution of the Riccati equation [3]:

$$\begin{aligned}\boldsymbol{P}_\alpha &= \boldsymbol{F}_\alpha^\top \boldsymbol{P}_\alpha \boldsymbol{F}_\alpha + \boldsymbol{Q} \\ &\quad - \boldsymbol{F}_\alpha^\top \boldsymbol{P}_\alpha \boldsymbol{G}_\alpha \left(\boldsymbol{R}_1 + \boldsymbol{G}_\alpha^\top \boldsymbol{P}_\alpha \boldsymbol{G}_\alpha\right)^{-1} \boldsymbol{G}_\alpha^\top \boldsymbol{P}_\alpha \boldsymbol{F}_\alpha.\end{aligned} \tag{13}$$

The elements of $\boldsymbol{Q}$, $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are chosen in order to prioritize the importance of the cost function terms as follows: reach the agent's designated position at constant altitude (with $\boldsymbol{Q}$), limit the input signal amplitude (with $\boldsymbol{R}_1$) and its variations (with $\boldsymbol{R}_2$).

Constraints. The constraint (12b) imposes that the point $\widetilde{\mathbf{x}}_\alpha^{\text{obj}}(k)$ is an equilibrium point of the linearized system for the objective control input $\widetilde{\mathbf{u}}_\alpha^{\text{obj}}(k)$. The UAV dynamics lead to $\widetilde{\mathbf{u}}_\alpha^{\text{obj}}(k) = \mathbf{0}_{4\times 1}$. Then, in the case of a UAV, the second term of the cost function (12a) is meant to minimize the amplitude of the control input.

The constraint (12d) imposes that the state remains inside the allowed range defined for the state variables in (6) and (7). In the case of an outgoing agent, $\mathbb{W}$ becomes $\mathbb{X}$ for (12d). In addition to these constraints on the state vector, bounds are also defined for the control input (12e) according to the physical limits of the actuators.

In (12f), the terminal set $\mathbb{V}_\alpha^\lambda(k)$ is computed such that $\mathbb{V}_\alpha^\lambda(k) = \mathbf{c}_\alpha^c(k) \oplus \lambda\left(\mathbb{V}_\alpha(k) \oplus \left\{-\mathbf{c}_\alpha^c(k)\right\}\right)$, where $\lambda \in [0, 1)$. The set $\mathbb{V}_\alpha^\lambda(k)$ is then a contracted version of $\mathbb{V}_\alpha(k)$ centered on its Chebyshev center. This constraint ensures that $\mathbb{V}_\alpha^\lambda(k)$ is a controlled λ-contractive set [4]. This constraint is dropped in the case of an outgoing agent.

Prediction horizon. The objective point is considered constant over the prediction horizon N_p, while it actually changes when the agents are moving. Thus, the value of N_p has to be sufficiently small such that the real objective point at the end of the prediction horizon is close to its approximated position. Moreover, the prediction horizon N_p has to be sufficiently large in order to satisfy both the constraints (12e) and (12f).

4.2 New objective for the remaining agents

As mentioned in Section 3.3, when several agents leave the workspace at the same time, there is no guarantee that the algorithm presented in [8] drives the remaining agents away from the outgoing agents. A new objective $\mathbf{c}_\alpha^{\min}$ has then to be considered for each agent $\alpha \in \Sigma_{\mathbb{W}}$ remaining in formation. This objective minimizes the norm of the vector from (10) while remaining in a safety region which has the strong property (by construction) that it does not contain the trajectories of the outgoing agents. This *safety region* is computed as the intersection of the *agent's contracted Voronoi cell* and the *agent's contracted working region*. The construction of these sets as well as the assignment of the weights to the considered agent's neighbors are presented below. When these elements are defined, the new objective $\mathbf{c}_\alpha^{\min} \in \mathbb{R}^2$ is then the solution of the optimization problem (14):

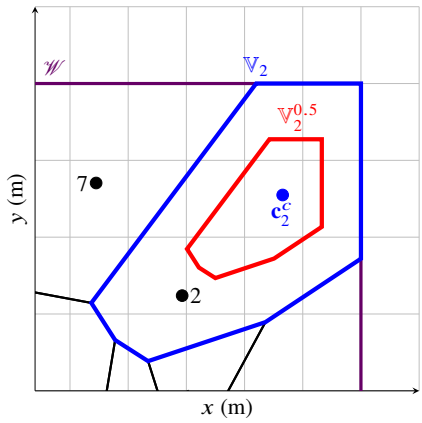

Fig. 4 Construction of the contracted Voronoi cell $\mathbb{V}_2^{0.5}$ with a contraction factor $\lambda_{\mathbb{V}} = 0.5$ from the Voronoi cell $\mathbb{V}_2$.

$$\begin{aligned}&\underset{\mathbf{c}_\alpha^{\min}}{\text{minimize}} && \sum_{\nu \in \mathrm{N}_\alpha} \omega_\nu \left\|\mathbf{c}_\alpha^{\min} - \mathbf{x}_\nu^w\right\|_2^2 \\ &\text{subject to} && \mathbf{c}_\alpha^{\min} \in \mathscr{W}_\alpha^{\lambda_{\mathscr{W}}} \cap \mathbb{V}_\alpha^{\lambda_{\mathbb{V}}}\end{aligned} \tag{14}$$

where N_α is the set of all neighbors of agent α, $\mathbf{x}_\nu^w$ is the projection on $\mathscr{W}$ of the state-space vector of the neighbor $\nu \in \mathrm{N}_\alpha$ and $\omega_\nu \geq 0$ is the weight attributed to the neighbor ν. The contracted Voronoi cell $\mathbb{V}_\alpha^{\lambda_{\mathbb{V}}}$ and the contracted working region $\mathscr{W}_\alpha^{\lambda_{\mathscr{W}}}$ are described in the following. The set $\Lambda \subset \Sigma$ is the set of all the outgoing agents and $\mathbf{x}_\alpha^w$ is the projection on $\mathscr{W}$ of the state-space vector of the agent $\alpha \in \Sigma$.

Contracted Voronoi cell. The set $\mathbb{V}_\alpha^{\lambda_{\mathbb{V}}}$ is a contracted Voronoi cell such that $\mathbb{V}_\alpha^{\lambda_{\mathbb{V}}} = \mathbf{c}_\alpha^c \oplus \lambda_{\mathbb{V}}\left(\mathbb{V}_\alpha \oplus \left\{-\mathbf{c}_\alpha^c\right\}\right)$, where $\mathbf{c}_\alpha^c$ is the Chebyshev center of $\mathbb{V}_\alpha$ and $\lambda_{\mathbb{V}} \in [0, 1)$ is the contraction factor. The $\mathscr{H}$-representation of the contracted cell is then $\mathbb{V}_\alpha^{\lambda_{\mathbb{V}}} = \left\{\mathbf{x} \in \mathbb{R}^2 \middle| \boldsymbol{H}_{\mathbb{V}_\alpha}\mathbf{x} \leq \lambda_{\mathbb{V}}\boldsymbol{\Theta}_{\mathbb{V}_\alpha} + (1 - \lambda_{\mathbb{V}})\boldsymbol{H}_{\mathbb{V}_\alpha}\mathbf{c}_\alpha^c\right\}$, with $\boldsymbol{H}_{\mathbb{V}_\alpha}$ and $\boldsymbol{\Theta}_{\mathbb{V}_\alpha}$ such that $\mathbb{V}_\alpha = \left\{\mathbf{x} \in \mathbb{R}^2 \middle| \boldsymbol{H}_{\mathbb{V}_\alpha}\mathbf{x} \leq \boldsymbol{\Theta}_{\mathbb{V}_\alpha}\right\}$.

