



**HAL**  
open science

# Connectivity-Maintenance Teleoperation of a UAV Fleet with Wearable Haptic Feedback

Marco Aggravi, Claudio Pacchierotti, Paolo Robuffo Giordano

## ► To cite this version:

Marco Aggravi, Claudio Pacchierotti, Paolo Robuffo Giordano. Connectivity-Maintenance Teleoperation of a UAV Fleet with Wearable Haptic Feedback. *IEEE Transactions on Automation Science and Engineering*, 2021, 18 (3), pp.1243-1262. 10.1109/TASE.2020.3000060 . hal-02866459

**HAL Id: hal-02866459**

**<https://inria.hal.science/hal-02866459>**

Submitted on 12 Jun 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.

# Connectivity-Maintenance Teleoperation of a UAV Fleet with Wearable Haptic Feedback

Marco Aggravi, *Member, IEEE*, Claudio Pacchierotti, *Senior Member, IEEE*,  
and Paolo Robuffo Giordano, *Senior Member, IEEE*

**Abstract**—This paper presents the design of a decentralized connectivity-maintenance algorithm for the teleoperation of a team of multiple UAVs, together with an extensive human subject evaluation in virtual and real environments. The proposed connectivity-maintenance algorithm enhances earlier works by improving their applicability, safety, effectiveness, and ease of use, by including: (i) an airflow-avoidance behavior that avoids stack downwash phenomena in rotor-based aerial robots; (ii) a consensus-based action for enabling fast displacements with minimal topology changes by having all follower robots moving at the leader’s velocity; (iii) an automatic decrease of the minimum degree of connectivity, enabling an intuitive and dynamic expansion/compression of the formation; and (iv) an automatic detection and resolution of deadlock configurations, i.e., when the robot leader cannot move due to counterbalancing connectivity- and external-related inputs. We also devised and evaluated different interfaces for teleoperating the team as well as different ways of receiving information about the connectivity force acting on the leader. Results of two human subject experiments show that the proposed algorithm is effective in various situations. Moreover, using haptic feedback to provide information about the team connectivity outperforms providing both no feedback at all and sensory substitution via visual feedback.

**Note to Practitioners**—The control of one drone is usually performed with a remote controller (similar to a joystick) that uses radio-wave signals. When controlling more than one drone, even a simple task such as moving the whole team around become very challenging. Developing an easy, yet efficient way to impart commands to a formation of drones is necessary to achieve any complex task. This work proposes a framework to control a fleet of drones (quadrotors) in an intuitive way, while receiving meaningful and effective information on the state of the formation. The proposed technique does not rely on any absolute positioning system (e.g., GPS) or centralized command center. Instead, it only uses the relative position of the drones with respect to each other, and all computations are designed in a decentralized fashion. These features make the proposed framework ready for deployment in different high-impact applications, such as in surveillance, search-and-rescue, and disaster response scenarios.

**Index Terms**—Multi-robot systems, Human-centered robotics.

## I. INTRODUCTION

NATURAL disasters are frighteningly on the rise. According to the Emergency Events Database (EM-DAT), the total number of disasters have shown a significant increase from 1960 onwards, especially in disasters related to weather and climate. In total, the number of weather- and climate-related disasters more than doubled over the past forty years, accounting for 6,392 events in the 20-year period 1996-2015.

It is vital to work on solutions able to mitigate the tragic aftermaths of these events. In fact, potentially hazardous events do not always need to end badly. Disasters occur due to the

combination of an hazard with exposed people and assets vulnerable to the hazard. They are characterized by a lack of resilience and poor ability to cope and respond in the affected area. Toward this important objective, robotics can make a real difference. The use of robots in disaster environments has indeed rapidly increased over the last years, thanks to their expendability, flexibility, and possibility to adapt to different environments and tasks [1]. In this respect, grounded Urban Search-And-Rescue (USAR) robots are already widely used [2], while aerial solutions are only recently raising interest [3]. Moreover, as disaster scenarios are generally highly dynamic and unstructured, it is important to enable human operators to control the robotic systems in a reactive and intuitive way [2].

This paper presents and evaluates an innovative, haptic-enabled framework for the intuitive teleoperation of a fleet of aerial robots, which can be useful in applications of, e.g., Search-And-Rescue, surveillance, mapping, patrol, inspection, and defense. It enables a human operator to naturally control a remote team of drones while receiving rich information about the status of the team and of the given task. The main contributions of our work can be summarized as follows:

- 1) we extend the decentralized connectivity-maintenance algorithm for the teleoperation of a fleet of aerial robots previously developed in [4], [5];
- 2) we introduce and discuss different methods enabling a human operator to control the formation and receive feedback information from it, exploring the possibility of employing innovative wearable haptic interfaces;
- 3) we carry out a human subject study in a simulated environment, comparing via statistical analysis different feedback and control techniques in two scenarios involving groups of up to nine drones;
- 4) we report the results of a real-world experiment, asking two expert operators to control groups of three and four drones in an indoor environment.

The decentralized connectivity-maintenance algorithm introduced in [4], [5] is able to ensure a minimal level of connectivity in a flexible way (i.e., allowing the team to gain/lose interaction links at runtime as long as global connectivity is preserved) by also including in the “connectivity maintenance action” additional constraints/requirements, such as limited sensing/communication and obstacle/inter-robot collision avoidance. While these works provide an interesting baseline for implementing higher-level behaviors in cluttered environments (e.g., exploration, navigation), they also fall short in several crucial aspects, ultimately limiting their applicability. Moreover, these works do not present any, even preliminary, human subject evaluation of their result. For example, a team of robots controlled as in [4], [5]:

- can become unstable due to downwash effects, if relying on Unmanned Aerial Vehicles (UAVs) such as quadrotors;

M. Aggravi, C. Pacchierotti, and P. Robuffo Giordano are with CNRS, Univ Rennes, Inria, IRISA, Rennes, France. E-mail: {marco.aggravi, claudio.pacchierotti, prg}@irisa.fr

- cannot guarantee the maintenance of a certain desired shape for the team, especially when moving in cluttered scenarios and requiring high connectivity values;
- can only impart a contraction of the team (higher commanded connectivity), but not its expansion, which would instead be useful in mapping and surveillance applications;
- can remain stuck while maneuvering close to obstacles or narrow passages.

In this paper, we directly address these limitations as well as extend the control framework to consider the human-robot interaction aspects mentioned in points 2), 3), and 4).

## II. RELATED WORKS

Fleets of UAVs can be beneficial during mountain rescue operations [6], as well as in surveillance, patrolling, and emerging humanitarian needs [7]. Such vast assortment of applications makes UAVs very interesting platforms to study. Moreover, as these target scenarios are highly dynamic and unstructured, it is interesting to research intuitive human-robot interfaces for their effective control.

*Teleoperation of UAVs.* Commercial aerial robots are often commanded using custom remote controllers communicating over radio frequencies or via dedicated smartphone applications. Toward a more intuitive, yet effective, control, researchers have been studying different human-robot interfaces.

A popular approach is using natural gestures obtained from cameras. Ng and Sharlin [8] use the on-board camera of a Parrot ARDrone to recognize the operator gestures. Users are able to initiate take off and landing procedures, as well as modify the drone's altitude or make it follow the operator. Sanna et al. [9] presents a natural user interface using a Microsoft Kinect depth camera. It employs a visual odometry algorithm to control complex maneuvers of a Parrot ARDrone by recognizing gestures and body postures. Pfeil et al. [10] focused on evaluating the user's experience when using a similar control approach. Results show that users appreciate techniques based on easy and understandable metaphors (e.g., intuitive gestures) and that they ultimately serve for a better interaction. Gaze control is another promising way to control drones in an intuitive manner. This approach is particularly useful for impaired users. For example, Hansen et al. [11] present an experimental study on gaze-based control for UAVs. Gaze is used to control a combination of the drone's translation, altitude, rotation, and speed. Another example of intuitive control technique has been presented by Yu et al. [12]. Signals from an EEG headset are transmitted wirelessly to a computer, then the decoded brain signals are converted to controls for the quadrotor. Three applications are presented: using the drone to play games, to see remotely, or to take pictures. Other intuitive control techniques consider wearable interfaces, such as suits and exoskeletons. For example, Rognon et al. [13] developed a wearable exoskeleton which includes an upper-body motion-tracking system, an arm support system to prevent fatigue, and a head-mounted display for first-person-view of the drone perspective. Other examples using wearables have been presented in [14], [15], [16]. Finally, more recently, infrared hand trackers such as the Leap Motion are gaining great popularity. For example, Sarkar et al. [17] uses a Leap

TABLE I  
NOMENCLATURE

Variable	Meaning
$N, N_o$	number of robots and obstacle points
$\mathbf{x}_i, \mathbf{v}_i, \ddot{\mathbf{x}}_i$	position, velocity, and acceleration of robot $i$
$\mathcal{O}$	set of obstacle points
$d_{ij}$	distance between robot $i$ and robot $j$
$d_{io}$	distance between robot $i$ and obstacle point $o_k$
$\mathcal{N}_i, \mathcal{S}_i$	the set of (logical) neighbors and sensing neighbors of robot $i$
$d_{ijk}$	distance from the obstacle point $o_k$ to the line-of-sight between $i$ and $j$
$\mathcal{G}$	the undirected communication/sensing graph
$\mathbf{A}, \mathbf{L}$	adjacency and Laplacian matrix of $\mathcal{G}$
$\lambda_2$	connectivity eigenvalue of $\mathcal{G}$
$\lambda_m$	minimum predefined value for $\lambda_2$
$\lambda_2^{\min}$	lower asymptote for $\lambda_2$
$\mu_\lambda$	exogenous contribution on $\lambda_2^{\min}$
$\lambda_{\min}^{\text{curr}, i}$	$i$ -th agent estimation of $\lambda_2$
$\alpha_{ij}$	hard requirement weight for the pair $(i, j)$
$d_{\min}, d_{\max}$	activation thresholds for $\alpha_{ij}$
$\beta_{ij}$	soft requirement weight for $(i, j)$
$d_0$	preferred inter-distance between two agents
$\gamma_{ij}$	communication constraint weight for $(i, j)$
$d_1, D$	activation thresholds for $\gamma_{ij}^a$
$d_{\min}^b, d_{\max}^b$	activation thresholds for $\gamma_{ij}^b$
$\phi_{ij}$	airflow disturbance avoidance weight for $(i, j)$
$d_{\min}^\phi, D^\phi$	activation thresholds for $f(d_{ij})$
$c_{\min}, c_{\max}$	activation thresholds for $\phi(c_{ij})$
$\mathbf{F}_i^e$	exogenous force for agent $i$
$\mathbf{F}_i^\lambda$	connectivity-maintenance force for agent $i$
$\sigma_v, \sigma_c$	activation parameters for the consensus on the leader's velocity and formation expansion behaviors
$\hat{\mathbf{v}}_{i,1}$	$i$ -th agent estimation of the leader velocity
$b_l, b_\mu$	damping factor for the consensus on leader velocity and deadlock resolution behaviors
$\hat{\mathbf{d}}_{c,i}$	$i$ -th agent estimation of the formation centroid
$\bar{v}_c$	desired speed for expanding the formation
$F_1^\perp$	projection of $\mathbf{F}_1^\lambda$ onto $\mathbf{F}_1^e$ for the leader agent
$P(F_1^\perp)$	scaling weight for $F_1^\perp$
$F_{\min}^\perp, F_{\max}^\perp$	activation thresholds for $P(F_1^\perp)$
$\mathbf{p}^L, \mathbf{p}^O, \mathbf{p}^H$	input velocities from the joypad, Omega.6 device, and Leap Motion
$b_i$	damping factor for external inputs
$f_m, \kappa_m$	frequency and period of activation for motor $m$

Motion for controlling a Parrot ARDrone with hand gestures. Hand position and orientation are translated into velocities and commands for the roll, pitch, and yaw of the quadrotor. Fernandez et al. [18] also employs a Leap Motion, comparing the performance of using hand gestures, speech commands, or a combination of the two. Subjects prefer the latter modality, leading to an interesting integrated control approach.

Although these approaches have been deemed effective and they are used in different applications, they rarely provide rich feedback information on the status of the robot or the given task. Such information is even more important when considering multiple robots in a team.

*Feedback information.* For this reason, it is important to research effective feedback techniques when operating one or multiple aerial robots. Commercial approaches often provide visual feedback from on-board cameras, while solutions providing other types of feedback (e.g., haptic, auditory) are very rare in the literature. For example, Rognon et al. [19] devise a cable-driven haptic guidance system for the body.

It applies haptic feedback relative to the distance between the drone and a predetermined trajectory, to correct user torso orientation and improve flight precision. Cacace et al. [20] use a vibrotactile armband to provide the operator with information about the robot deviation with respect to the planned trajectory. Similarly, Tsykunov et al. [21] use a wearable tactile glove to provide information about the drones formation state (extension, contraction, and displacement) and propagation direction (increasing or decreasing drone-to-drone distance). Lee et al. [22], Robuffo Giordano et al. [23], and Franchi et al. [24] use a grounded haptic interface to control the fleet leader and provide haptic feedback regarding the status of the team. Similarly, Claret et al. [25] and Staub et al. [26] use a grounded haptic interface to control an heterogeneous robotic team composed of one drone and one robotic manipulator. In our previous work [27], we use a wearable device for the forearm to control a fleet of drones in a simulated environment. The position of the hand, tracked by a Leap Motion, controls the position of the robot leader, while the wearable device provides information on the status of the team and the task. Results show that wearable haptic feedback outperforms sensory substitution via visual feedback. An improved version of the same device will be used in this work as one of our feedback devices.

As the system becomes more complex, it becomes more important to provide rich information to the human operator. This is of course the case when controlling a fleet of drones.