Example 3 (Construction of a contracted Voronoi cell) Figure 4 illustrates the construction of the contracted version of the Voronoi cell $\mathbb{V}_2$ (in blue) with a contraction factor $\lambda_{\mathbb{V}} = 0.5$. The Chebyshev center of $\mathbb{V}_2$ is $\mathbf{c}_2^c$ and the contracted Voronoi cell centered on $\mathbf{c}_2^c$ is $\mathbb{V}_2^{0.5}$ (in red). □

Contracted working region. The workspace $\mathscr{W}$ is such that $\mathscr{W} = \left\{\mathbf{x} \in \mathbb{R}^2 \middle| \boldsymbol{H}_{\mathscr{W}}\mathbf{x} \leq \boldsymbol{\Theta}_{\mathscr{W}}\right\}$, where $\boldsymbol{H}_{\mathscr{W}} \in \mathbb{R}^{s\times 2}$ and

$\mathbf{\Theta}_{\mathscr{W}} \in \mathbb{R}^s$, with s the number of sides of $\mathscr{W}$. Each hyperplane defined by the $|\Lambda|$ outgoing agents defines two half-spaces. An agent $\alpha \in \Sigma_{\mathbb{W}}$ then belongs to the intersection of $|\Lambda|$ convex half-spaces $\mathscr{S}_\alpha = \left\{\mathbf{x} \in \mathbb{R}^2 \middle| \boldsymbol{H}_{\mathscr{S}_\alpha}\mathbf{x} \leq \mathbf{\Theta}_{\mathscr{S}_\alpha}\right\}$, where $\boldsymbol{H}_{\mathscr{S}_\alpha} \in \mathbb{R}^{|\Lambda|\times 2}$ and $\mathbf{\Theta}_{\mathscr{S}_\alpha} \in \mathbb{R}^{|\Lambda|}$ such that:

$$\mathscr{S}_\alpha = \left\{\mathbf{x} \in \mathbb{R}^2 \middle| \left(\mathbf{x} - \mathbf{x}_l^w\right)^\top \left(\mathbf{x}_\alpha^w - \mathbf{x}_l^w\right) > 0, \forall l \in \Lambda\right\}$$

where $\mathbf{x}_l^w$ is the position of the outgoing agent $l \in \Lambda$. The set $\mathscr{S}_\alpha$ is computed as the intersection of convex sets, thus $\mathscr{S}_\alpha$ is a convex set [10]. Notice that $\mathscr{S}_\alpha$ can be unbounded. The *working region* of agent α, denoted by $\mathscr{W}_\alpha$ is the intersection of $\mathscr{S}_\alpha$ with the workspace $\mathscr{W}$. The working region $\mathscr{W}_\alpha$ is then also a convex set. The Chebyshev center of the working region $\mathscr{W}_\alpha$ is denoted by $\mathbf{c}^c_{\mathscr{W}_\alpha}$. The set $\mathscr{S}_\alpha$ is contracted to $\mathscr{S}^c_\alpha$ in the same way as the Voronoi cell:

$$\mathscr{S}^c_\alpha = \left\{\mathbf{x} \in \mathbb{R}^2 \;\middle|\; \boldsymbol{H}_{\mathscr{S}_\alpha}\mathbf{x} \leq \underbrace{\lambda_{\mathscr{W}}\mathbf{\Theta}_{\mathscr{S}_\alpha} + (1-\lambda_{\mathscr{W}})\,\boldsymbol{H}_{\mathscr{S}_\alpha}\mathbf{c}^c_{\mathscr{W}_\alpha}}_{\mathbf{\Theta}_{\mathscr{S}^c_\alpha}}\right\}.$$

A non-minimal $\mathscr{H}$-representation of the contracted working region $\mathscr{W}_\alpha^{\lambda_{\mathscr{W}}}$ of agent α is then:

$$\mathscr{W}_\alpha^{\lambda_{\mathscr{W}}} = \left\{\mathbf{x} \in \mathbb{R}^2 \middle| \begin{pmatrix}\boldsymbol{H}_{\mathscr{W}} \\ \boldsymbol{H}_{\mathscr{S}_\alpha}\end{pmatrix}\mathbf{x} \leq \begin{pmatrix}\mathbf{\Theta}_{\mathscr{W}} \\ \mathbf{\Theta}_{\mathscr{S}^c_\alpha}\end{pmatrix}\right\}. \tag{15}$$

Example 4 (Construction of a contracted working region) Figure 5 illustrates the construction of *the contracted working region* $\mathscr{W}_2^{0.5}$ for agent 2 with a contraction factor $\lambda_{\mathscr{W}} = 0.5$ when the agents 6 and 7 leave the workspace. The objective points $\mathbf{x}_6^{\text{ref}}$ and $\mathbf{x}_7^{\text{ref}}$ that are used to define the hyperplanes $\mathscr{H}_6$ and $\mathscr{H}_7$ lie on those planes outside the scope of the figure. The unbounded set $\mathscr{S}_2$ corresponds to the lightly grayed space between the two hyperplanes $\mathscr{H}_6$ and $\mathscr{H}_7$. The *working region* $\mathscr{W}_2 = \mathscr{S}_2 \cap \mathscr{W}$ (red dotted line) of the agent 2 has $\mathbf{c}^c_{\mathscr{W}_2}$ (in red) for Chebyshev center. Then, the set $\mathscr{S}^c_2$, in dark gray, is computed as the contraction of the set $\mathscr{S}_2$. Finally, *the contracted working region* of the agent 2 with a contraction factor $\lambda_{\mathscr{W}} = 0.5$ is $\mathscr{W}_2^{0.5}$ (blue dotted line) and it is such that $\mathscr{W}_2^{0.5} = \mathscr{S}^c_2 \cap \mathscr{W}$. □

Remark 1 Notice that the set $\mathscr{S}_\alpha$ can be the same for several agents, e.g. in Fig. 5, $\mathscr{S}_1 = \mathscr{S}_2 = \mathscr{S}_5$. This means that these agents share the same working region (i.e. $\mathscr{W}_1 = \mathscr{W}_2 = \mathscr{W}_5$) and contracted working region (i.e. $\mathscr{W}_1^{0.5} = \mathscr{W}_2^{0.5} = \mathscr{W}_5^{0.5}$).

Neighbors' weights. The weights $\omega_\nu \geq 0$, with $\nu \in \mathrm{N}_\alpha$, are assigned depending on the position of the neighbor ν. If $\mathbf{x}_\nu^w \in \mathscr{S}_\alpha$ and the distance between ν and each of the hyperplanes defined by the outgoing agents is greater than the distance between α and these hyperplanes, i.e.:

$$d\left(\mathbf{x}_\nu^w, \mathscr{H}_l\right) \geq d\left(\mathbf{x}_\alpha^w, \mathscr{H}_l\right), \forall l \in \Lambda,$$

then ν is a *heavy neighbor* (inducing an important attractive behavior) and it will receive a weight ω_h. If ν is an *outgoing agent*, it will receive a weight ω_o. Else, ν is a *light neighbor* (inducing a lighter attractive behavior than a heavy neighbor) and it will receive a weight ω_l. The assigned weights are such that $\omega_h > \omega_l > \omega_o$ in order not to cross the path of the outgoing agents. The advantage with respect to [8] is that the outgoing agents do not have the same weight as the light neighbors to induce an even less attractive behavior. The weight ω_o will often be chosen such that $\omega_o = 0$.

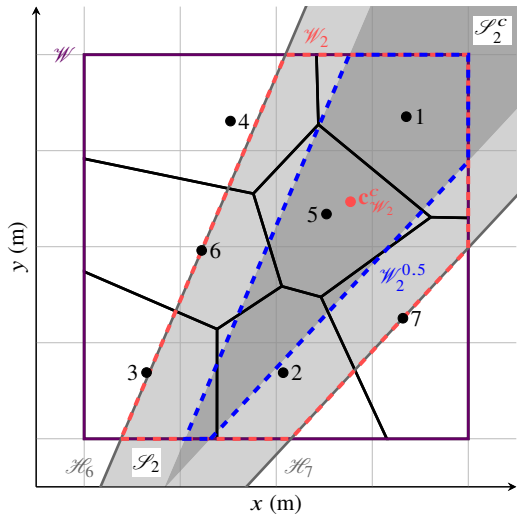

Fig. 5 Example of the construction of the set $\mathscr{W}_2^{0.5} = \mathscr{W}_1^{0.5} = \mathscr{W}_5^{0.5}$.