*Formation control.* Having an operator controlling a fleet of drones poses several challenges, as the operator needs to effectively control multiple drones at the same time. Researchers have proposed different approaches to address these issues. For instance, in [28] a “leader-follower” approach has been proposed with the formation having the flexibility to split/join links at runtime because of sensing constraints or other requirements. The operator directly controls a single robot (the leader), and the other robots coarsely follow the leader motion by implementing spring-like couplings among connected pairs. This flexible approach simplifies the operator’s task since she/he must only focus on the leader robot, however it also provides a very limited control over the formation shape and overall behavior. Extensions of these ideas by adding more global constraints/requirements have been explored in e.g., [4], [29], [30] where the leader-follower approach is completed with the maintenance of global properties of interest such as connectivity or rigidity. This effectively gives the operator a way to have more control over the formation overall behavior, while still guaranteed a good level of flexibility in the individual behavior of the robots. In [31], [32] the possibility of changing the leader identity online for optimizing the tracking performance of the operator’s commands has also been explored. At the opposite, other approaches have focused on more ‘rigid’ scenarios, in which the operator controls the collective motion of the whole formation (e.g., the velocity of its barycenter) while the robots maintain a desired shape during motion. Examples of this approach can be found in [22] for formations defined in terms of relative positions, and in [33] for formations defined in terms of relative bearing measurements. A summary of several possibilities for interfacing an operator with multiple robots can be found in, e.g., [24].

### III. PRELIMINARIES

In this section, we briefly summarize the decentralized connectivity-maintenance algorithm introduced in [4], [5], which will then be extended in Sect. IV. First, as classically done, let us define  $\mathbf{1}_N \in \mathbb{R}^N$  and  $\mathbf{0}_N \in \mathbb{R}^N$  as vectors with all ones and zeros, respectively, and  $\mathbf{I}_N \in \mathbb{R}^{N \times N}$  as the identity matrix of dimension  $N$ . Furthermore, the symbol  $\otimes$  will denote the Kronecker product, and we will use the shorthand  $\mathbf{1}_{N_3} = \mathbf{1}_N \otimes \mathbf{I}_3$ . Some quantities of interest, used throughout the manuscript, are defined in Table I for the reader’s convenience.

#### A. Background

We consider a group of  $N$  robots modeled as 3D-point masses with positions  $\mathbf{x}_i \in \mathbb{R}^3$ ,  $i \in \{1, \dots, N\}$ , and double integrator dynamics

$$\ddot{\mathbf{x}}_i = \mathbf{u}_i, i \in \{1, \dots, N\}, \quad (1)$$

with  $\mathbf{u}_i \in \mathbb{R}^3$  being the force (control action) applied to robot  $i$ , and  $\mathbf{v}_i = \dot{\mathbf{x}}_i$  the  $i$ -th robot velocity. Moreover, we define  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N) \in \mathbb{R}^{3N}$  and  $\mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_N) \in \mathbb{R}^{3N}$ . This way of modeling the robot formation is quite common in the field [4], [5], [34]. The environment is modeled as a collection of  $N_o$  3D obstacle points  $\mathcal{O} = \{\mathbf{o}_k \in \mathbb{R}^3, k \in \{1, \dots, N_o\}\}$ . Interactions among robots are represented by an undirected graph  $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ , where  $\mathbf{V} = \{1, \dots, N\}$  is the vertex set and  $\mathbf{E} = \{e_{ij} = (i, j) \mid i \in \mathbf{V}, j \in \mathbf{V}, i \neq j\}$  is the edge set. Each edge  $e_{ij}$  is associated with a scalar weight  $A_{ij} \in [0, 1]$ , that represents the “quality” of the edge connectivity, ranging from  $A_{ij} = 0$  if  $e_{ij}$  is lost/disconnected (e.g., because of sensing constraints) to  $A_{ij} = 1$  if  $e_{ij}$  is fully connected.

Following [4], each  $A_{ij}$  is further partitioned as the product of three sub-weights

$$A_{ij} = \alpha_{ij}\beta_{ij}\gamma_{ij}, \quad (2)$$

where  $\gamma_{ij} \in [0, 1]$  takes into account the *sensing/communication constraints* (i.e., the physical ability of a pair  $(i, j)$  to communicate or sense each other),  $\beta_{ij} \in [0, 1]$  accounts for a set of *soft requirements* (requirements which should be preferably met by the pair  $(i, j)$ , e.g., maintaining a desired interdistance), and  $\alpha_{ij} \in [0, 1]$  represents a set of *hard requirements* (requirements which must be necessarily met by the pair  $(i, j)$ , e.g., avoiding inter-robot and obstacle collisions). Accordingly, two neighboring definitions are introduced: the set of *sensing neighbors*  $\mathcal{S}_i = \{j \in \mathbf{V} \mid \gamma_{ij} > 0\}$ , i.e., the agents  $j$  with whom agent  $i$  can physically exchange information as per the sensing model  $\gamma_{ij}$ , and the usual set of (logical) *neighbors*  $\mathcal{N}_i = \{j \in \mathbf{V} \mid A_{ij} > 0\}$ .

The *adjacency matrix*  $\mathbf{A} \in \mathbb{R}^{N \times N}$  associated to graph  $\mathcal{G}$  and induced by weights (2) is defined as

$$[A]_{ij} = \begin{cases} 1 & \text{if } A_{ij} > 0 \\ 0 & \text{otherwise} \end{cases}.$$

If weights  $A_{ij}$  are designed as symmetric (i.e.,  $A_{ij} = A_{ji}$ ), then  $\mathbf{A} = \mathbf{A}^T$  as expected. From matrix  $\mathbf{A}$ , one can build the so-called *Laplacian matrix* associated to graph  $\mathcal{G}$  as  $\mathbf{L} = \text{diag}(\delta_i) - \mathbf{A}$ , where  $\delta_i = \sum_{j=1}^N A_{ij}$ . As well known, the

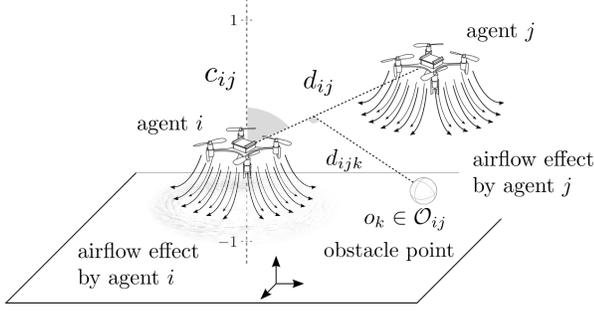


Fig. 1. Given a robot pair  $(i, j)$ ,  $d_{ij}$  denotes the distance among  $i$  and  $j$ ,  $d_{ijk}$  is the distance between the segment joining  $i$  and  $j$  and an obstacle point  $o_k$ , and  $c_{ij}$  is the cosine of the angle between the direction from robot  $i$  to robot  $j$  and the vertical axis of agent  $i$  (highlighted in grey). To ensure the robots line-of-sight visibility constraint in (6),  $\gamma_{ij}$  will vanish if  $d_{ijk}$  is smaller than the minimum threshold  $d_{\min}^o$ . The propellers of robots with vertical take-off and landing capabilities (such as quadrotors) produce strong turbulence that can destabilize other flying robots passing below.

second-smallest eigenvalue of  $\mathbf{L}$ , denoted as  $\lambda_2$ , is a metric for determining the connectivity of a graph, since  $\lambda_2 > 0$  for a connected graph and  $\lambda_2 = 0$  otherwise [35].

Finally, we assume that any pair of (sensing) neighbors  $(i, j)$  (for which  $\gamma_{ij} > 0$ ) is able to measure its relative position  $\mathbf{x}_j - \mathbf{x}_i$ , that each robot  $i$  has a measurement of its own velocity  $\mathbf{v}_i$ , and that any robot  $i$  can measure its relative position  $\mathbf{o}_k - \mathbf{x}_i$  w.r.t. any obstacle point  $\mathbf{o}_k$  within a sensing range.

### B. Weight definition

Here we briefly recall the design guidelines introduced in [4] for encoding in  $A_{ij}$  (a) the sensing/communication constraints of maximum range and non-occluded visibility among robot pairs (accounted by weight  $\gamma_{ij}$ ), (b) the *soft* requirement of maintaining a preferred inter-distance among robot pairs (accounted by weight  $\beta_{ij}$ ), and (c) the *hard* requirement of avoiding inter-robot collisions among robot pairs (accounted by weight  $\alpha_{ij}$ ). This machinery will then be exploited in Sec. IV-A.

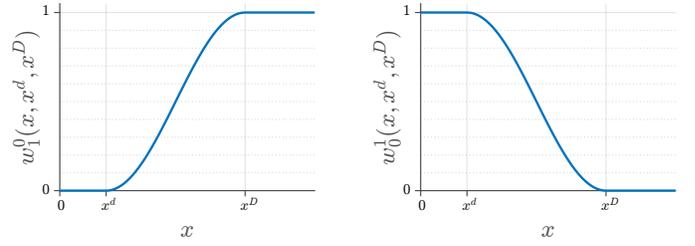
For a robot pair  $(i, j)$  and an obstacle point  $o_k$ , let  $d_{ij}$  represent the distance among  $i$  and  $j$  and  $d_{ijk}$  the distance from  $o_k$  to the segment joining  $i$  and  $j$  (see Fig. 1). Weight  $\beta_{ij}(d_{ij})$  (soft requirement) is taken as the bell-shaped curve

$$\beta_{ij} = e^{-\frac{(d_{ij} - d_0)^2}{\sigma}},$$

with  $d_0 > 0$  being a preferred interdistance and  $\sigma > 0$  a spread parameter. Since  $\beta_{ij}$  has a unique maximum for  $d_{ij} = d_0$ , it will then aim at keeping the preferred interdistance  $d_0$  for the pair  $(i, j)$ .

The definition of the other weights is more articulated, and it requires the introduction of two auxiliary functions

$$w_1^0(x, x^d, x^D) = \begin{cases} 0 & x \leq x^d \\ \frac{1}{2} (1 - \cos(ax + b)) & x^d < x \leq x^D \\ 1 & x^D < x \end{cases} \quad (3)$$



(a) Rising weight function

(b) Falling weight function

Fig. 2. (a) Rising weight function  $w_1^0(x, x^d, x^D)$  with  $x^d, x^D$  minimal and maximal  $x$  axis threshold, respectively. (b) Falling weight function  $w_0^1(x, x^d, x^D)$ , with  $x^d, x^D$  minimal and maximal  $x$  axis threshold, respectively.

and

$$w_0^1(x, x^d, x^D) = \begin{cases} 1 & x \leq x^d \\ \frac{1}{2} (1 + \cos(ax + b)) & x^d < x \leq x^D \\ 0 & x^D < x \end{cases} \quad (4)$$

with  $a = \pi/(x^D - x^d)$  and  $b = -ax^d$ . Function  $w_1^0(x, x^d, x^D)$  is a “rising” activation function with lower/upper thresholds set at  $x^d$  and  $x^D$  (a possible shape is shown in Fig. 2a). Similarly, function  $w_0^1(x, x^d, x^D)$  is a “falling” activation function with lower/upper thresholds set at  $x^d$  and  $x^D$  (a possible shape is shown in Fig. 2b). These two functions can be conveniently exploited/combined for implementing more complex behaviors such as those summarized hereafter and further developed in Sec. IV-A. With this setting, weight  $\alpha_{ij}$  is then defined as

$$\alpha_{ij} = \left( \prod_{k \in \mathcal{S}_i} \alpha_{ik}^* \right) \cdot \left( \prod_{k \in \mathcal{S}_j / \{i\}} \alpha_{jk}^* \right) = \alpha_i \cdot \alpha_j / i, \quad (5)$$

where  $\alpha_{ij}^* = w_1^0(d_{ij}, d_{\min}, d_{\max})$  and  $0 < d_{\min} < d_{\max}$  are minimum/maximum safety distances for a robot pair. As explained in [4], this particular construction also allows the implementation of the *hard constraint* of inter-robot collision avoidance. In fact, in short, if the distance  $d_{ij}$  among two robots  $(i, j)$  falls below  $d_{\min}$ , then  $\alpha_{ik} = 0 \forall k \in \mathcal{S}_i$  and  $\alpha_{jk} = 0 \forall k \in \mathcal{S}_j$ , thus forcing the disconnection of the whole graph  $\mathcal{G}$  (as robots  $i$  and  $j$  would lose all their edges). Finally, weight  $\gamma_{ij}$  is

$$\gamma_{ij} = \gamma_{ij}^a \prod_{o_k \in \mathcal{O}_{ij}} \gamma_{ij}^b = \gamma_{ij}^a \gamma_{ijk}, \quad (6)$$

where  $\gamma_{ij}^a = w_1^0(d_{ij}, d_1, D)$  is a “falling” activation function that models a maximum range sensing/communication constraint (with parameters  $0 < d_1 < D$ ), and  $\gamma_{ij}^b = w_1^0(d_{ijk}, d_{\min}^o, d_{\max}^o)$  is a “rising” activation function that models a line-of-sight visibility constraint w.r.t. an obstacle point  $o_k$  (the weight vanishes if  $d_{ijk}$  is smaller than the minimum threshold  $d_{\min}^o$ ). In (6), the set  $\mathcal{O}_{ij}$  contains all the obstacle points visible to either robot  $i$  or  $j$ . Furthermore, as a byproduct, the product sequence  $\gamma_{ijk}$  also encodes the *hard requirement* of obstacle collision avoidance (besides the line-of-sight sensing/communication constraint).