Example 5 (Classification of the neighbors) In Fig. 6, agents 6 and 10 are the outgoing agents, their trajectories lying on the hyperplane $\mathscr{H}_6$ and $\mathscr{H}_{10}$ (in green). The objective points that are used to define $\mathscr{H}_6$ and $\mathscr{H}_{10}$ lie on these planes outside the figure's scope. For this example, the case of the remaining agents $4, 16, 17 \in \Sigma_{\mathbb{W}} \setminus \{6, 10\}$ is studied[6]. The dashed lines[7] pass by the agent (4, 16 or 17 here) whose neighbors are sorted and are parallel to a hyperplane (either $\mathscr{H}_6$ or $\mathscr{H}_{10}$), i.e. the sets: $\left\{\mathbf{x} \in \mathbb{R}^2 \middle| d\left(\mathbf{x}, \mathscr{H}_l\right) = d\left(\mathbf{x}_\alpha^w, \mathscr{H}_l\right)\right\}$, with $\alpha \in \{4, 16, 17\}$ and $l \in \{6, 10\}$. For the three considered remaining agents, the neighbors lying on the same side of the dashed lines as the outgoing agents are *light neighbors* and receive the weight ω_l, while the others are considered as *heavy neighbors* and receive the weight ω_h. If a neighbor of agent 16 had been in the gray area, it would have received the weight ω_h. None of the neighbors of agent 17 are outgoing agents. Agent 6 is a neighbor of agent 4 and receives the weight ω_o. Agents 6 and 10 are neighbors of agent 16 and receive the weight ω_o. □

Remark 2 If all the outgoing agents follow parallel trajectories, all the neighbors of an agent lying between two of them

[6] These agents are chosen to illustrate different possible cases: agents 4 and 17 have heavy and light neighbors, one with an outgoing neighbor and one without, while agent 16 is the pathological case of an agent with only light neighbors.

[7] The dashed lines in Fig. 6 are piecewise lines made with the leftmost or the rightmost lines.

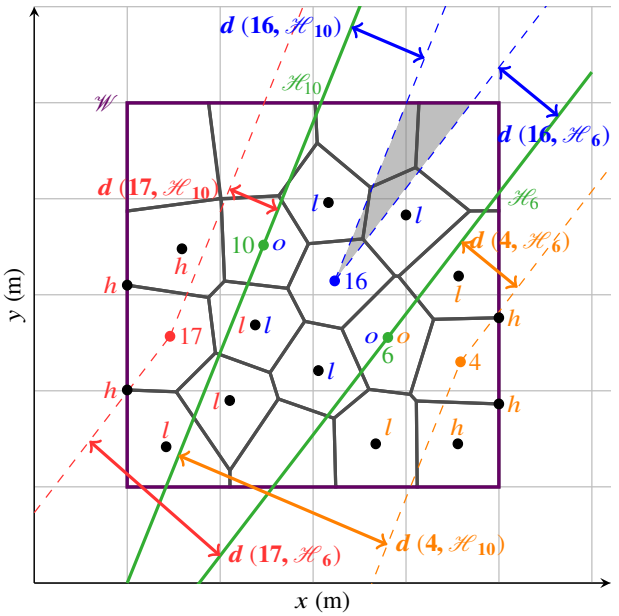

Fig. 6 Classification of the neighbors of agents 4, 16 and 17 when agents 6 and 10 leave the workspace.

will receive a weight ω_l (or ω_o if the neighbor is an outgoing agent).

4.3 New reconfiguration algorithm

In [8], whenever an agent has to leave the workspace, the remaining agents would track their neighbors' barycenter. Here, in order to limit the movement of all the remaining agents and thus the energy consumption, a new control strategy is proposed for them.

Each remaining agent needs to know the outgoing agents with a possible collision risk in order to compute its working region and its new objective point. The set of the outgoing agents considered by agent $\alpha \in \Sigma_{\mathbb{W}}$ is denoted by O_α. Agent α will then track the reference given by (14) as long as O_α will contain at least one outgoing agent. When O_α is empty, the objective of agent α is the Chebyshev center $\mathbf{c}_\alpha^c$ of its Voronoi cell.

The set O_α, with $\alpha \in \Sigma_{\mathbb{W}}$, contains all the outgoing agents seen by the agent α. As described in the previous section, whenever an agent $l \in \Sigma$ leaves the formation, it has for objective a point outside the workspace defining a hyperplane $\mathscr{H}_l$. The orthogonal plane to $\mathscr{H}_l$ passing by the outgoing agent is denoted by $\mathscr{O}_l$. The plane $\mathscr{O}_l$ defines two half-spaces, $\mathscr{X}_o^l$ and $\mathscr{X}_r^l$, where $\mathscr{X}_o^l$ contains the objective point of the outgoing agent, i.e.:

$$\mathscr{X}_o^l = \left\{ \mathbf{x} \in \mathbb{R}^2 \middle| \left(\mathbf{x}_l^{\mathrm{ref}} - \mathbf{x}_l^w\right)^\top \left(\mathbf{x} - \mathbf{x}_l^w\right) > 0 \right\}$$

$$\mathscr{X}_r^l = \left\{ \mathbf{x} \in \mathbb{R}^2 \middle| \left(\mathbf{x}_l^{\mathrm{ref}} - \mathbf{x}_l^w\right)^\top \left(\mathbf{x} - \mathbf{x}_l^w\right) < 0 \right\}.$$

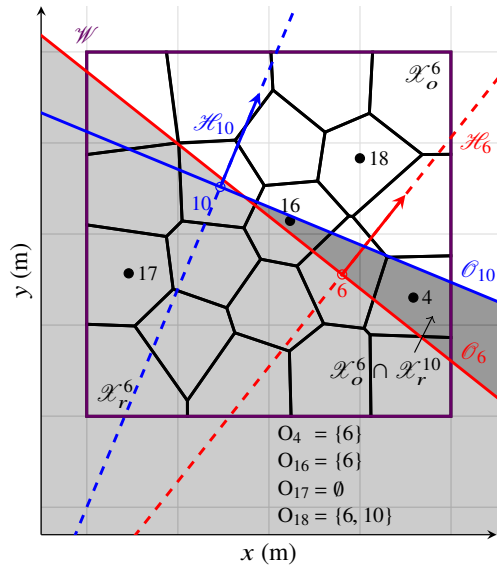

Fig. 7 Construction of the sets O_α with $\alpha \in \{4, 16, 17, 18\}$ when agents 6 and 10 leave the workspace.

If a remaining agent $\alpha \in \Sigma_{\mathbb{W}}$ belongs to $\mathscr{X}_o^l$, the outgoing agent l is added to O_α. When α switches from $\mathscr{X}_o^l$ to $\mathscr{X}_r^l$, the outgoing agent is removed from O_α. If a remaining agent $\alpha \in \Sigma$ belongs to $\mathscr{X}_r^l$, it will be said to be "behind" the outgoing agent l. This procedure is described in Algorithm 1. In the case of several outgoing agents, this procedure is applied to all the remaining agents for each outgoing agent.

Algorithm 1: Construction of the set O_α of the outgoing agents considered by the agent α

Input: The positions $\mathbf{x}_l^w$ and objectives $\mathbf{x}_l^{\mathrm{ref}}$ of the outgoing agents, with $l \in \Lambda$, the position $\mathbf{x}_\alpha^w$ and the previous set O_α of the agent $\alpha \in \Sigma_{\mathbb{W}} \setminus \Lambda$
Output: The updated set O_α of the agent $\alpha \in \Sigma_{\mathbb{W}} \setminus \Lambda$
for $l \in \Lambda$ **do**
  if $\left(\mathbf{x}_l^{ref} - \mathbf{x}_l^w\right)^\top \left(\mathbf{x}_\alpha^w - \mathbf{x}_l^w\right) > 0$ **and** $l \notin \mathrm{O}_\alpha$ **then**
    $\mathrm{O}_\alpha \leftarrow \mathrm{O}_\alpha \cup \{l\}$;
  else if $\left(\mathbf{x}_l^{ref} - \mathbf{x}_l^w\right)^\top \left(\mathbf{x}_\alpha^w - \mathbf{x}_l^w\right) < 0$ **and** $l \in \mathrm{O}_\alpha$ **then**
    $\mathrm{O}_\alpha \leftarrow \mathrm{O}_\alpha \setminus \{l\}$;
  end if
end for