We conclude recalling that  $\alpha_{ij} = \alpha_{ji}$ ,  $\beta_{ij} = \beta_{ji}$ ,  $\gamma_{ij} = \gamma_{ji}$ , and, therefore,  $A_{ij} = A_{ji}$  as required [4].

### C. Connectivity maintenance control

The connectivity maintenance control consists in a decentralized gradient descent of a potential function  $V^\lambda(\lambda_2(\mathbf{x}(t))) \geq 0$  of the connectivity eigenvalue  $\lambda_2(\mathbf{x}(t))$  when weighting the edges of graph  $\mathcal{G}$  with the weights (2). In what follows, w.l.o.g. we denote  $\lambda_2(\mathbf{x}(t))$  simply as  $\lambda_2(t)$ . The potential  $V^\lambda(\lambda_2(t))$  is designed to have a vertical asymptote at  $\lambda_2^{\min} > 0$ , to smoothly decrease from  $\lambda_2^{\min}$  to  $\lambda_2^{\max} = \lambda_2^{\min} + \Delta$ , and to vanish for  $\lambda_2(t) \geq \lambda_2^{\max}$ , see Fig. 3 for a possible shape. The connectivity maintenance control is then designed as the gradient descent

$$\mathbf{F}_i^\lambda = -\frac{\partial V^\lambda(\lambda_2(t))}{\partial \mathbf{x}_i}, \quad (7)$$

whose evaluation by robot  $i$  can be made decentralized by exploiting, in particular, a decentralized estimation  $\hat{\lambda}_2^i(t)$  of the true value  $\lambda_2(t)$  [4]. The control input for robot  $i$  in (1) is then finally designed as

$$\mathbf{u}_i = \mathbf{F}_i^\lambda - \mathbf{B}_i \mathbf{v}_i + \mathbf{F}_i^e, \quad (8)$$

where the connectivity force  $\mathbf{F}_i^\lambda$  is evaluated using estimated quantities for attaining decentralization (e.g., the aforementioned  $\hat{\lambda}_2^i(t)$  instead of the true  $\lambda_2(t)$ ),  $\mathbf{B}_i \in \mathbb{R}^{3 \times 3}$  is a damping term (for stabilization purposes), and  $\mathbf{F}_i^e \in \mathbb{R}^3$  an additional exogenous force that may act on robot  $i$ . In presence of bounded external forces  $\mathbf{F}_i^e$ , the connectivity maintenance control (8) is guaranteed to always maintain  $\lambda_2(t) > \lambda_2^{\min} > 0$ ,  $\forall t$ .

An extension of (7) has further been proposed in [5] for allowing the possible *online* variation of the lower bound  $\lambda_2^{\min}$  of  $V^\lambda(\lambda_2(t))$  by means of an external input (e.g., an additional control/planning module, or a human operator). This feature makes it possible to change at runtime the minimum degree of connectivity of the group (i.e., the location of the vertical asymptote of  $V^\lambda(\lambda_2(t))$ ) and, thus, obtain a more flexible behavior (e.g., to compact or loosen the group during motion depending on the situation at hand). With reference to Fig. 3, this is obtained as follows: let  $\lambda_m > 0$  be a *constant parameter* representing a pre-defined *minimum* lower-bound for the degree of connectivity  $\lambda_2^{\min}(t)$ , i.e., the maximum degree of dispersion that can be allowed for the robot group at any time  $t$ . Let also  $\mu_\lambda(t) \geq 0$  be the desired increment above  $\lambda_m$ , and  $\eta > 0$  a safety margin: the goal is to set  $\lambda_2^{\min}(t) = \lambda_m + \mu_\lambda(t)$  subject

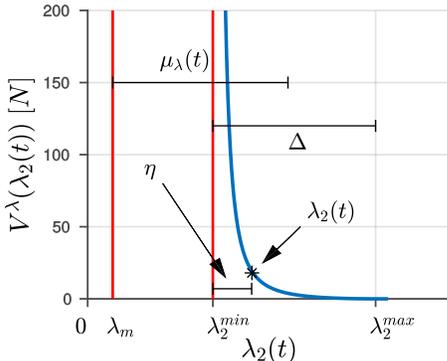


Fig. 3. Potential function  $V^\lambda(\lambda_2(t))$ , which is used to compute the connectivity force  $\mathbf{F}_i^\lambda$ . The asymptote  $\lambda_2^{\min}$  is computed from (9), using the actual value of  $\lambda_2(t)$ , a desired increment  $\mu_\lambda(t)$ , a safe margin  $\eta$ , and the minimal value  $\lambda_m$ . The limit value  $\lambda_2^{\max}$  is computed from  $\lambda_2^{\min}$  and a fixed increment  $\Delta > 0$ .

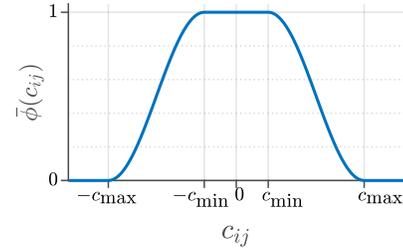


Fig. 4. Function  $\bar{\phi}(c_{ij})$ . Values  $c_{\min}$  and  $c_{\max}$  are the two activation/deactivation thresholds ( $0 < c_{\min} < c_{\max} < 1$ ).

to the constraint that  $\lambda_2^{\min}(t) \leq \lambda_2(t) - \eta$  (the location of the vertical asymptote must always be smaller than the current  $\lambda_2(t)$  minus a safety margin  $\eta$ ). The value of  $\lambda_2^{\min}(t)$  is then taken as

$$\lambda_2^{\min}(t) = \lambda_m + \text{sat}_{[0, \lambda_2(t) - \lambda_m - \eta]}(\mu_\lambda(t)), \quad (9)$$

where  $\text{sat}_{[0, \lambda_2(t) - \lambda_m - \eta]}$  is a linear saturation with 0 and  $\lambda_2(t) - \lambda_m - \eta$  being its lower and upper limits, respectively. This saturation function guarantees that, whatever the desired increment  $\mu_\lambda(t)$ ,  $\lambda_2^{\min}(t) \leq \lambda_2(t) - \eta$  as required.

In the context of [5], as well as here, the desired increment  $\mu_\lambda(t)$  is communicated to a specific robot (the “leader” w.l.o.g. taken as robot 1) and then propagated to the other robots in the group via a consensus law. In particular, letting  $\hat{\mu}_i(t)$  be the  $i$ -th robot estimation of  $\mu_\lambda(t)$ , the following decentralized propagation is implemented

$$\begin{cases} \dot{\hat{\mu}}_i &= k_\mu \sum_{j \in \mathcal{N}_i} (\hat{\mu}_j - \hat{\mu}_i), & i = 2, \dots, N \\ \dot{\hat{\mu}}_1 &= k_\mu \sum_{j \in \mathcal{N}_1} (\hat{\mu}_j - \hat{\mu}_1) + b_\mu (\mu_\lambda - \hat{\mu}_1), & b_\mu > 0, k_\mu > 0. \end{cases} \quad (10)$$

Each robot can then obtain its own estimation  $\hat{\lambda}_{2,i}^{\min}(t)$  of  $\lambda_2^{\min}(t)$  by evaluating (9) on the estimated  $\hat{\mu}_i(t)$  (replacing  $\mu_\lambda(t)$ ) and on the estimated  $\hat{\lambda}_2^i(t)$  (replacing  $\lambda_2(t)$ ). Assuming that these estimations are fast enough compared to the group dynamics (which is typically the case), all  $\hat{\lambda}_{2,i}^{\min}(t)$  will agree on the true value  $\lambda_2^{\min}(t)$  apart from negligible transients (see also [5]). Each robot  $i$  can then implement the control (8) by plugging  $\hat{\lambda}_{2,i}^{\min}(t)$  into the computation of the connectivity force  $\mathbf{F}_i^\lambda$  and obtain the desired effect: an online adjustment of the minimum degree of connectivity as the robot group moves in the environment, with  $\mu_\lambda(t)$  (communicated by an any external “entity” to the leader) driving this adjustment.

## IV. DECENTRALIZED CONNECTIVITY MAINTENANCE ALGORITHM

This Section details the proposed improvements over the decentralized connectivity maintenance algorithm described in Sec. III. As explained in Sec. I, they are meant to overcome: airflow/downwash disturbances, which may happen when employing UAVs (Sec. IV-A); undesired changes in the formation topology (Sec. IV-B), which may happen when navigating cluttered environments; the lack of an active expansion action of the formation (Sec. IV-C), for being able to extend the area covered by the robotic team; and the mitigation of any stuck/deadlock situation (Sec. IV-D), which may happen when maneuvering close to obstacles or through narrow passages.

### A. Airflow disturbance avoidance

*Motivation (see Fig. 1):* robots with vertical take-off and landing capabilities (such as quadrotors) can be destabilized by airflow effects (or downwash effects) if they fly closely above each other [36], [37]. The strong turbulence produced by the robot propellers are hard to model (and thus to compensate for), and they may lead to highly unstable behaviors as well as to a quick and total loss of control of the bottom robot(s).

To address this important yet often neglected issue, Yeo et al. [38], [39], [40] developed empirical models and custom airflow sensors to estimate and compensate for any wind turbulence. However, instead of further instrumenting the robots, another possibility is to avoid the occurrence of a downwash destabilization in the first place, by preventing any robot from flying above and too close to another one. This requirements can be conveniently enforced by modifying the weighting machinery described in Sec. III-A so as to impose, similarly to [37], an additional constraint on the mutual robot positions. This is obtained by appending an additional weight  $\phi_{ij}$  to (2) for encoding this additional requirement as a *hard constraint*.

Weight  $\phi_{ij}$  should penalize any situation in which two robots  $i$  and  $j$  are nearly on top of each other *and* their distance  $d_{ij}$  is small enough for letting the turbulence generated by the top robot to affect the stability of the bottom one. Since  $\phi_{ij}$  is meant to be a hard constraint, its construction will follow that of weight  $\alpha_{ij}$  in (5). This behavior is achieved in two steps: first, with reference to Fig. 1, let

$$c_{ij} = \left( \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \right)^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (11)$$

be the cosine of the angle between the direction from robot  $i$  to robot  $j$  and the vertical (in grey in Fig. 1). Therefore,  $c_{ij} = 1$  if robot  $j$  is exactly on top of robot  $i$ ,  $c_{ij} = 0$  if they are at the same height, and  $c_{ij} = -1$  if robot  $i$  is exactly on top of robot  $j$ . Then, we define

$$\bar{\phi}(c_{ij}) = \begin{cases} w_0^1(c_{ij}, c_{\min}, c_{\max}) & \text{if } c_{ij} \geq 0 \\ w_1^0(c_{ij}, -c_{\max}, -c_{\min}) & \text{if } c_{ij} < 0 \end{cases} \quad (12)$$

as the function weighting the angular position between robots  $i$  and  $j$ , with  $0 < c_{\min} < c_{\max} < 1$  being the two activation/deactivation thresholds, see Fig. 4 for an illustration. One can verify the smoothness of  $\bar{\phi}(c_{ij})$  and how it encodes the desired behavior:  $\bar{\phi} = 0$  if  $c_{ij} \rightarrow \pm 1$  and  $\bar{\phi} = 1$  if  $c_{ij} \rightarrow 0$ .

The second necessary step is to introduce another function of the distance  $d_{ij}$  whose role is to activate/deactivate  $\bar{\phi}$  as  $d_{ij}$  becomes too small/large, respectively. This is defined as

$$f(d_{ij}) = w_1^0(d_{ij}, d_{\min}^\phi, D^\phi), \quad (13)$$

where, as usual,  $0 < d_{\min}^\phi < D^\phi$  are activation/deactivation parameters. With these settings, we then define the weight

$$\phi_{ij}^*(c_{ij}, d_{ij}) = \bar{\phi}(c_{ij}) (1 - f(d_{ij})) + f(d_{ij}) \quad (14)$$

for encoding the desired behavior. If the distance is large enough ( $d_{ij} \geq D^\phi$ ), then  $\phi_{ij}^*(c_{ij}, d_{ij}) = 1$  regardless of the robot relative position (thus, no airflow effects are considered

even if the robots happen to be vertically aligned); if the distance is small enough ( $d_{ij} \leq d_{\min}^\phi$ ), then  $\phi_{ij}^*(c_{ij}, d_{ij}) = \bar{\phi}(c_{ij})$  and any vertical alignment among robots  $i$  and  $j$  is penalized; if the distance is between  $d_{\min}$  and  $D^\phi$ , the weight  $\phi_{ij}^*(c_{ij}, d_{ij})$  smoothly changes between the two conditions. Finally, following (5), the total weight  $\phi_{ij}$  is defined as

$$\phi_{ij} = \left( \prod_{k \in \mathcal{S}_i} \phi_{ik}^* \right) \left( \prod_{k \in \mathcal{S}_j / \{i\}} \phi_{jk}^* \right) = \phi_i \cdot \phi_j / i. \quad (15)$$

As explained in Sec. III-B, this construction allows for turning the airflow avoidance requirement into a *hard* constraint. Weight  $\phi_{ij}$  is then appended to (2) as

$$A_{ij} = \alpha_{ij} \beta_{ij} \gamma_{ij} \phi_{ij}. \quad (16)$$

Since  $\phi_{ij} = \phi_{ji}$  (see the Appendix B),  $A_{ij} = A_{ji}$  as required. Also, analogously to  $\alpha_{ij}$ , weight  $\phi_{ij}$  is ultimately a function of the relative positions between robots  $i$  and  $j$  and their neighbors. Therefore, its evaluation (and the evaluation of its gradient) can be decentralized, as explained in [4] for  $\alpha_{ij}$ .

This disturbance avoidance technique can be extended to also consider other similar issues, such as ground disturbances. To the best of our knowledge, this is the first time that such avoidance technique is employed in applications of shared control of multiple robots.

### B. Consensus on the leader velocity

*Motivation (see Fig. 5 and [https://youtu.be/\\_eMsbMlmqaU?t=67](https://youtu.be/_eMsbMlmqaU?t=67)):* Let us consider a robotic team moving in a cluttered environment. The user controls the motion of the robot leader via the force input  $\mathbf{F}_1^e$ , as detailed in (8). Whenever this external force is applied to the leader, the response of the rest of the formation depends on the minimum connectivity value: if  $\lambda_2^{\min}$  is small (Fig. 5a), the formation stretches as the leader moves forward; if  $\lambda_2^{\min}$  is high (Fig. 5b), the formation keeps a more compact shape. However, in both cases, the initial shape is lost. For example, in Fig. 5a (low  $\lambda_2^{\min}$ , more flexibility), when the leader arrives at the target location, the last robot is still midway. Conversely, in Fig. 5b (high  $\lambda_2^{\min}$ , more rigidity), the robotic team arrives compact, but the positions of the robots with respect to each other are very different from the beginning. Another drawback of using high  $\lambda_2^{\min}$  values is the risk of deadlock/stall situations, which will be addressed in Sec. IV-D.

In order to maintain the connectivity of the formation as well as its initial shape (as much as possible), we propose a consensus-driven approach on the leader robot velocity. When this feature is enabled, each robot tracks the leader velocity in a decentralized way, regardless of  $\lambda_2^{\min}$ . In this way, the operator can control the group as a whole while still benefiting from a small  $\lambda_2^{\min}$  (and, thus, a high flexibility in losing/gaining edges in presence of obstacles or other obstructions).

Let us consider w.l.o.g robot 1 as the leader of the team, moving at velocity  $\mathbf{v}_1$ . We define  $\hat{\mathbf{v}}_{i,1} \in \mathbb{R}^3$  as the  $i$ -th robot estimation of  $\mathbf{v}_1$ , similarly to (10). The robots then implement

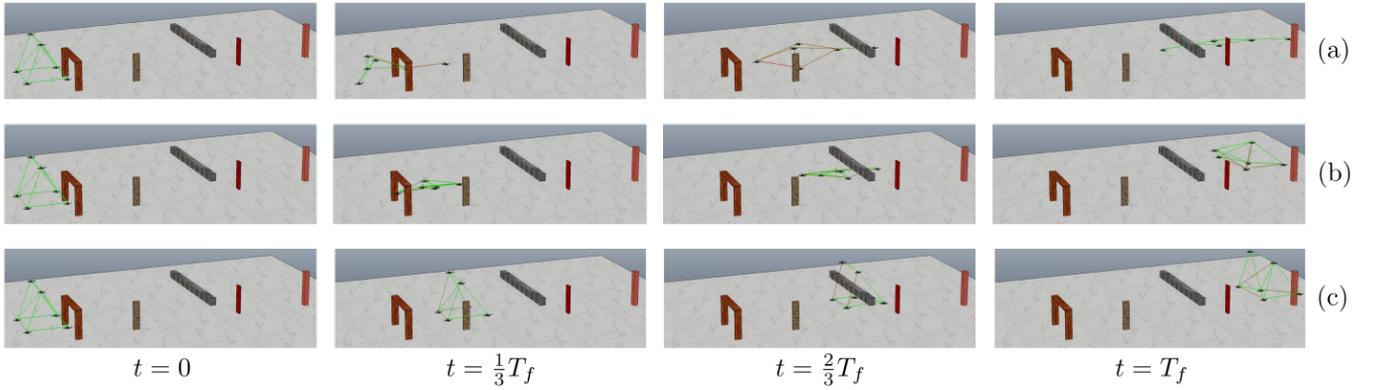


Fig. 5. Consensus on the leader velocity. (a) With a small  $\lambda_2^{\min} = \lambda_m = 0.01$ , the formation stretches while the leader robots reaches the target location ( $T_f = 165$  s). (b) With a high  $\lambda_2^{\min} = 1.5$ , the robot formation moves compactly, but complex maneuvers are necessary to avoid the obstacles. Moreover, the topology of the formation changes in a consistent way during the trial ( $T_f = 142$  s). (c) With our feature ( $b_l = 1, \sigma_v = 1$ ) and a small  $\lambda_2^{\min} = \lambda_m$ , the robot formation is rather compact, yet flexible and easy to maneuver through the cluttered environment ( $T_f = 68$  s). Video available at [https://youtu.be/\\_eMsbMlmqaU?t=67](https://youtu.be/_eMsbMlmqaU?t=67).

$$\begin{cases} \dot{\hat{\mathbf{v}}}_{i,1} = k_v \sum_{j \in \mathcal{N}_i} (\hat{\mathbf{v}}_{j,1} - \hat{\mathbf{v}}_{i,1}), & i = 2, \dots, N \\ \dot{\hat{\mathbf{v}}}_{1,1} = k_v \sum_{j \in \mathcal{N}_1} (\hat{\mathbf{v}}_{j,1} - \hat{\mathbf{v}}_{1,1}) + b_l (\mathbf{v}_1 - \hat{\mathbf{v}}_{1,1}) \end{cases} \quad (17)$$

with  $b_l > 0$  and  $k_v > 0$ . Then, the estimated  $\hat{\mathbf{v}}_{i,1}$  is added to (8) as

$$\begin{aligned} \mathbf{u}_i &= \mathbf{F}_i^\lambda - \mathbf{B}_i \mathbf{v}_i + \mathbf{F}_i^e + \sigma_v \mathbf{B}_i \hat{\mathbf{v}}_{i,1} \\ &= \mathbf{F}_i^\lambda + \mathbf{B}_i (\sigma_v \hat{\mathbf{v}}_{i,1} - \mathbf{v}_i) + \mathbf{F}_i^e, \quad i = 2, \dots, N, \end{aligned} \quad (18)$$

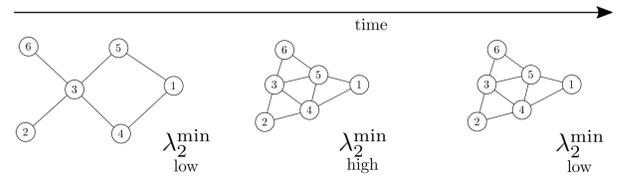
where  $\sigma_v \in \{0, 1\}$  is a binary activation/deactivation parameter. When active ( $\sigma_v = 1$ ) and in a “steady-state” regime (i.e.,  $\mathbf{F}_i^\lambda = 0, \mathbf{F}_i^e = 0$  and  $\mathbf{v}_1 \approx \text{const}$ ), (18) makes each robot velocity  $\mathbf{v}_i$  converge toward the estimated leader velocity  $\hat{\mathbf{v}}_{i,1}$ .

Fig. 5c shows a representative behavior of this approach. Changes in the formation shape are minimal, the formation keeps a rather compact shape, and it is still able to lose/gain edges as needed for maneuvering through the obstacles.

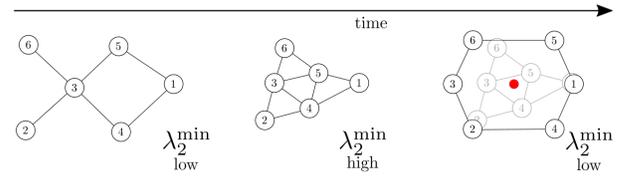
### C. Formation expansion

*Motivation (see Fig. 6 and [https://youtu.be/\\_eMsbMlmqaU?t=105](https://youtu.be/_eMsbMlmqaU?t=105)):* an increase of  $\lambda_2^{\min}$  automatically results in an increase of  $V^\lambda(\lambda_2(t))$ , which in turn results in a compacting of the formation (for reaching the required higher  $\lambda_2(t)$ ). On the other hand, a decrease of  $\lambda_2^{\min}$  does not have any direct effect on  $V^\lambda(\lambda_2(t))$ : in fact, the connectivity control actively compacts the formation if *more* connectivity is requested, but the opposite does not happen (see Fig. 6a). This behavior may not be desirable, as it prevents the human operator from directly commanding an expansion of the formation when decreasing  $\lambda_2^{\min}$ . Such a feature might instead be useful in mapping and surveillance applications, where the robotic team should cover as much ground as possible whenever commanded to do so.

In order to address this issue, we introduce a new feature that activates an automatic expansion of the formation w.r.t. the group centroid  $\frac{1}{N} \sum_{i=1}^N \mathbf{x}_i$  whenever  $\lambda_2 \geq \lambda_2^{\max}$  (i.e., whenever  $\mathbf{F}_i^\lambda = 0$ , see Fig. 3). This sought behavior is illustrated in Fig. 6b. This new feature requires each robot to have access to its relative positions w.r.t. the centroid,  $\mathbf{x}_i - \frac{1}{N} \sum_{j=1}^N \mathbf{x}_j$



(a) Feature not active (as in [4]). If  $\lambda_2$  is high (i.e., the formation is compact), decreasing the lower asymptote  $\lambda_2^{\min}$  will not make the formation expand. This design prevents the human operator from being able to control the expansion of the formation.



(b) Feature active. In this case, if possible (i.e., if  $\lambda_2 \geq \lambda_2^{\max}$ ), the formation will expand as soon as the operator commands a lower  $\lambda_2^{\min}$ .

Fig. 6. Formation expansion. (a) If the feature is not active, the formation does not automatically expand when  $\lambda_2 \geq \lambda_2^{\max}$ . (b) If the feature is active, the formation automatically expands up to  $\lambda_2 = \lambda_2^{\max}$ . The human operator can activate/deactivate this feature at will.

while, however, each robot is only assumed to measure its relative position  $\mathbf{x}_j - \mathbf{x}_i$  w.r.t. any neighboring robot (see Sec. III). Let then  $\hat{\mathbf{m}}_{c,i} \in \mathbb{R}^3$  be the  $i$ -th estimation of  $\mathbf{x}_i - \frac{1}{N} \sum_{j=1}^N \mathbf{x}_j$ . As detailed in Appendix A, the following decentralized estimation algorithm

$$\dot{\hat{\mathbf{d}}}_{c,i} = k_c \sum_{j \in \mathcal{N}_i} (\hat{\mathbf{d}}_{c,j} - \hat{\mathbf{d}}_{c,i}) - (\mathbf{x}_j - \mathbf{x}_i), \quad k_c > 0, \quad (19)$$

ensures that  $\hat{\mathbf{d}}_{c,i} \rightarrow \mathbf{x}_i - \frac{1}{N} \sum_{j=1}^N \mathbf{x}_j$ . It is important to highlight that (19) only requires availability of the relative robots positions and the communication of  $\hat{\mathbf{d}}_{c,j}$  by neighboring robots.

Finally, let  $\hat{\lambda}_{2,i}^{\max} = \hat{\lambda}_{2,i}^{\min} + \Delta$  be the estimation of  $\lambda_{2,i}^{\max}$  computed by robot  $i$  (see end of Sec. III-C),  $\sigma_c \in \{0, 1\}$  a binary activation/deactivation parameter for this expanding feature, and  $\bar{v}_c > 0$  the desired speed for moving away from the centroid. If the expansion behavior is activated, the control

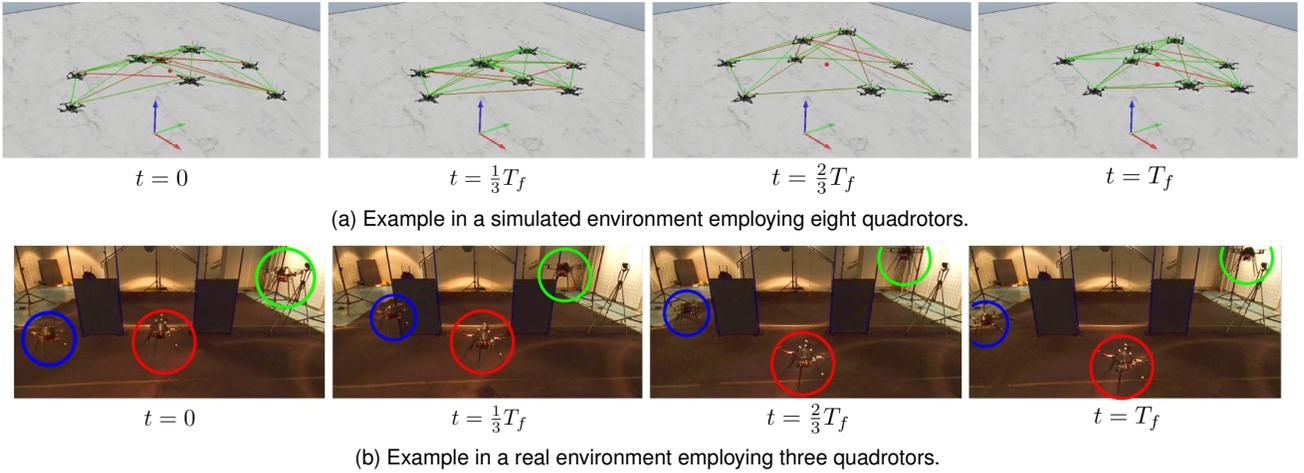


Fig. 7. Formation expansion behavior. (a) Increasing  $\mu_\lambda$  compacts the formation ( $t = 1/3 T_f$ ), decreasing  $\mu_\lambda$  expands the formation w.r.t. the centroid ( $t = 2/3 T_f$ ), and increasing  $\mu_\lambda$  one more time compacts the formation again ( $t = T_f$ ). The centroid is depicted as a red point.  $N = 8$ ,  $T_f = 90$  s. Video available at [https://youtu.be/\\_eMsbMlmqaU?t=106](https://youtu.be/_eMsbMlmqaU?t=106). (b) The same demonstration as in (a) is performed using three quadrotors in a real environment. Each robot is highlighted with a colored circle;  $N = 3$ ,  $T_f = 76$  s. Video available at [https://youtu.be/\\_eMsbMlmqaU?t=139](https://youtu.be/_eMsbMlmqaU?t=139).