Example 6 (Constructions of the sets O_α*)* In Fig. 7, agents 6 and 10 are leaving the formation. The planes $\mathscr{H}_6$ and $\mathscr{H}_{10}$ are the dashed lines passing by agents 6 and 10. The solid lines orthogonal to $\mathscr{H}_6$ and $\mathscr{H}_{10}$ are the hyperplanes $\mathscr{O}_6$ and $\mathscr{O}_{10}$. The direction of the movement of agents 6 and 10 is indicated by the corresponding arrows along $\mathscr{H}_6$ and $\mathscr{H}_{10}$, allowing one to define the half-spaces $\mathscr{X}_o^6$ and $\mathscr{X}_o^{10}$ containing the objective points of agents 6 and 10, and their complements $\mathscr{X}_r^6$ and $\mathscr{X}_r^{10}$. In Fig. 7 the set $\mathscr{X}_o^6$ (in white), the set $\mathscr{X}_r^6$ (in

light gray) and the intersection $\mathscr{X}_o^6 \cap \mathscr{X}_o^{10}$ (in dark gray) are presented. Agent 17 belongs to $\mathscr{X}_r^6 \cap \mathscr{X}_r^{10}$ since $\forall l \in \{6, 10\}$, the inequality $\left(\mathbf{x}_l^{\text{ref}} - \mathbf{x}_l^w\right)^\top \left(\mathbf{x}_{17} - \mathbf{x}_l^w\right) < 0$ holds. Thus, it does not consider the agents 6 and 10 as outgoing agents for the definition of its *working region*, i.e. $\mathrm{O}_{17} = \emptyset$. The working region $\mathscr{W}_{17}$ of agent 17 is not defined. Agent 17 will then track its Chebyshev center. However, agents 4 and 16 both belong to $\mathscr{X}_o^6 \cap \mathscr{X}_r^{10}$. Thus, agents 4 and 16 consider only the agent 6 as an outgoing agent for the construction of its working region and $\mathrm{O}_4 = \mathrm{O}_{16} = \{6\}$. The working regions $\mathscr{W}_4$ and $\mathscr{W}_{16}$ are defined by considering that agent 6 is the only outgoing agent. Finally, agent 18 belongs to $\mathscr{X}_o^6 \cap \mathscr{X}_o^{10}$ thus $\mathscr{W}_{18}$ is defined by considering that agents 6 and 10 are outgoing agents, i.e. $\mathrm{O}_{18} = \{6, 10\}$. □

For an agent $\alpha \in \Sigma_{\mathbb{W}}$, the working region $\mathscr{W}_\alpha$ defined in the previous section is obtained by considering only the outgoing agents belonging to O_α. If O_α is empty, the working region is not defined and the remaining agent α tracks its Chebyshev center. The control input for such an agent α is then computed as presented in the previous sections. Algorithm 2 summarizes the procedure applied at each time instant for each cooperating agent remaining in the formation.

Algorithm 2: Computation of a cooperating agent's input based on the new reconfiguration strategy

Input: The positions of the neighbors $\mathbf{x}_\nu^w$, with $\nu \in \mathrm{N}_\alpha$, the positions of the outgoing agents $\mathbf{x}_l^w$, with $l \in \Lambda$, the position $\mathbf{x}_\alpha^w$ and the previous set O_α of the agent $\alpha \in \Sigma_{\mathbb{W}} \setminus \Lambda$
Output: The control input $\widetilde{\mathbf{u}}_\alpha$ for the agent $\alpha \in \Sigma_{\mathbb{W}} \setminus \Lambda$
begin
  Compute the Voronoi cell $\mathbb{V}_\alpha$ of the agent α;
  Call Algorithm 1 to update the set O_α;
  if $\mathrm{O}_\alpha \neq \emptyset$ **then**
    Compute the contracted working region $\mathscr{W}_\alpha^{\lambda_{\mathbb{W}}}$;
    Compute the contracted Voronoi cell $\mathbb{V}_\alpha^{\lambda_{\mathbb{V}}}$;
    for $\nu \in \mathrm{N}_\alpha$ **do**
      if $\nu \in \mathrm{O}_\alpha$ **then**
        $\omega_\nu \leftarrow \omega_o$
      else if $\forall l \in \mathrm{O}_\alpha$, $d\left(\mathbf{x}_\nu^w, \mathscr{H}_l\right) > d\left(\mathbf{x}_\alpha^w, \mathscr{H}_l\right)$ **and** $\left(\mathbf{x}_\nu^w - \mathbf{x}_l^w\right)^\top \left(\mathbf{x}_\alpha^w - \mathbf{x}_l^w\right) > 0$ **then**
        $\omega_\nu \leftarrow \omega_h$
      else
        $\omega_\nu \leftarrow \omega_l$
      end if
    end for
    Compute $\mathbf{c}_\alpha^{\min}$ by solving (14);
    $\widetilde{\mathbf{x}}_\alpha^{\text{obj}} \leftarrow \left[\mathbf{c}_\alpha^{\min\top}\ \mathbf{0}_{1\times 10}\right]^\top$;
  else
    Compute the Chebyshev center $\mathbf{c}_\alpha^c$ of $\mathbb{V}_\alpha$ by solving (9);
    $\widetilde{\mathbf{x}}_\alpha^{\text{obj}} \leftarrow \left[\mathbf{c}_\alpha^{c\top}\ \mathbf{0}_{1\times 10}\right]^\top$;
  end if
  Compute $\widetilde{\mathbf{u}}_\alpha$ by solving (12);
end

5 Simulations and Results

5.1 Simulation scenarios

This paragraph describes the two simulation scenarios illustrating the proposed approach.

First scenario. The multi-agent system Σ is composed of twenty[8] agents. They start at a hovering state at random locations in a given bounded polygon $\mathscr{W}$ such that $\mathbf{x}_\alpha^{\text{init}} = \left[x_\alpha\ y_\alpha\ \bar{z}\ \mathbf{0}_{1\times 9}\right]^\top$, with $\alpha \in \Sigma_{\mathbb{W}}$. At $t = 0$ s, they start tracking their Chebyshev center $\mathbf{x}_\alpha^c = \left[\mathbf{c}_\alpha^{c\top}\ \bar{z}\ \mathbf{0}_{1\times 9}\right]^\top$. At $t = 12$ s, three agents have to leave the workspace $\mathbb{W}$ because they are required for another mission. The agents in the formation then follow the strategy described in Section 4.3 until the outgoing agents are outside $\mathbb{W}$.

Second scenario. The same twenty agents are deployed inside the workspace $\mathbb{W}$ with the same initial conditions as in the first scenario. Once they reach a static Chebyshev configuration, two agents are subject to an actuator fault consisting in a loss of efficiency of 40% on one of their rotors. These agents then leave the workspace $\mathbb{W}$ while the remaining ones follow the strategy from Section 4.3.

5.2 Simulation parameters

The control algorithm for deployment and reconfiguration of a multi-agent system presented in this paper is tested in simulation with MATLAB. The optimization solvers for the problems (12) and (14) are generated with CVXGEN [17]. This tool generates a custom solver for a given optimization problem with a quadratic cost function and linear constraints. This solver is easily portable to any platform since the C sources of the program are provided. The toolbox MPT3.0 [13] is used to compute all the geometric elements. The simulations are run on a desktop PC with Intel Core i7-4790S (4×3.2 GHz) processor and 16 GB DDR3 RAM. The mean total run time of the solver for problem (12) is 1.1 ms. The mean total run time of the solver for problem (14) is 75 μs. The full simulation's mean total run time is 83.486 s.

The agents in Σ obey the nonlinear continuous-time model from Section 2.2. They evolve in the state-space $\mathbb{X}$ with the constraints from (6) and (7). The plane $\mathscr{W}$ (i.e. the projection of $\mathbb{W}$ on the first two directions of the state-space) has for $\mathscr{H}$-representation:

$$\mathscr{W} = \left\{ \mathbf{x} \in \mathbb{R}^2 \middle| \begin{bmatrix} -0.0566 & -0.0389 \\ 0.0672 & -0.1093 \\ 0.0528 & 0.0238 \\ -0.0165 & 0.0521 \\ -0.0492 & 0.0316 \end{bmatrix} \mathbf{x} \leq \begin{bmatrix} 0.9976 \\ 0.9917 \\ 0.9983 \\ 0.9985 \\ 0.9983 \end{bmatrix} \right\}.$$

[8] The proposed decentralized deployment algorithm has been simulated for 100 agents.

Tuning parameters. The main objective of the agents is to reach their objective point while remaining at constant altitude with a null yaw angle regardless of the speed. Thus $\boldsymbol{Q}$ is chosen to be $\boldsymbol{Q} = \text{diag}\,(10, 10, 100, 0.1, 0.1, 100, 0.1, 0.1, 100, 0.1, 0.1, 100)$. The weights $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are set to $\boldsymbol{R}_1 = 0.1\boldsymbol{I}_4$ and $\boldsymbol{R}_2 = 100\boldsymbol{R}_1$ to follow these objectives while keeping the input signal $\widetilde{u}_i^\alpha$ such that $\left|\widetilde{u}_i^\alpha\right| \leq 0.15$, with $i \in \{1, \ldots, 4\}$ and $\alpha \in \Sigma$. These weights are the same for the agents in formation and the outgoing agents. If the contraction factor λ from Section 4.1 is too small, the terminal set might not be reached over the prediction horizon. It is then set to $\lambda = 0.5$ for the problem to be feasible. If the contraction factors $\lambda_{\mathbb{V}}$ and $\lambda_{\mathscr{W}}$ from Section 4.2 are too small, the contracted working region might be empty. They are then set to $\lambda_{\mathbb{V}} = 0.5$ and $\lambda_{\mathscr{W}} = 0.8$. The weights for the heavy/light/outgoing neighbors from Section 4.2 are set to $\omega_h = 100$, $\omega_l = 10$ and $\omega_o = 0$. With such weights, even with an important number of light neighbors, the heavy neighbors will still be dominating and the outgoing agents will not have any attractive effect over the remaining agents. The prediction horizon is chosen to be $N_p = 10$.