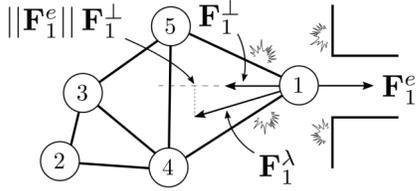


Fig. 8. Formation deadlock example (top view). If the formation needs to pass through a narrow passage and  $\lambda_2^{\min}$  is too high, the formation may stall. The connectivity force  $\mathbf{F}_1^\lambda$  acting on the leader may completely counterbalance the exogenous force  $\mathbf{F}_1^e$  commanded by the human operator, resulting in a stall of the leader and, in turn, of the whole formation.

input (8) is modified as

$$\mathbf{u}_i = \mathbf{F}_i^\lambda - \mathbf{B}_i \mathbf{v}_i + \mathbf{F}_i^e - \lambda_c \frac{\hat{\mathbf{d}}_{c,i}}{\|\hat{\mathbf{d}}_{c,i}\|}, \quad k_v > 0, \quad (20)$$

with

$$\lambda_c = \begin{cases} \bar{v}_c & \text{if } \hat{\lambda}_2^i(t) \geq \hat{\lambda}_{2,i}^{\max} \text{ and } \sigma_c = 1 \\ 0 & \text{otherwise} \end{cases}. \quad (21)$$

#### D. Automatic deadlock resolution

*Motivation (see Fig. 8 and [https://youtu.be/\\_eMsbMlmqaU?t=153](https://youtu.be/_eMsbMlmqaU?t=153)):* if the robots need to pass through a narrow passage and  $\lambda_2^{\min}$  is too high, the formation may become stalled. In fact, the connectivity force  $\mathbf{F}_1^\lambda$  acting on the leader may completely counterbalance the exogenous force  $\mathbf{F}_1^e$  commanded by the human operator, resulting in a stall for the leader and, in turn, for the whole formation.

A straightforward solution to this issue is lowering  $\lambda_2^{\min}$  for allowing a stretching/loosening of the formation. However, we found this solution not always desirable, since it is often not straightforward to recognize this situation and understand how much to lower  $\lambda_2^{\min}$  for resolving the deadlock. Moreover, even when the operator resolves the deadlock by acting solely on  $\lambda_2^{\min}$ , the maneuver can take a significant amount of time.

To address this undesirable circumstance, we therefore introduce an automatic deadlock resolution feature that can detect a deadlock situation, lower  $\lambda_2^{\min}$  until the leader is unlocked, and then increase  $\lambda_2^{\min}$  again once the deadlock is over. To this end, we introduce the quantity

$$F_1^\perp = \frac{(\mathbf{F}_1^\lambda)^T \mathbf{F}_1^e}{\|\mathbf{F}_1^e\|} \cdot \frac{1}{\|\mathbf{F}_1^\lambda\|} = \frac{(\mathbf{F}_1^\lambda)^T \mathbf{F}_1^e}{\|\mathbf{F}_1^\lambda\|^2}, \quad (22)$$

which represents the ratio between the projection of  $\mathbf{F}_1^\lambda$  on the direction of  $\mathbf{F}_1^e$  and the norm of  $\mathbf{F}_1^e$ . If  $F_1^\perp \geq 0$ , the two forces  $\mathbf{F}_1^\lambda$  and  $\mathbf{F}_1^e$  are not competing; if  $-1 < F_1^\perp < 0$ , the two forces are competing but  $\mathbf{F}_1^e$  is still able to overcome  $\mathbf{F}_1^\lambda$  (and thus the leader can still move); and, if  $F_1^\perp = -1$ , the connectivity force  $\mathbf{F}_1^\lambda$  fully counterbalances the external force  $\mathbf{F}_1^e$ , stalling the formation (as in Figs. 8 and 9(a)). Exploiting this quantity, we then define an activation function

$$\varphi(F_1^\perp) = w_0^1(-F_1^\perp, F_{\min}^\perp, F_{\max}^\perp), \quad (23)$$

where, as usual,  $0 < F_{\min}^\perp < F_{\max}^\perp < 1$  are activation/deactivation thresholds.

Function  $\varphi(F_1^\perp)$  can now be used for automatically changing  $\lambda_2^{\min}$  whenever  $F_1^\perp$  becomes too close to  $-1$ . Recalling Sec. III-C,  $\lambda_2^{\min}$  is obtained by propagating via (10) the desired increment  $\mu_\lambda$  (received by the leader) and then using (9) to add the (saturated) increment to the minimum connectivity  $\lambda_m$ . We propose to modify (10) as

$$\begin{cases} \dot{\hat{\mu}}_i = \sum_{j \in \mathcal{N}_i} (\hat{\mu}_j - \hat{\mu}_i), & i = 2, \dots, N \\ \dot{\hat{\mu}}_1 = \sum_{j \in \mathcal{N}_1} (\hat{\mu}_j - \hat{\mu}_1) + b_\mu (\mu_\lambda \varphi(F_1^\perp) - \hat{\mu}_1), & b_\mu > 0 \end{cases}, \quad (24)$$

i.e., by replacing the desired increment  $\mu_\lambda$  with a scaled version  $\mu_\lambda \varphi(F_1^\perp)$ , which approaches zero as  $F_1^\perp \rightarrow -1$ . In this way,  $\lambda_2^{\min} \rightarrow \lambda_m$  as the group approaches a deadlock situation. It is again important to highlight that (24) is decentralized, since  $\varphi(F_1^\perp)$  can be computed locally by the leader robot.

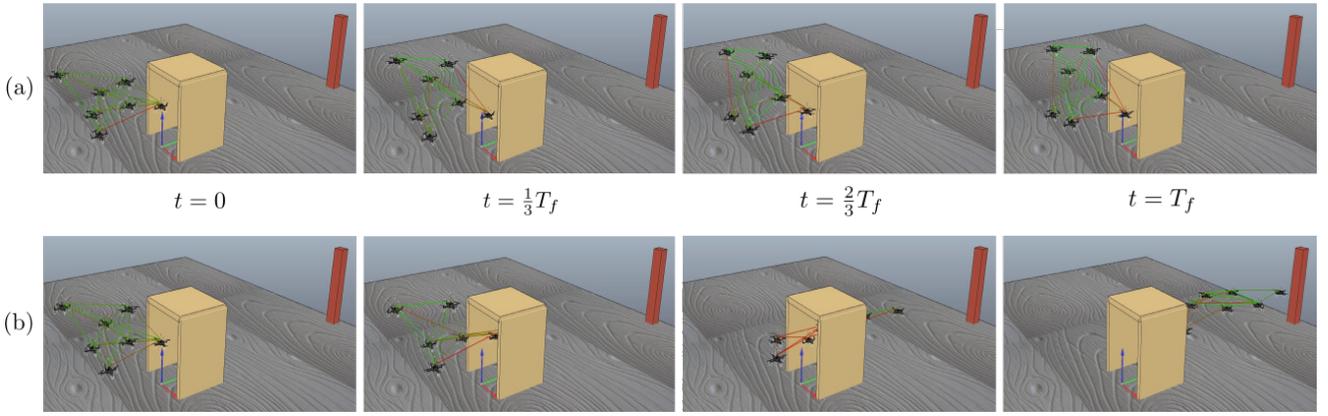


Fig. 9. Automatic deadlock resolution. (a) Whenever a formation with a high value for  $\lambda_2^{\min}$  has to pass through a narrow passage, the leader robot may stall, i.e., the connectivity force  $\mathbf{F}_1^\lambda$  acting on the leader may completely counterbalance the exogenous force  $\mathbf{F}_e^1$  commanded by the human operator preventing the formation to proceed;  $T_f = 33$  s. (b) Activating the proposed automatic deadlock resolution feature,  $\lambda_{\min}^{\text{curr}}$  is reduced for each robot  $i$  to  $\lambda_{\gamma}$ , following the consensus on  $\mu_1$ . This strategy allows the removal of links, resulting in a lower  $\mathbf{F}_1^\lambda$  and allowing the team to move again;  $T_f = 87$  s. Video available at [https://youtu.be/\\_eMsbMlmqaU?t=153](https://youtu.be/_eMsbMlmqaU?t=153).

## V. DEMONSTRATION OF THE ALGORITHM

In this Section, we analyze the effects of each improvement on the behavior of the formation w.r.t. the previous connectivity control framework [4], [5].

We perform four simulations using the V-REP robot simulator (Coppelia Robotics GmbH, CH). The control architecture is implemented using Simulink by MATLAB (USA) on a GNU/Linux machine (16 GB RAM,  $4 \times 2.60$  GHz Intel Core i7-6600U, Intel HD Graphics 520), using the Robot Operating System (ROS) framework as a bridge between Simulink and V-REP. The virtual 3D scene is populated with a fleet of six, eight, or seven quadrotors, depending on the case (see below). The visualization refresh rate and the control loop are set to 200 Hz. As all our algorithms are designed to be decentralized, each drone only needs to know the *relative* position of neighboring agents. Such a measure can be obtained through, e.g., camera, laser, sonar, or infrared sensors mounted on the robots. In this demonstration, and in the experiment of Sec. VI-A, we simulate the presence of sonars able to provide each drone with this information, i.e., we did not use any centralized information to track the robots.

We also perform one demonstration in a real-world setup. In this case, we employ three quadrotors from MikroKopter GmbH (HiSystems GmbH, DE). These robots are controlled at 50 Hz. In this demonstration, and in the experiment of Sec. VI-B, for the sake of simplicity but without any loss of generality, we retrieved the UAVs positions at 100 Hz using a centralized optical tracking system (Vicon Motion Systems Ltd, UK) composed of twelve cameras. However, as before, each robot is provided *only* with the relative positions of the surrounding agents, keeping the overall framework decentralized.

Table II reports the parameters used in these demonstrations.

### A. Airflow disturbance avoidance

This feature enables the avoidance of dangerous situations due to the airflow disturbances. In order to address this issue, in Sec. IV-A we have introduced a new weight  $\phi_{ij}$  which prevents two quadrotors from flying too close when on top of each other, so as to avoid the occurrence of dangerous situations. The

positive effects of this weight can be seen at [https://youtu.be/\\_eMsbMlmqaU?t=450](https://youtu.be/_eMsbMlmqaU?t=450). With the proposed approach, whenever two robots are close to a dangerous configuration, they naturally move to a safer configuration by, e.g., flying higher/lower or sideways. This behavior is always active and can be appreciated in several instances during the experiments of Sec. VI.

### B. Consensus on the leader velocity

This feature enables the operator to ask all the robots in the group to track the velocity of the leader, thus avoiding an excessive stretching of the team especially when moving in cluttered environments. Fig. 5 reports an illustrative example of this feature, also shown in [https://youtu.be/\\_eMsbMlmqaU?t=66](https://youtu.be/_eMsbMlmqaU?t=66). The operator moves the robotic team across a cluttered scene. The simulation shows the results of three different maneuvers: (a) using a small  $\lambda_2^{\min}$  with no consensus on the leader velocity leads to a stretched formation; (b) using a high  $\lambda_2^{\min}$  with no consensus on the leader velocity leads to a team which is compact but hard to maneuver; (c) using a small  $\lambda_2^{\min}$  with consensus on the leader velocity leads to a compact, yet flexible and easy-to-maneuver formation. In our implementation, this feature can be activated/deactivated by the user whenever necessary.

### C. Formation expansion

This feature enables the operator to expand the formation w.r.t. the team centroid. Fig. 7a shows an illustrative example in simulation. At first, the desired increment  $\mu_\lambda$  is increased, compressing the formation (from  $t = 0$  to  $t = 1/3 T_f$ ). At this point, even if  $\mu_\lambda$  is decreased again, the team would not go back to its initial stretched configuration. To achieve this desired behavior, the operator activates the proposed formation expansion feature, commanding the robots to expand from the centroid (from  $t = 1/3 T_f$  to  $t = 2/3 T_f$ ). Once the expanded formation has reached the limit condition of (21), increasing and decreasing  $\mu_\lambda$  causes a compression and decompression of the formation (from  $t = 2/3 T_f$  to  $t = T_f$ ), respectively. Fig. 7b shows the same behavior in a real-world scenario. This feature can be activated/deactivated by the user whenever deemed necessary.

TABLE II  
PARAMETERS SETTING (J\* INDICATES BOTH JN AND JV MODALITIES)

Common parameters			
$\lambda_m = 0.01$	$\mu_{\min} = 0$	$\mu^{\text{inc}} = 0.1$	$\Delta = 1$
$\bar{\mu}_{\min}^{\lambda} = 0.1$	$\bar{\mu}_{\max}^{\lambda} = 1$		
Connectivity-maintenance improvements; Sec V			
$d_{\min} = 0.5 \text{ m}$	$d_{\max} = 1.4 \text{ m}$	$d_{\min}^{\circ} = 0.55 \text{ m}$	$d_{\max}^{\circ} = 0.7 \text{ m}$
$D = 3.5 \text{ m}$	$d_1 = 2 \text{ m}$	$d_{\min}^{\phi} = 0.5 \text{ m}$	$D^{\phi} = 1.4 \text{ m}$
$c_{\min} = 0.98$	$c_{\max} = 0.5$	$F_{\min}^{\perp} = 0.1$	$F_{\max}^{\perp} = 0.4$
$b_i = 1$	$\bar{v}_c = 0.5 \text{ m/s}$	$b_{\mu} = 1$	$d_0 = 2 \text{ m}$
$b_i = 0.7$	$b_{\lambda} = 2.5$	$\eta = 0.3$	$\mu_{\max} = 5$
Experiments - Simulated environments; Sec VI-A			
$d_{\min} = 0.5 \text{ m}$	$d_{\max} = 1.75 \text{ m}$	$D = 3.5 \text{ m}$	$d_1 = 1.75 \text{ m}$
$d_{\min}^{\circ} = 0.5 \text{ m}$	$d_{\max}^{\circ} = 0.6 \text{ m}$	$d_{\min}^{\phi} = 0.5 \text{ m}$	$D^{\phi} = 1.3 \text{ m}$
$c_{\min} = 0.98$	$c_{\max} = 0.5$	$d_0 = 1.75 \text{ m}$	$\mu_{\max} = 5$
$\eta = 0.7$	$b_i = 0.7 \text{ (J*)}$	$b_i = 0.5 \text{ (OK)}$	$b_i = 3 \text{ (LC)}$
Experiments - Real environment; Sec VI-B			
$d_{\min} = 0.7 \text{ m}$	$d_{\max} = 1.5 \text{ m}$	$D = 3 \text{ m}$	$d_1 = 1.5 \text{ m}$
$d_0 = 1.5 \text{ m}$	$\mu_{\max} = 3$	$c_{\min} = 0.98$	$c_{\max} = 0.76$
$d_{\min}^{\circ} = 0.5 \text{ m}$	$d_{\max}^{\circ} = 0.6 \text{ m}$	$d_{\min}^{\phi} = 0.7 \text{ m}$	$D^{\phi} = 1.5 \text{ m}$
$b_i = 3 \text{ (LC)}$	$\eta = 0.7$		

#### D. Automatic deadlock resolution

This feature enables the operator to resolve possible deadlock situations due to conflicting  $\mathbf{F}_1^{\lambda}$  and  $\mathbf{F}_1^c$  forces. Fig. 9 illustrates how this feature works. The operator is asked to steer seven simulated robots through a small aperture (in yellow). Fig. 9a and [https://youtu.be/\\_eMsbMlmqaU?t=153](https://youtu.be/_eMsbMlmqaU?t=153) illustrates a deadlock situation: due to counterbalancing  $\mathbf{F}_1^{\lambda}$  and  $\mathbf{F}_1^c$  forces, the leader cannot pass through the opening without breaking the formation. By activating the proposed unlocking feature,  $\lambda_2^{\min}$  is automatically lowered until the leader can finally move. Then, once the formation has passed through the obstacle,  $\lambda_2^{\min}$  is automatically increased again until its original value. This behavior is visible in Fig. 9b and [https://youtu.be/\\_eMsbMlmqaU?t=173](https://youtu.be/_eMsbMlmqaU?t=173).

## VI. EXPERIMENTAL EVALUATION

To evaluate the effectiveness and viability of the proposed approach, we carried out two human subject experiments.

#### A. Experiment in a simulated environment

In this first experiment, users control a robotic team in a virtual environment simulated using V-REP. A video can be seen at [https://youtu.be/\\_eMsbMlmqaU?t=220](https://youtu.be/_eMsbMlmqaU?t=220).

1) *Virtual environment and task:* We consider two different virtual scenes, S1 and S2. The first one employs  $N = 5$  robots navigating inside a complex labyrinth-like structure which needs to be explored (see Fig. 14a). The user is required to steer the robotic team through the labyrinth until the end of the corridor (shown as a red dot in Fig. 14a). On the other hand, S2 considers  $N = 9$  robots deployed in a simpler environment (see Fig. 14b). The user is required to steer the robotic team either over or below three walls until it reaches the end of the circuit (shown as a red dot in Fig. 14b). In both cases, the virtual scene is displayed on an LCD screen posed in front of the user. It shows the environment from different points of

view, including a top view of the formation, a first-person view of the robot leader, and a bird's-eye prospective (see Fig. 14). We consider two different scenes to prove that our approach is quite general and can be either employed in cases with several robots as well as while navigating complex structures.

The virtual scenes are managed by one computer (6 GB RAM,  $4 \times 3.07$  GHz Intel Xeon CPU, Gallion 0.4 on NV94 graphic card), while the control loop computing  $\mathbf{A}$  and  $\mathbf{F}_i^{\lambda}(\mathbf{x}, \mathcal{O})$  runs on another computer (same specifications as in Sec. V). The communication between the two machines is done through Ethernet, with ROS acting as bridge. Both V-REP simulation and the control loop run at 200 Hz. The control parameters used in this evaluation are reported in Tab. II.

2) *Master interfaces:* The human user steers the robotic team by controlling the motion of the robot leader, as explained in Sec. III. We consider three different types of input devices:

(J) an off-the-shelf joystick Gamepad F310 (Logitech, CH), (O) a grounded haptic device Omega.6 (Force Dimension, CH), (L) a Leap Motion controller (Leap Motion Inc., USA).

In condition J, the analog thumbsticks of the joystick imparts velocities to the leader in 3D space (see Fig. 10). The left-hand thumbstick commands velocity along the  $x$  and  $y$  axes of the robot, while the right-hand thumbstick commands velocities along the  $z$  axis of the robot. We can define the interaction force for the leader robot as

$$\mathbf{F}_1^e(t) = \mathbf{B}_i (\mathbf{p}^J(t) - \mathbf{p}^J(t_0)), \quad (25)$$

where  $\mathbf{p}^J(t) \in \mathbb{R}^3$  is the input velocity from the joystick at time  $t$ , and  $\mathbf{B}_i \in \mathbb{R}^{3 \times 3}$  is a matrix whose diagonal contains the gains  $b_i \in \mathbb{R}^+$ . Finally, the lower shoulder buttons R2 and L2 respectively increase and decrease the external input  $\mu_{\lambda}(t)$  for the asymptote value of  $\lambda_2$  within the range  $\mu_{\lambda}(t) \in [\mu_{\min}, \mu_{\max}]$ .

In condition O, the Omega.6 end-effector is used to control the velocity of the robot leader. Let us define  $\mathbf{p}^O(t) \in \mathbb{R}^3$  as the position of the Omega.6 end-effector w.r.t. to a world reference frame (see Fig. 11). To enable a comfortable operation, the keyboard works as a clutch. Whenever the space key is not pressed, we save the Omega's position at that moment,  $\mathbf{p}^O(t_c)$ . Whenever the space key is pressed, the difference between the current position of the Omega and  $\mathbf{p}^O(t_c)$  is used to give input forces to the leader,

$$\mathbf{F}_1^e(t) = \mathbf{B}_i (\mathbf{p}^O(t) - \mathbf{p}^O(t_c)), \quad (26)$$

where  $\mathbf{B}_i \in \mathbb{R}^{3 \times 3}$  is again a matrix whose diagonal contains the gains  $b_i \in \mathbb{R}^+$ . This clutching approach takes inspiration from [41], [42], [43], and it has been proven to increase comfort. It allows the user to start the motion with the end-effector at an arbitrary position, enabling him to pause, move to a more comfortable or suitable position, and then resume the control of the robot. Finally, rotating the pen-shaped end-effector clockwise or counterclockwise respectively increases or decreases  $\mu_{\lambda}(t) \in [\mu_{\min}, \mu_{\max}]$ .

Condition L takes an approach very similar to O. Here,  $\mathbf{p}^L(t) \in \mathbb{R}^3$  defines the position of the user's hand w.r.t. to the world reference frame, as tracked by the Leap Motion, and  $\mathbf{p}^L(t_c)$  indicates the hand pose when the clutch is pressed (see



Fig. 10. Joypad used in conditions JN and JV. The left-hand thumbstick commands velocity along the  $x$  and  $y$  axes of the robot, while the right-hand thumbstick modifies its altitude. The two lower shoulder buttons R2 and L2 increases and decreases the external input  $\mu_\lambda(t)$  for the asymptote value of  $\lambda_2$ , respectively.

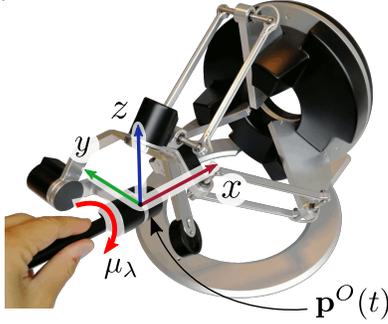


Fig. 11. The grounded haptic interface Omega.6 used in condition OK. Whenever the clutch is activated at time  $t_c$ , a displacement of the end-effector w.r.t. its initial position at  $t_c$  is translated into a velocity on the same axis for the robot. Moreover, rotating the pen end-effector increases (clockwise direction) or decreases (counterclockwise direction)  $\mu_\lambda(t)$ .

Fig. 12). We can then define

$$\mathbf{F}_1^e(t) = \mathbf{B}_i (\mathbf{p}^L(t) - \mathbf{p}^L(t_c)). \quad (27)$$

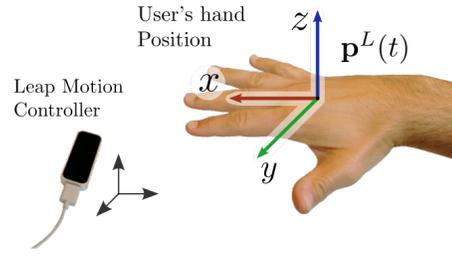
Finally, the closure of the user hand enables to control  $\mu_\lambda(t)$ : a fully-open hand sets  $\mu_\lambda(t) = \mu_{\min}$ , whereas a fully-closed hand sets  $\mu_\lambda(t) = \mu_{\max}$ . This way of controlling  $\mu_\lambda(t)$  takes inspiration from [27], where we also intuitively relate the action of closing the hand to a ‘‘closure’’ of the formation.

3) *Feedback modalities*: We also consider four different ways of providing the operator with feedback information about the connectivity of the team: no feedback, visual feedback, kinesthetic feedback, and cutaneous feedback. Due to the limited actuation capabilities of some of our input interfaces (e.g., the joystick) as well as time constraints, we do not test all the above feedback conditions with all the input interfaces of Sec. VI-A2. We select four representative experimental conditions:

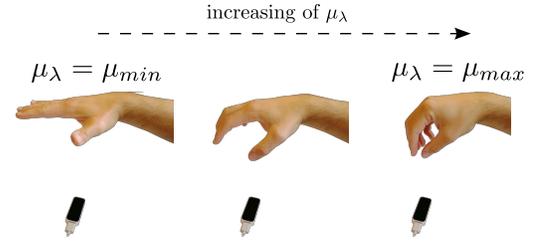
- (JN) the joystick controls the leader and we provide no feedback about the connectivity force,
- (JV) the joystick controls the leader and visual feedback provides information about the connectivity force,
- (OK) the Omega.6 controls the leader and kinesthetic feedback provides information about the connectivity force,
- (LC) the Leap Motion controls the leader and cutaneous feedback provides information about the connectivity force.

In condition JN, the leader is controlled with the joystick, as described in Sec. VI-A2, and the user receives no feedback about the connectivity force.

In condition JV, the leader is again controlled with the joystick, but this time the user receives information about the connectivity force through a visual representation of  $\mathbf{F}_1^\lambda$ : a



(a) Whenever the clutch is activated at time  $t_c$ , a displacement of the user's hand w.r.t. its initial position at  $t_c$  is translated into a velocity on the same axis for the leader robot.



(b) By closing and opening the hand, the user controls  $\mu_\lambda$ .

Fig. 12. The Leap Motion tracker used in condition LC. Whenever the clutch is activated at time  $t_c$ , a displacement of the user's hand w.r.t. its position at  $t_c$  is translated into a velocity on the same axis to be given to the robot. Closing and opening the hand increases and decreases  $\mu_\lambda(t)$ , respectively.

blue segment spawns from the leader along the direction of  $\mathbf{F}_1^\lambda$ . This information enables the operator to know how much he is deviating from the current connectivity level, i.e., a high  $\|\mathbf{F}_1^\lambda\|$  means that the current action results in a significant decrease of  $\lambda_2$ . In addition to  $\mathbf{F}_1^\lambda$ , we also provide the user with information about the difference between the input for the asymptote  $\mu_\lambda(t)$  and the current estimate of  $\lambda_2^{\min}$  calculated by the leader,  ${}^1\lambda_{\min}^{\text{curr}}(t)$ . This difference  $\bar{\mu}_\lambda(t) = {}^1\lambda_{\min}^{\text{curr}}(t) - (\lambda_m + \mu_\lambda(t))$  is plotted on the screen in front of the user, next to the different views of the virtual environment (see Sec. VI-A1).

In condition OK, the leader is controlled with the Omega.6, as described in Sec. VI-A2, which also provides the user with kinesthetic feedback about the connectivity-maintenance force  $\mathbf{F}_1^e$ . Therefore, the user feels the end-effector of the Omega.6 moving toward the direction of the connectivity force. As in JV, we also provide the user with information about the difference between the input for the asymptote  $\mu_\lambda(t)$  and the current estimate of  $\lambda_2^{\min}$  calculated by the leader,  ${}^1\lambda_{\min}^{\text{curr}}(t)$ . We define this difference as  $\bar{\mu}_\lambda(t) = {}^1\lambda_{\min}^{\text{curr}}(t) - (\lambda_m + \mu_\lambda(t))$ , and

$$F^o(\bar{\mu}_\lambda) = k_\mu w_1^0(\bar{\mu}_\lambda(t), \bar{\mu}_{\min}^\lambda, \bar{\mu}_{\max}^\lambda) \quad (28)$$

as our weight function, with  $0 < \bar{\mu}_{\min}^\lambda < \bar{\mu}_{\max}^\lambda$ , being appropriate thresholds for  $\bar{\mu}_\lambda(t)$ , and  $k_\mu = 0.02$  (see (3) and Fig. 2a for details on the weight function). Whenever  $\bar{\mu}_\lambda(t) > \bar{\mu}_{\min}^\lambda$ , the Omega provides the user with vibrotactile stimuli of frequency 100 Hz and intensity  $F^o(\bar{\mu}_\lambda(t))$ .