Control input. The control action is separated into three elements $\mathbf{u}_\alpha(k) = \widetilde{\mathbf{u}}_\alpha(k) + \bar{\mathbf{u}}_\alpha + \mathbf{u}_\alpha^z(k)$. The signal $\widetilde{\mathbf{u}}_\alpha(k)$ is the input signal provided by solving the optimization problem (12). The signal $\bar{\mathbf{u}}_\alpha = \bar{u}_\alpha \mathbf{1}_{4\times1}$ is the value of the input signal when the UAV is at rest at a hovering condition. Its value is given in Section 2.3. Finally, the signal $\mathbf{u}_\alpha^z = u_\alpha^z \mathbf{1}_{4\times1}$ is meant to compensate the loss of upward thrust when the UAV tilts. The upward thrust applied to the UAV by one rotor is:

$$\mathbf{t}_i(k) = \frac{1}{4} mg \cos\theta(k) \cos\phi(k) \mathbf{z}_{\text{Earth}}, \text{ with } i \in \{1, \ldots, 4\}\,.$$

The signal u_α^z is then such that $\mathbf{t}_i(k) + K u_\alpha^z(k) \mathbf{z}_{\text{Earth}} = \frac{1}{4} mg \mathbf{z}_{\text{Earth}}$ or $u_\alpha^z(k) = \frac{mg}{4K}\left(1 - \cos\theta(k)\cos\phi(k)\right)$. With such a control input, the variation in altitude is less than 10 cm which justifies two-dimensional illustrations.

Rate limitation. Finally, when an agent has to leave the workspace, a rate limitation is imposed on the variation of its objective. The variation rate per second is equal to a fourth of the difference between the objective and the position of the agent when it starts to leave the workspace.

5.3 Analysis of the first scenario

Twenty agents, represented by the black dots in Fig. 8, start from random positions in the bounded polygon $\mathscr{W}$, with an initial state-space vector $\mathbf{x}_\alpha^{\text{init}} = \left[x_\alpha \; y_\alpha \; \bar{z} \; \mathbf{0}_{1\times9}\right]^\top$, $\alpha \in \Sigma_{\mathbb{W}}$. At $t = 0$ s, they start tracking the Chebyshev center of their associated Voronoi cell, represented by asterisks in Fig. 8. During the first part of the simulation, between $t = 0$ s and $t = 12$ s, the agents are deploying inside the workspace $\mathbb{W}$.

At $t = 12$ s, agents 1, 12 and 19 (as numbered in Fig. 9) have to leave the workspace towards the points $\mathbf{x}_1^{\text{ref}} = \left[20 \; 40 \; \bar{z} \; \mathbf{0}_{1\times10}\right]^\top$, $\mathbf{x}_{12}^{\text{ref}} = \left[10 \; 40 \; \bar{z} \; \mathbf{0}_{1\times10}\right]^\top$ and $\mathbf{x}_{19}^{\text{ref}} = \left[15 \; 40 \; \bar{z} \; \mathbf{0}_{1\times10}\right]^\top$.

The goal of Fig. 9 is to present the construction of the objects necessary to compute the objective point of the agent 10 at $t = 12$ s. In Fig. 9, the half-lines with an arrow passing by agents 1, 12 and 19 are part of $\mathscr{H}_1$, $\mathscr{H}_{12}$ and $\mathscr{H}_{19}$ indicating the trajectories of these agents. The lines orthogonal to these half-lines are $\mathscr{O}_1$, $\mathscr{O}_{12}$ and $\mathscr{O}_{19}$ used for the construction of the sets O_α of the outgoing agents considered by the agent α, with $\alpha \in \Sigma_{\mathbb{W}}$, as described in Section 4.3. The dashed polygons are the contracted working region $\mathscr{W}_{10}^{0.8}$ (in red) and the contracted Voronoi cell $\mathbb{V}_{10}^{0.5}$ (in black). The set O_{10} of the outgoing agents considered by the agent 10, as defined in Section 4.3, is such that $\mathrm{O}_{10} = \{12, 19\}$ since the agent 10 is "behind" agent 1 as illustrated in the figure. The dashed lines passing by agent 10 (in black) are used to classify its neighbors in light/heavy/outgoing, based on the process described in Section 4.2. The objective points of all the remaining agents are also presented in Fig. 9, represented by black diamonds, except for agent 3 since it is "behind" all the outgoing agents and tracks its Chebyshev center.

In Fig. 10, the distances between four agents from $\Sigma_{\mathbb{W}}$ and their objectives are presented. During the first part, from $t = 0$ s to $t = 12$ s, the agents are deployed inside the workspace $\mathbb{W}$. At $t = 12$ s, the objective point of agents 5, 10 and 11 are changed to $\mathbf{x}_5^{\min}$, $\mathbf{x}_{10}^{\min}$ and $\mathbf{x}_{11}^{\min}$, while agent 3 keeps tracking its Chebyshev center. Then, each time an agent passes "behind" an outgoing agent (when the outgoing agent is removed from O_α), there is an important variation of the distance since the classification of the agent's neighbors

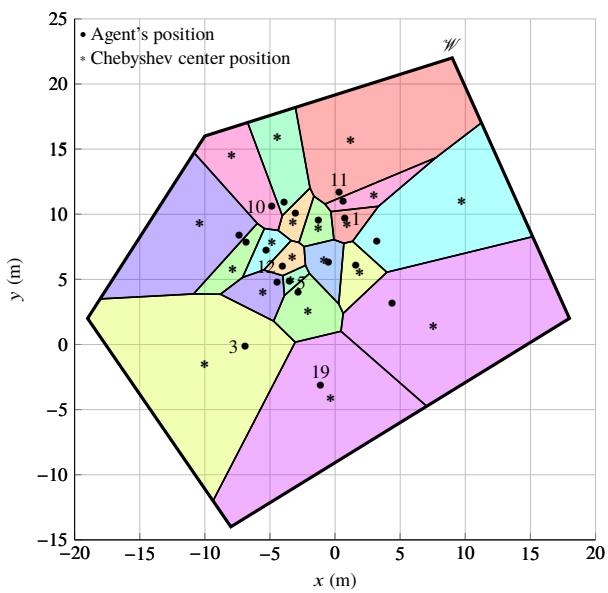

Fig. 8 Starting position of the multi-agent system in $\mathscr{W}$.

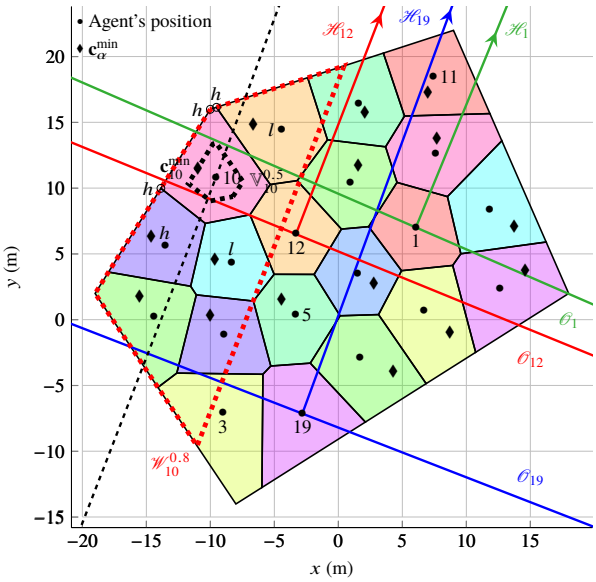

Fig. 9 Position of the agents and their objectives at $t = 12$ s and construction of $\mathbf{c}_{10}^{\min}$.