In condition LC, the operator controls the robot leader with a Leap Motion controller, as described in Sec. VI-A2, and receives information about  $\mathbf{F}_1^e$  through a wearable cutaneous device worn on the forearm. The wearable device is able to provide skin stretch, pressure, and vibrotactile stimuli. It is equipped with two servo motors, that actuate an elastic belt wrapped around the arm, and four vibrotactile motors, positioned evenly around the arm (see Fig. 13). Its design has

been presented in [27]. For rendering information about  $\mathbf{F}_1^e$ , we use the four vibrating motors. Each independently-controlled motor  $m \in \{C_T, C_L, C_D, C_R\}$  has an effective vibration frequency range  $f_m \in [0 - 280]$  Hz and a square-wave-like vibration pattern  $\kappa_m$ , i.e., when motor  $m$  is activated, it vibrates at frequency  $f_m$  for a period of  $2\kappa_m$  with duty cycle 50%. Let us define the select function for each direction  $i = \{x, y, z\}$  of  $\mathbf{F}_1^e$  as

$$d_i^F = \begin{cases} 0 & |F_{1,i}^e| \leq f_{\min} \\ \max(\min(F_{1,i}^e/F_{\max}, 1), -1) & \text{else} \end{cases},$$

where  $F_{\max} > 0$  is the maximum force we can apply, and  $f_{\min} > 0$  is a force threshold below which we do not activate the feedback. We use the approach described in [27], [44] to map the 3D vector force  $\mathbf{F}_1^e$  into inputs for the four vibrotactile motors. Left  $C_L$  and right  $C_R$  motors are used to provide information about  $\mathbf{F}_1^e$  along the  $y$ -axis (right/left),

$$\begin{cases} f_L = 0, f_R = -280d_y^F, \kappa_m = 200 \text{ ms} & \text{if } \text{sgn}(d_y^F) < 0 \\ f_R = 0, f_L = 280d_y^F, \kappa_m = 200 \text{ ms} & \text{if } \text{sgn}(d_y^F) > 0 \end{cases},$$

whereas for force components along the  $x$  and  $z$  axis (top/down and front/back), we use motors  $C_T$  and  $C_D$ . Top motor  $C_T$  is activated when the force points up *or* forward, while down motor  $C_D$  is activated when the force points down *or* backwards [44]. Of course, motors  $C_T$  and  $C_D$  need different vibration patterns for the two different directions they convey. This is achieved by changing the interval between each vibration. In particular,  $C_T$  will vibrate in bursts of  $\kappa_T = 100$  ms when the force vector points up and of  $\kappa_T = 200$  ms when it points forward. Similarly,  $C_D$  will vibrate in bursts of  $\kappa_D = 100$  ms when the force vector points backwards and of  $\kappa_D = 200$  ms when it points down. To avoid sudden and possibly confusing changes between up/forward or down/backward stimuli, we weight the forward/backward feedback more than the top/down feedback by multiplying the former of  $\zeta = 1.5$ , i.e.,

$$\begin{cases} f_T = 280d_x^F, \kappa_T = 100 \text{ ms} & \text{if } d_z^F = 0 \text{ and } \delta d_x^F > f_{\min} \text{ or} \\ & d_z^F \neq 0 \text{ and } \zeta d_x^F \geq |d_z^F| \\ f_T = -280d_x^F, \kappa_T = 200 \text{ ms} & \text{if } d_z^F = 0 \text{ and } d_x^F < -f_{\min} \text{ or} \\ & d_x^F \neq 0 \text{ and } d_z^F < |\zeta d_x^F| \end{cases},$$

$$\begin{cases} f_D = -280d_z^F, \kappa_D = 100 \text{ ms} & \text{if } d_x^F = 0 \text{ and } \delta d_z^F < -f_{\min} \text{ or} \\ & d_x^F \neq 0 \text{ and } \zeta d_z^F \leq |d_x^F| \\ f_D = 280d_z^F, \kappa_D = 200 \text{ ms} & \text{if } d_x^F = 0 \text{ and } d_z^F > f_{\min} \text{ or} \\ & d_x^F \neq 0 \text{ and } d_z^F > |\zeta d_x^F| \end{cases},$$

This approach has been already proven capable of providing 3D information [44]. Finally, we provide information on the connectivity difference  $\bar{\mu}_\lambda(t) = {}^1\lambda_{\min}^{\text{curr}}(t) - (\lambda_m + \mu_\lambda(t))$  using the pressure/squeeze feedback of the wearable device. In particular, the device will squeeze the user's arm of  $F^o(\bar{\mu}_\lambda(t))$  mm, with  $k_\mu = 76.5$  (see (28)).

*Subjects:* Twelve participants took part in the experiment, including one woman and eleven men (age 23–32 years old). One practice trial was allowed. Users performed one repetition of the task per feedback condition per scenario, yielding a total of 8 trials per user and 96 trials in total. Users were asked

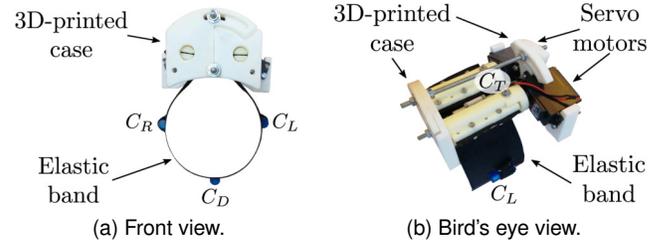
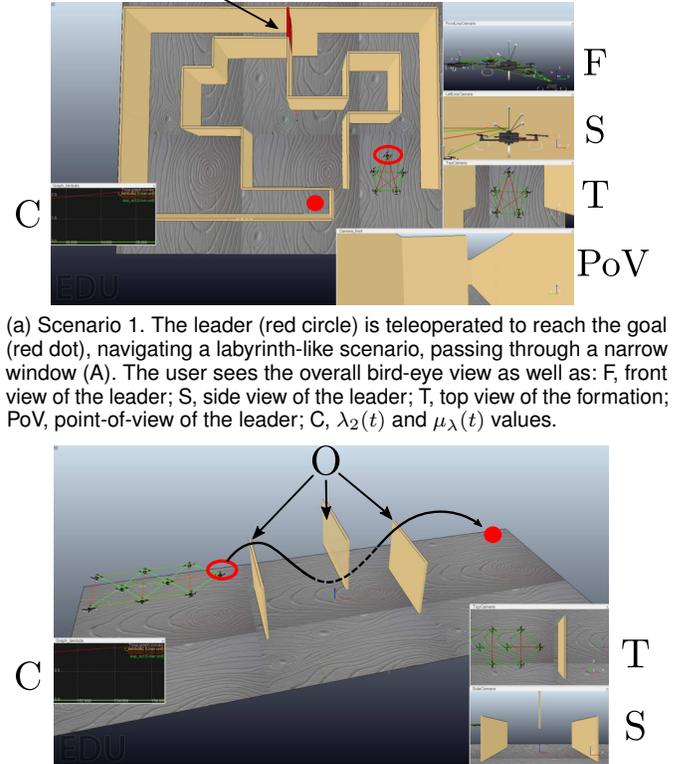


Fig. 13. Wearable cutaneous device used in condition LC. It is equipped with two servo motors, actuating an elastic belt wrapped around the arm, and four vibrotactile motors positioned evenly around the arm. Vibrotactile feedback is used to provide information about  $\mathbf{F}_1^\lambda$  and squeeze feedback is used to provide information about  $\bar{\mu}_\lambda(t)$ .



(a) Scenario 1. The leader (red circle) is teleoperated to reach the goal (red dot), navigating a labyrinth-like scenario, passing through a narrow window (A). The user sees the overall bird-eye view as well as: F, front view of the leader; S, side view of the leader; T, top view of the formation; PoV, point-of-view of the leader; C,  $\lambda_2(t)$  and  $\mu_\lambda(t)$  values.

(b) Scenario 2. The leader robot (red circle) is teleoperated to reach the goal (red dot), navigating above and below three obstacles (O). The user sees the overall bird-eye view as well as: T, top view of the formation, centred on the leader; S, side view of the scene, fixed; C,  $\lambda_2(t)$  and  $\mu_\lambda(t)$  values.

Fig. 14. Experiment in a simulated environment. We considered two teleoperation scenarios: (a) S1, five robots navigate a labyrinth-like environment; (b) S2, nine robots are asked to perform a pass-above-and-below task. Video available at [https://youtu.be/\\_eMsbMlmaqU?t=220](https://youtu.be/_eMsbMlmaqU?t=220).

to complete the task as fast as possible, taking however into account the feedback received.

*Results:* We registered (i) the task completion time, (ii) the connectivity force commanded to the robots, (iii) the average total force commanded to the robots, and (iv) the maximum total force commanded to the robots. A low value of these metrics denotes the best performance. To compare them, we ran two-way repeated-measures ANOVA tests. Feedback conditions (JN, JV, OK, LC) and scenarios (S1, S2) were treated as within-subject factors. All data passed the Shapiro-Wilk normality test. Sphericity was assumed for variables with only two levels of repeated measures (i.e., scenario). A Greenhouse-Geisser correction was applied to tests involving

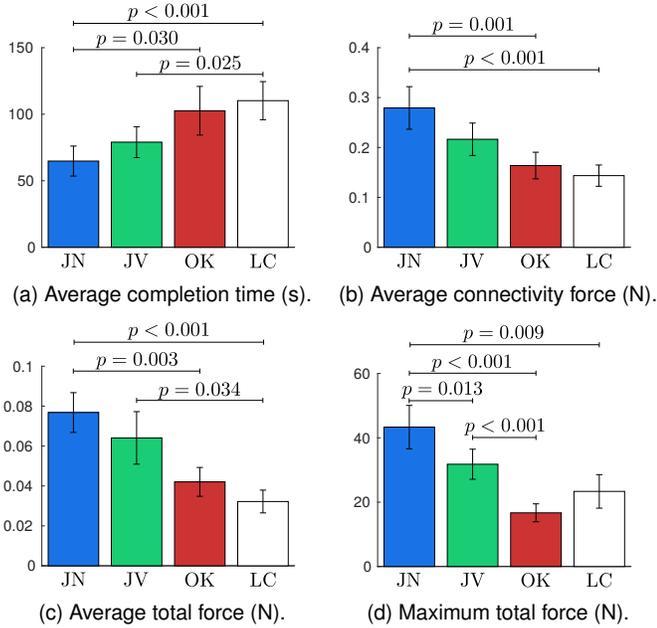


Fig. 15. Experiment in a simulated environment. Objective measures. Mean and 95% confidence interval of the (a) average task completion time, (b) average connectivity force, (c) average and (d) maximum commanded force.

data that violate the sphericity assumption. When the ANOVA reported a statistically significant difference, we ran post hoc analysis with Bonferroni adjustments. Statistically significant  $p$  values of this post-hoc comparison are reported in Fig. 15.

Fig. 15a shows the average completion time. The task starts when the leader moves for the first time and finishes when all the robots reach the end of the circuit. Mauchly's Test of Sphericity indicated that the assumption of sphericity was violated for the feedback condition variable ( $\chi^2(5) = 12.715, p = 0.027$ ). The two-way repeated-measure ANOVA with Greenhouse-Geisser correction revealed a statistically significant change in completion time for scenario ( $F(1, 11) = 7.649, p = 0.018$ ) and feedback condition ( $F(1.888, 20.766) = 10.627, p = 0.001$ , partial  $\eta^2 = 0.491$ ). Fig. 15b shows the average connectivity force commanded to the robots. It is calculated as the mean over time of

$$\sum_{i=1}^N \frac{\|\mathbf{F}_i^\lambda(\mathbf{x}, \mathcal{O})\|}{N},$$

where  $N$  is the number of robots in the scene (5 in S1 and 9 in S2). The data passed the Mauchly's Test of Sphericity. The two-way repeated-measure ANOVA revealed a statistically significant change in completion time for scenario ( $F(1, 11) = 9.061, p = 0.012$ ) and feedback condition ( $F(3, 33) = 14.256, p < 0.001$ ). Fig. 15c shows the average total force commanded to the robots. It is calculated as the mean over time of  $\sum_{i=1}^N \|\mathbf{u}_i\|/N$ . Mauchly's Test of Sphericity indicated that the assumption of sphericity was violated for the feedback condition variable ( $\chi^2(5) = 15.221, p = 0.010$ ). The two-way repeated-measure ANOVA with Greenhouse-Geisser correction revealed a statistically significant change in completion time for feedback condition ( $F(1.928, 21.208) = 13.754, p < 0.001$ ). Fig. 15d shows the maximum force commanded to the robots. It

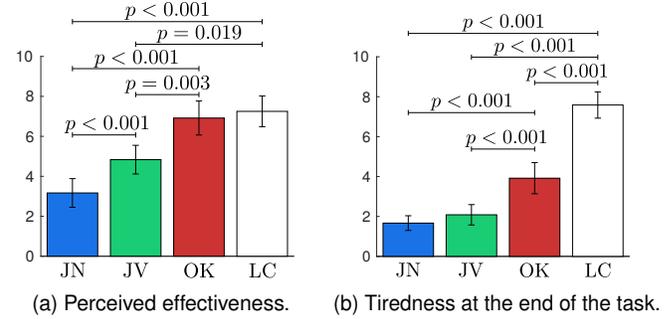


Fig. 16. Experiment in a simulated environment. Subjective measures. Mean and 95% confidence interval of the perceived (a) effectiveness and (b) tiredness at the end of the task.

is calculated as the maximum value registered during each trial for  $\mathbf{u}_i, i = 1 \dots, N$ . Mauchly's Test of Sphericity indicated that the assumption of sphericity was violated for the feedback condition variable ( $\chi^2(5) = 12.800, p = 0.026$ ). The two-way repeated-measure ANOVA with Greenhouse-Geisser correction revealed a statistically significant change in completion time for scenario ( $F(1, 11) = 27.038, p < 0.001$ ) and feedback condition ( $F(1.548, 17.032) = 26.211, p < 0.001$ ). This latter metric is particularly relevant, as it provides information on the maximum acceleration/thrust requested to the robots. A failure in providing such acceleration, even for a short period of time, may result in a loss of connectivity for the team. While this situation cannot happen in our simulated scenario, where the drones can apply infinite thrusts, it may happen when implementing the system in a real scenario. This point is also discussed in Sec. VII.