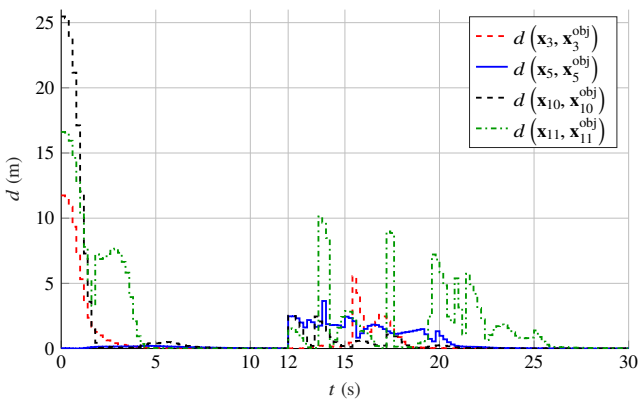

Fig. 10 Distances between the agents and their objectives for the first scenario.

is changed. For instance, at $t = 13.6$ s, $t = 17.2$ s and $t = 19.6$ s, the agent 11 passes behind the agents 1, 12 and 19, respectively, inducing a variation in the distance (in green). In the end, when all the outgoing agents have left $\mathbb{W}$, all the remaining agents converge towards their Chebyshev centers. The variations that can be observed for agent 3 (in red) around $t = 16$ s are due to the important movements of its neighbors, leading to an important modification of its Voronoi cell and thus of its Chebyshev center.

Finally, Fig. 11 presents the variations of the four components of agents 11's input signal $\widetilde{\mathbf{u}}_{11}$ and $\bar{u}_{11} + u_{11}^z$. The signal provided by the MPC remains inside the constraints. The amplitude of the oscillations is large in order to respect the constraints on the pitch and roll angles. For this example, it was chosen to present only the input signals for agent 11 because it is the worst case scenario, this agent having a small working region for a long time.

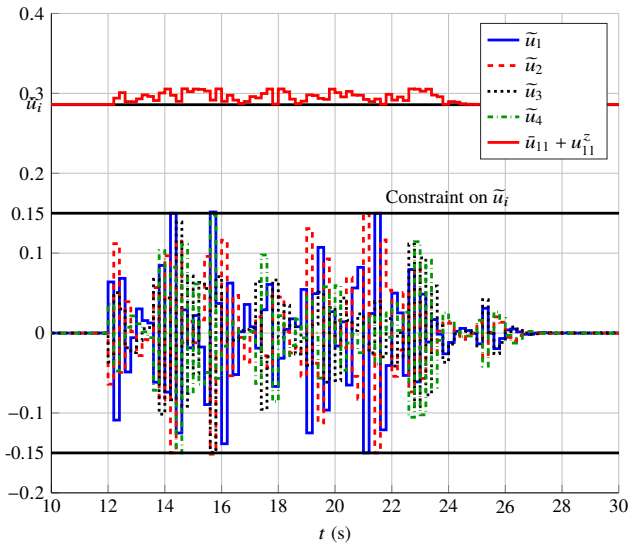

Fig. 11 Control inputs of agent 11 for the first scenario when agents 1, 12 and 19 leave the workspace.

A video showing the simulation is available on the Laboratoire des Signaux et Systèmes' YouTube channel at `https://youtu.be/nT2gtQQJQfM`.

5.4 Analysis of the second scenario

The twenty agents start from the positions shown in Fig. 8. They start tracking the Chebyshev center of their Voronoi cell at $t = 0$ s. They are deploying inside the workspace $\mathbb{W}$ between $t = 0$ s and $t = 5$ s. However, at $t = 5$ s, agents 1 and 7 are subject to an actuator fault. Both these agents undergo a loss of efficiency of 40% on one of their rotor. The loss is on rotor 1 (as numbered in Fig. 1) for agent 1 and on rotor 3 for agent 7. The matrices $\boldsymbol{G}_1$ and $\boldsymbol{G}_7$ from (11) are then modified to become $\boldsymbol{G}_1^f = \operatorname{diag}(0.6, 1, 1, 1)\,\boldsymbol{G}_1$ and $\boldsymbol{G}_7^f = \operatorname{diag}(1, 1, 0.6, 1)\,\boldsymbol{G}_7$.

Given the type of fault they are subject to, the agents will leave $\mathscr{W}$ by following a trajectory that does not request the use of the faulty rotor. These points are $\mathbf{x}_1^{\text{ref}} = \left[40\ y_1(t = 5\text{ s})\ \bar{z}\ \mathbf{0}_{1\times 9}\right]^\top$ and $\mathbf{x}_7^{\text{ref}} = \left[x_7(t = 5\text{ s})\ 40\ \bar{z}\ \mathbf{0}_{1\times 9}\right]^\top$. Moreover, the input signals $\bar{\mathbf{u}}_1 + \mathbf{u}_1^z$ and $\bar{\mathbf{u}}_7 + \mathbf{u}_7^z$ are modified to mitigate the loss of efficiency by being respectively multiplied by $\operatorname{diag}\left(0.6^{-1}, 1, 1, 1\right)$ and $\operatorname{diag}\left(1, 1, 0.6^{-1}, 1\right)$. The constraint on the input corresponding to the faulty rotor (u_1 for agent 1 and u_3 for agent 7) is also multiplied by 0.6^{-1}.

The objective of Fig. 12 is to present the construction of the objects necessary to compute the objective point of the agent 6 at $t = 5$ s. In Fig. 12, the half-lines with an arrow passing by agents 1 and 7 are a part of $\mathscr{H}_1$ and $\mathscr{H}_7$ indicating their trajectories. The lines orthogonal to these half-lines are $\mathscr{O}_1$ and $\mathscr{O}_7$ used for the construction of the

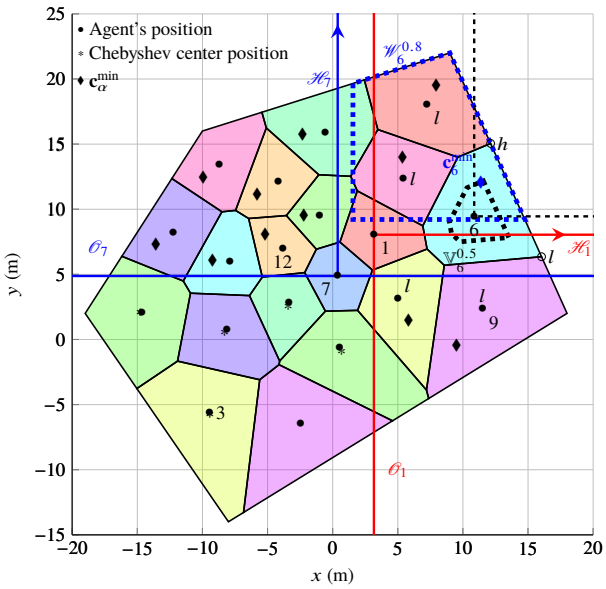

Fig. 12 Position of the agents and their objectives at $t = 5$ s and construction of $\mathbf{c}_6^{\min}$.

sets O_α of the outgoing agents considered by the agent α, with $\alpha \in \Sigma_{\mathbb{W}}$ described in Section 4.3. The dashed polygons are the contracted working region $\mathscr{W}_6^{0.8}$ (in blue) and the contracted Voronoi cell $\mathbb{V}_6^{0.5}$ (in black). The set O_6, as defined in Section 4.3, is such that $\mathrm{O}_6 = \{1, 7\}$. The dashed lines passing by agent 6 (in black) are used to classify its neighbors in light/heavy/outgoing, based on the process described in Section 4.2. The objective points of all the remaining agents are also presented in Fig. 9, represented by black diamonds, except for the agents in the bottom left quarter of the polygon which are "behind" the outgoing agents and keep tracking their Chebyshev centers represented by asterisks.

The distances between four agents from $\Sigma_{\mathbb{W}}$ and their objectives are presented in Fig. 13. As in the previous scenario, the first part of the movement consists in the deployment of the agents inside $\mathbb{W}$. Then, at $t = 5$ s, the objective point of agents 6, 9 and 12 is changed to $\mathbf{x}_6^{\min}$, $\mathbf{x}_9^{\min}$ and $\mathbf{x}_{12}^{\min}$, while the objective of agent 3 remains $\mathbf{x}_3^c$. The behavior of the agents is then similar to the behavior that was observed with the previous scenario. The behavior of agent 3's neighbors do not cause an important change in its Chebyshev center, leading to the distance between this agent and its objective to be close to 0 from $t = 4$ s onward.

Finally, Fig. 14 presents the variations of the four components of agent 7's input signal $\widetilde{\mathbf{u}}_7$ and the first and third components of $\bar{\mathbf{u}}_7 + \mathbf{u}_7^z$. The signal provided by the MPC remains inside the constraints. The constraint on $\widetilde{u}_3$ is not shown on this figure since the faulty rotor is not in demand and the variations of its input are really small. The behavior of the other faulty agent is identical, the variations of $\widetilde{u}_1$ and $\widetilde{u}_2$ being replaced by variations on $\widetilde{u}_3$ and $\widetilde{u}_4$. The behavior of the agents of $\Sigma_{\mathbb{W}}$ is the same as in the previous scenario.

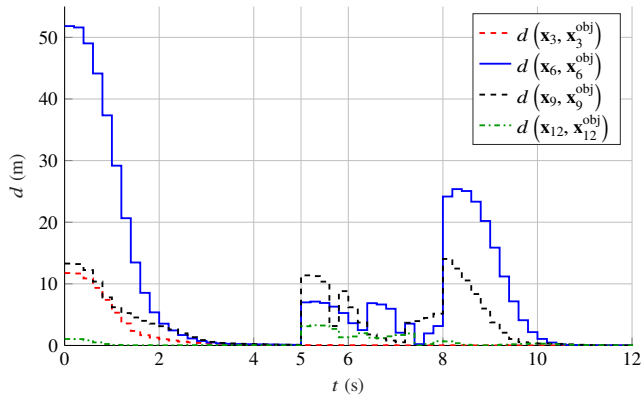

Fig. 13 Distances between the agents and their objectives for the second scenario with an actuator fault occurring on the agents 1 and 7 at $t = 5$ s.

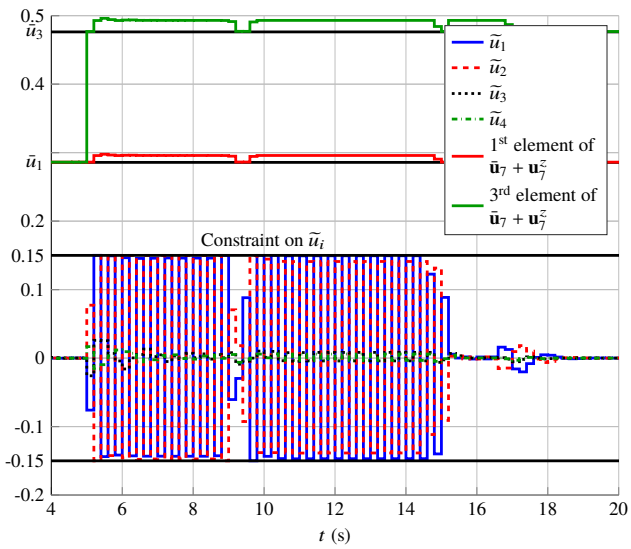

Fig. 14 Control inputs of agent 7 for the second scenario when the agent leaves the workspace with an actuator fault occurring at $t = 5$ s.

A video showing the simulation results is available on the Laboratoire des Signaux et Systèmes' YouTube channel at `https://youtu.be/nT2gtQQJQfM`.

6 Conclusions and Future Works

This paper presents a reconfiguration technique for a multi-agent system deployed over a two-dimensional convex bounded polygonal workspace. In [8], the case of one agent leaving the workspace is studied and it is generalized in this paper to the case of several agents leaving the workspace at the same time. The proposed technique is based on decentralized linear model predictive control and dynamic Voronoi tessellation. The agents remaining inside the workspace tracking their Chebyshev center change their objective to avoid potential collisions with the outgoing agents. The positions of these outgoing agents and their objectives partition the plane into several working regions for the remaining

agents. The new objective for each remaining agent is then a point belonging to the intersection of the agent's contracted working region and its contracted Voronoi cell. These new objective points are then computed at each time instant, while the non-cooperating agents leave the workspace. When these agents are outside the bounded area, the remaining agents resume tracking their Chebyshev center. Using this technique, non-cooperating agents are able to leave the aforementioned workspace while avoiding collisions with agents remaining inside of it. Two simulation scenarios are presented to test the reconfiguration of the multi-agent system with different types of trajectories for the outgoing agents.

As future works, the control algorithm will be applied to an experimental testbed of UAVs available at the Networked Autonomous Vehicles Lab at Concordia University. In order to improve the resilience of the control algorithm presented in this paper, other types of faults will be considered such as position sensor faults which makes the Voronoi tessellation uncertain. Future works will also consider an extension of the proposed results with respect to (variable) communication delays between the agents. Moreover, while this algorithm is used as a controller in this paper, it will be modified to be used as a path planning algorithm and combined with other controllers to add resilience to perturbations and noises.

Appendix A UAV Model

The state-space model of the UAV is written by considering the notations from Section 2.2. The state-space model itself is derived from Lagrangian mechanics [5]:

$$\dot{x} = v_x \tag{16}$$

$$\dot{y} = v_y \tag{17}$$

$$\dot{z} = v_z \tag{18}$$

$$\dot{\phi} = \omega_x + \left(\omega_y \sin\phi + \omega_z \cos\phi\right) \tan\theta \tag{19}$$

$$\dot{\theta} = \omega_y \cos\phi - \omega_z \sin\phi \tag{20}$$

$$\dot{\psi} = \omega_y \frac{\sin\phi}{\cos\theta} + \omega_z \frac{\cos\phi}{\cos\theta} \tag{21}$$

$$\dot{v}_x = \frac{f_t}{m} \left(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi\right) \tag{22}$$

$$\dot{v}_y = \frac{f_t}{m} \left(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi\right) \tag{23}$$

$$\dot{v}_z = \frac{f_t}{m} \cos\phi \cos\theta - g \tag{24}$$

$$\dot{\omega}_x = \frac{I_y - I_z}{I_x} \omega_y \omega_z + \frac{\tau_x}{I_x} \tag{25}$$

$$\dot{\omega}_y = \frac{I_z - I_x}{I_y} \omega_x \omega_z + \frac{\tau_y}{I_y} \tag{26}$$

$$\dot{\omega}_z = \frac{I_x - I_y}{I_z} \omega_x \omega_y + \frac{\tau_z}{I_z} \tag{27}$$

where m is the UAV's mass, I_x, I_y and I_z are the moments of inertia with respect to the axes $\mathbf{x}_{\text{UAV}}$, $\mathbf{y}_{\text{UAV}}$ and $\mathbf{z}_{\text{UAV}}$ illustrated in Fig. 1 and g is the gravitational acceleration. The numerical values of all the model parameters are presented in Table 1.

Table 1 Values of UAV model's parameters

Mass	$m = 1.4$ kg
Inertia components	$I_x = I_y = 0.03$ kg $\cdot$ m^2
	$I_z = 0.04$ kg $\cdot$ m^2
Arm's length	$L = 0.2$ m
Thrust to moment ratio	$C = 4$ m
DC motor gain	$K = 12$ N
DC motor mechanical time constant	$\omega_m^{-1} = 5$ ms
Gravitational acceleration	$g = 9.81$ m $\cdot$ s^{-2}

References

1. Abdolhosseini, M., Zhang, Y.M., Rabbath, C.A.: An efficient model predictive control scheme for an unmanned quadrotor helicopter. Journal of Intelligent & Robotic Systems **70**, 27–38 (2013)
2. Beard, R.W., McLain, T.W., Nelson, D.B., Kingston, D., Johanson, D.: Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. Proceedings of the IEEE **94**(7), 1306–1324 (2006)
3. Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E.N.: The explicit linear quadratic regulator for constrained systems. Automatica **38**(1), 3–20 (2002)
4. Blanchini, F., Miani, S.: Set-theoretic methods in control. Springer (2007)
5. Bouabdallah, S., Murrieri, P., Siegwart, R.: Design and control of an indoor micro quadrotor. In: IEEE International Conference on Robotics and Automation, vol. 5, pp. 4393–4398 (2004)
6. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press (2004)
7. Casbeer, D.W., Beard, R.W., McLain, T.W., Li, S.M., Mehra, R.K.: Forest fire monitoring with multiple small UAVs. In: American Control Conference, pp. 3530–3535 (2005)
8. Chevet, T., Stoica Maniu, C., Vlad, C., Zhang, Y.M.: Voronoi-based UAVs formation deployment and reconfiguration using MPC techniques. In: 2018 International Conference on Unmanned Aircraft Systems, pp. 9–14 (2018)
9. Cortes, J., Martinez, S., Karatas, T., Bullo, F.: Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation **20**(2), 243–255 (2004)
10. Eggleston, H.: Convexity. Cambridge University Press (1958)

11. Gu, Y., Seanor, B., Campa, G., Napolitano, M.R., Rowe, L., Gururajan, S., Wan, S.: Design and flight testing evaluation of formation control laws. IEEE Transactions on Control Systems Technology **14**(6), 1105–1112 (2006)
12. Hatleskog, J., Olaru, S., Hovd, M.: Voronoi-based deployment of multi-agent system. In: IEEE 57th Conference on Decision and Control, pp. 5403–5408 (2018)
13. Herceg, M., Kvasnica, M., Jones, C.N., Morari, M.: Multi-parametric toolbox 3.0. In: European Control Conference, pp. 502–510 (2013)
14. Lai, L.C., Yang, C.C., Wu, C.J.: Time-optimal control of a hovering quad-rotor helicopter. Journal of Intelligent & Robotic Systems **45**, 115–135 (2006)
15. Laliberte, A.S., Rango, A.: Texture and scale in object-based analysis of subdecimeter resolution unmanned aerial vehicle (UAV) imagery. IEEE Transactions on Geoscience and Remote Sensing **47**(3), 761–770 (2009)
16. Lozano, R.: Unmanned aerial vehicles: Embedded control. John Wiley & Sons (2013)
17. Mattingley, J., Boyd, S.: CVXGEN: A code generator for embedded convex optimization. Optimization and Engineering **13**(1), 1–27 (2012)
18. Merino, L., Caballero, F., Martínez-de Dios, J.R., Maza, I., Ollero, A.: An unmanned aircraft system for automatic forest fire monitoring and measurement. Journal of Intelligent & Robotic Systems **65**(1), 533–548 (2012)
19. Moarref, M., Rodrigues, L.: An optimal control approach to decentralized energy-efficient coverage problems. 19th IFAC World Congress **47**(3), 6038–6043 (2014)
20. Murray, R.M.: Recent research in cooperative control of multivehicle systems. Journal of Dynamic Systems, Measurement, and Control **129**(5), 571–583 (2007)
21. Nex, F., Remondino, F.: UAV for 3D mapping applications: A review. Applied Geomatics **6**(1), 1–15 (2014)
22. Nguyen, M.T., Rodrigues, L., Stoica Maniu, C., Olaru, S.: Discretized optimal control approach for dynamic multi-agent decentralized coverage. In: IEEE International Symposium on Intelligent Control, pp. 335–340 (2016)
23. Nguyen, M.T., Stoica Maniu, C.: Voronoi based decentralized coverage problem: From optimal control to model predictive control. In: 24th Mediterranean Conference on Control and Automation, pp. 1307–1312 (2016)
24. Nguyen, M.T., Stoica Maniu, C., Olaru, S.: Optimization-based control for multi-agent deployment via dynamic Voronoi partition. 20th IFAC World Congress **50**(1), 1828–1833 (2017)
25. d'Oleire Oltmanns, S., Marzolff, I., Peter, K.D., Ries, J.B.: Unmanned aerial vehicle (UAV) for monitoring soil erosion in Morocco. Remote Sensing **4**(11), 3390–3416 (2012)
26. Papatheodorou, S., Tzes, A.: Cooperative visual convex area coverage using a tessellation-free strategy. In: IEEE 56th Conference on Decision and Control, pp. 4662–4667 (2017)
27. Quintero, S.A.P., Copp, D.A., Hespanha, J.P.: Robust coordination of small UAVs for vision-based target tracking using outputâĂŘfeedback MPC with MHE. In: Cooperative Control of Multi-Agent Systems, pp. 51–83. Wiley-Blackwell (2017)
28. Rinaldi, F., Chiesa, S., Quagliotti, F.: Linear quadratic control for quadrotors UAVs dynamics and formation flight. Journal of Intelligent & Robotic Systems **70**, 203–220 (2013)
29. Schrijver, A.: Theory of linear and integer programming. John Wiley & Sons (1998)
30. Sharifi, F., Chamseddine, A., Mahboubi, H., Zhang, Y.M., Aghdam, A.G.: A distributed deployment strategy for a network of cooperative autonomous vehicles. IEEE Transactions on Control Systems Technology **23**(2), 737–745 (2015)
31. Sharifi, F., Mirzaei, M., Zhang, Y.M., Gordon, B.W.: Cooperative multi-vehicle search and coverage problem in an uncertain environment. Unmanned Systems **3**(1), 35–47 (2015)
32. Sharifi, F., Zhang, Y.M., Aghdam, A.G.: A distributed deployment strategy for multi-agent systems subject to health degradation and communication delays. Journal of Intelligent & Robotic Systems **73**, 623–633 (2014)
33. Siebert, S., Teizer, J.: Mobile 3D mapping for surveying earthwork projects using an unmanned aerial vehicle (UAV) system. Automation in Construction **41**, 1–14 (2014)
34. Singh, S.N., Pachter, M., Chandler, P., Banda, S., Rasmussen, S., Schumacher, C.: Input–output invertibility and sliding mode control for close formation flying of multiple UAVs. International Journal of Robust and Nonlinear Control **10**(10), 779–797 (2000)
35. Stevens, B.L., Lewis, F.L., Johnson, E.N.: Aircraft control and simulation: dynamics, controls design, and autonomous systems. John Wiley & Sons (2015)
36. Stipanović, D.M., Inalhan, G., Teo, R., Tomlin, C.J.: Decentralized overlapping control of a formation of unmanned aerial vehicles. Automatica **40**(8), 1285–1296 (2004)
37. Varga, A.: Solving Fault Diagnosis Problems. Springer (2017)
38. Voronoï, G.: Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Journal für die reine und angewandte Mathematik **134**, 198–287 (1908)
39. Zhang, Y.M., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Annual Reviews in Control **32**(2), 229–252 (2008)

Thomas Chevet received a M.Eng. degree in nuclear engineering from INSTN in 2016 and a M.Eng. degree in automatic control from CentraleSupélec in 2017, both in Paris region, France. He is currently a Ph.D. candidate at the Laboratoire des Signaux et Systèmes (L2S), Paris-Saclay University, France. His research interests include predictive control, control of multi-agent systems and of unmanned aerial vehicles.

Cristina Vlad received the B.Sc. in Electrical Engineering from University "Politehnica" Bucharest, Romania and the Ph.D. degree in Automatic Control from University Paris XI, France, in 2009 and 2013, respectively. She is currently an Assistant Professor at CentraleSupélec, Automatic Control Departement, with the Laboratoire des Signaux et Systèmes (L2S), both within the Paris-Saclay University in France. Her research interests include analysis and control of hybrid systems, energy optimization for automotive applications and model predictive control of multi-agent systems.

Cristina Stoica Maniu received the B.Sc. (2004) and the M.Sc. (2005) degrees in Electrical Engineering from University "Politehnica" Bucharest, the Ph.D. degree (2008) and the HDR – "Habilitation à Diriger des Recherches" – (2014) in Automatic Control from University Paris XI, France. She is currently a Professor at CentraleSupélec with the Laboratoire des Signaux et Systèmes (L2S), Paris-Saclay University. Since 2017, she is a member of the IFAC Technical Committee on Control Education and an IEEE senior member. Her main research activities include set-membership state estimation of uncertain systems, robust predictive control, cooperative control of multi-agent systems, fault detection and fault tolerant control.

Youmin Zhang received his B.S., M.S., and Ph.D. degrees from Northwestern Polytechnical University with specialization in automatic controls, Xi'an, China, in 1983, 1986, and 1995, respectively. He is currently a Professor at the Department of Mechanical, Industrial and Aerospace Engineering and the Concordia Institute of Aerospace Design and Innovation (CIADI), Concordia University, Montreal, Canada. His current research interests include fault diagnosis and fault-tolerant (flight) control systems, cooperative GNC of unmanned aerial/space/ground/surface vehicles. He has authored four books, over 500 journal and conference papers, and book chapters. Dr. Zhang is a Fellow of CSME, a Senior Member of AIAA and IEEE, Vice-President of International Society of Intelligent Unmanned Systems, and a member of the Technical Committee for several scientific societies. He is an Editorial Board Member, Editor-in-Chief, Editor-at-Large, Editor or Associate Editor of several international journals. He has served as the General Chair, the Program Chair, and IPC Member of several international conferences.