Immediately after the experiment, we also measured users' experience. Participants were asked to rate each feedback using a slider that ranged from 0 to 10, where a score of 0 meant "very low" and a score of 10 meant "very high". This time we ran one-way repeated-measures ANOVA tests. Feedback conditions (JN, JV, OK, LC) were treated as within-subject factor. All data passed the Shapiro-Wilk normality test. Again, when the ANOVA reported a statistically significant difference, we ran post hoc analysis with Bonferroni adjustments. Statistically significant  $p$  values are again reported in Fig. 16.

Figure 16a shows the user's perceived effectiveness. Mauchly's Test of Sphericity indicated that the assumption of sphericity was violated ( $\chi^2(5) = 16.023, p = 0.007$ ). The one-way repeated-measure ANOVA with Greenhouse-Geisser correction revealed a statistically significant difference between the means ( $F(1.637, 18.004) = 31.713, p < 0.001$ ). Figure 16b shows how much each user felt tired at the end of the task. The data passed the Mauchly's Test of Sphericity. The one-way repeated-measure ANOVA revealed a statistically significant difference between the means ( $F(3, 33) = 207.941, p < 0.001$ ). Finally, subjects were asked to choose the condition and the way to control  $\mu_\lambda(t)$  they preferred the most. Condition OK was preferred by eight subjects, while condition LC was preferred by four subjects. On the other hand, for controlling the connectivity lower limit  $\mu_\lambda(t)$ , ten subjects preferred closing the hand (LC) while two people preferred using the joystick (JN and JV).

Finally, we also asked users if they had any additional remark or suggestion. Five subjects out of twelve said that they mostly

focused on the general view of the scene, making little use of the other views. Four subjects said that they appreciated haptic feedback for its capability of providing additional information without overloading the visual channel.

From these results, we can see that conditions providing haptic feedback (OK and LC) behaved very similarly in most metrics. To determine whether their difference can be truly considered statistically negligible, we used the two one-sided t-test approach (TOST) [45], [46], [47]. In fact, we recall that a lack of statistical difference in the ANOVA does not mean equivalence. We carried out this equivalence testing between conditions OK and LC for all metrics. The tests revealed statistical equivalence between the two haptic conditions for all metrics but maximum total force and tiredness.

### B. Experiment in a real environment

We also tested our system in a real environment. Two expert operators carried out two indoor teleoperation experiments, considering either three and four quadrotors. A video can be seen at [https://youtu.be/\\_eMsbMImqaU?t=405](https://youtu.be/_eMsbMImqaU?t=405).

*Real environment and task:* We use quadrotors from MikroKopter GmbH (HiSystems GmbH, Germany), controlled at 50 Hz using two computers. The first machine (16 GB RAM,  $4 \times 2.80$  GHz Intel Xeon CPU E5-1603 CPU, NVIDIA Quadro K2000) controls the quadrotors using the Robotics Open Source Software (CNRS/LAAS, France) over Wi-Fi, while the second machine (16 GB RAM,  $4 \times 2.60$  GHz Intel Core i7-6600U CPU, Intel HD Graphics 520) computes the robots position references as per our control algorithm of Sec. III. The two machines communicate through ROS. Robots positions are registered using an optical tracking system composed of twelve cameras, running at 100 Hz. Robots fly in a  $8 \times 8 \times 3$  m room delimited by a safety net, as shown in Fig. 17. Two  $1 \times 3$  m panels are placed in the middle of the room, separated by a 2-m-wide passage. The configuration of the room and the position of the obstacles were considered known. The task consisted in moving the robots across the room and back, passing first between the two panels and then either on their right or left. The parameters used for this experiment are again those reported in Table II.

*Master interfaces and feedback modalities:* For this experiment, we only considered condition LC (see Sec. VI-A), as it performed the best in our previous experiment and was also one of the most appreciated. We performed the task twice per user, with three and four quadrotors in the room.

*Subjects:* Two expert operators took part in this experiment (age 30–31, both men). They both knew the connectivity control algorithm and the meaning of the feedback, as well as how to control the robots. No practice trial was allowed. The two operators were asked to perform one repetition of the task per each scenario. Users were asked to complete the task as fast as possible, taking however in account the feedback received. Each user's session lasted around 15 minutes.

*Results:* Figures 17 and 18 show the robots trajectories for experiments with three and four quadrotors, respectively. Specifically, Figs. 17a and 18a show the grid of obstacles known to the robots (left-hand side) as well as the robots trajectories at time  $t = T_f/2$  and  $t = T_f$  (central and right-hand sides). Each robot is identified with a color: red for the

leader robot (robot 1), green for robot 2, blue for robot 3, and cyan for robot 4. The circles indicate the robots starting position. Figs. 17b and 18b then show three shots of the scene, taken from two different cameras. Both operators greatly appreciated the control interface and feedback modality.

## VII. DISCUSSION AND CONCLUSIONS

This paper presents a decentralized connectivity-maintenance strategy for the teleoperation of a team of UAVs, together with an experimental evaluation in virtual and real environments.

The proposed connectivity-maintenance framework is inspired by the algorithm of [4], [5], which is here enhanced along several directions, with the objective of improving its applicability, effectiveness, and ease of use. The first improvement introduces an airflow-avoidance technique. It prevents the occurrence of dangerous stack downwash phenomena when using rotor-based UAVs. The second improvement implements a consensus-based velocity control. It enables all follower robots to track the leader's velocity. With respect to simply compacting the team by increasing  $\lambda_2^{\min}$ , this feature enables fast movements while maintaining a high flexibility and minimal change of topology of the formation. The third improvement enables the automatic decrease of the connectivity eigenvalue minimum asymptote, with the objective of achieving a dynamic expansion of the formation. When this feature is active, the formation tends to always to cover as much ground as possible, which can be useful in applications of surveillance and mapping. Of course, the user can still control the maximum expansion by acting on  $\lambda_2^{\min}$ . Finally, the fourth improvement is the automatic detection and resolution of deadlock configurations. These situations are caused by conflicts between the connectivity force (which comprises various actions, such as collision avoidance) and the external force acting on the leader, which can literally prevent the leader from moving. They mostly happen when steering the team in cluttered environments. The airflow disturbance avoidance is always active, while the other three new features can be activated/deactivated at the operator's will.

The first experiment is described in Sec. VI-A. We enrolled twelve naive participants to control a team of drones in two virtual reality scenarios. We tested four experimental conditions, considering different ways of controlling the robot leader (joystick, JN and JV; grounded haptic interface, OK; Leap Motion, LC) and of providing feedback information about the status of the team (no feedback, JN; visual representation, JV; kinesthetic feedback, OK; wearable cutaneous feedback, LC).

Fig. 15 shows that providing haptic feedback about the connectivity force, either through the grounded (OK) or the wearable (LC) device, significantly improved all metrics except completion time. Providing the same information through visual feedback (JV) did not lead to the same improvement, meaning that haptic feedback is more effective than visual feedback at conveying this type of information. However, conditions OK and LC also led to higher completion times, meaning that users took their time to follow the suggestions provided by the haptic devices. This behavior is of course expected and desirable, as we want users to pay attention to the connectivity force and act accordingly. On the other hand, surprisingly, providing visual feedback (JV) did not produce the same effect, meaning

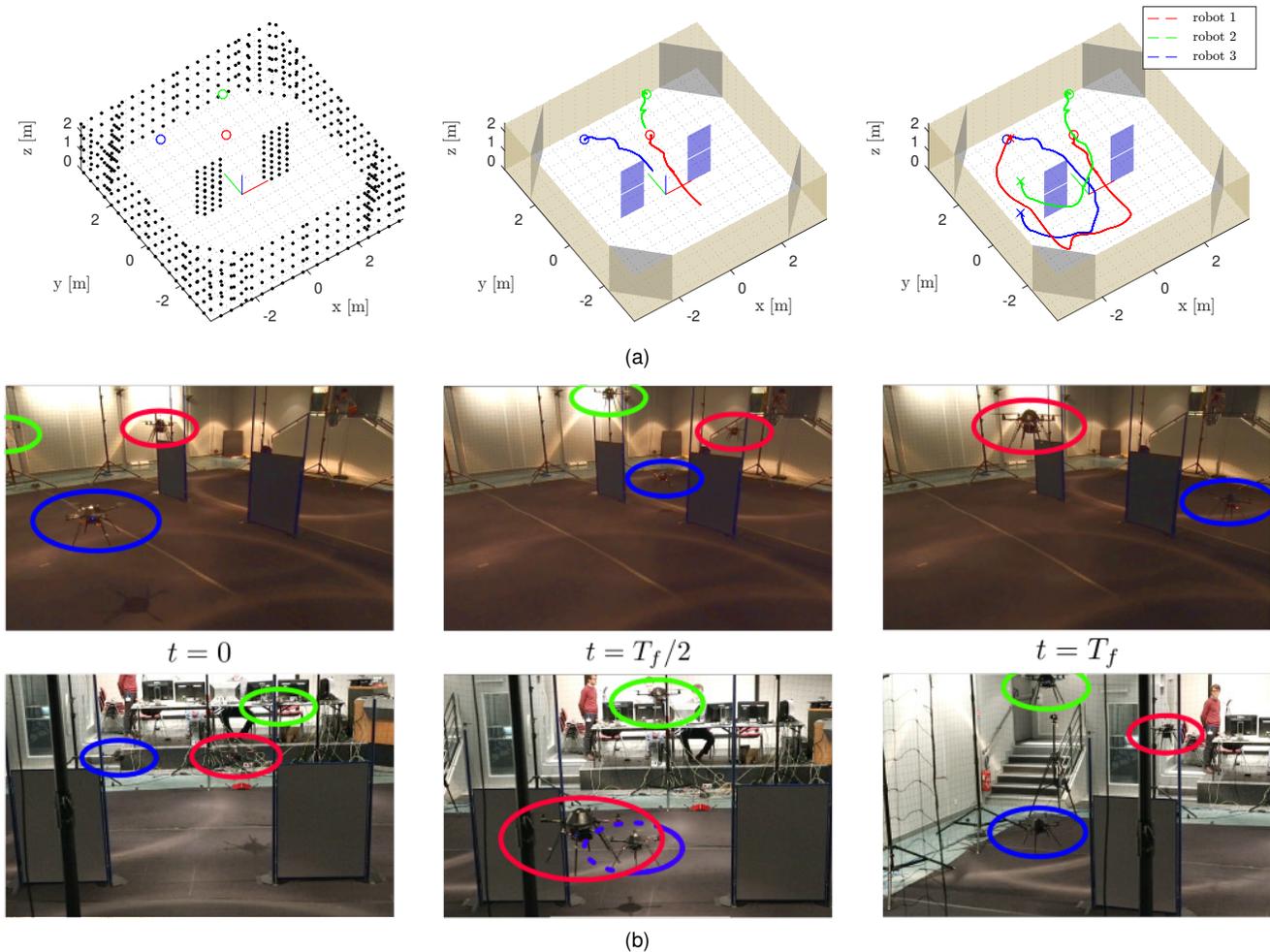


Fig. 17. Experiment in a real environment. The red circle indicates the leader robot (robot 1), green and blue circles indicate robot 2 and 3, respectively. (a) from left to right: robot initial positions at time  $t = 0$  and grid of obstacles; robot trajectories and positions at time  $t = T_{\text{final}}/2$ ; robot trajectories and positions at time  $t = T_{\text{final}}$ . (b) video stills of the experiment at time  $t = 0$ ,  $t = T_{\text{final}}/2$ , and  $t = T_{\text{final}} = 129$  s (left to right) and from two cameras (top and bottom). Video available at [https://youtu.be/\\_eMsbMImqaU?t=405](https://youtu.be/_eMsbMImqaU?t=405).

that users did not pay enough attention to the proposed visual representation of the connectivity force. To better understand this result, we asked the participants why they did not follow the suggestions provided by the visual feedback as precisely as when provided via haptic feedback. Ten subjects out of twelve mentioned that the visual representation was too easy to disregard and ignore. Finally, it is also important to notice that subjects were provided with very rich visual information during the experiment (e.g., five views of the scene). In cases where less information is available, feedback on the status of the team may be even more valuable/useful.

Among the considered metrics, the maximum total force holds a very important status, since it provides information about the maximum acceleration/thrust requested to the robots, a failure in providing such acceleration may result in a loss of connectivity for the team. While this undesired behavior cannot happen in our simulated scenario, where the virtual drones can apply infinite thrusts, it can obviously happen when implementing the system in a real environment. The world fastest drone can accelerate up to  $35 \text{ m/s}^2$  [48], while the more common DJI Phantom 3 Pro drone can accelerate up to  $31 \text{ m/s}^2$  [49]. From Fig. 15d, we can see that a team of DJI Phantom 3 Pro drones would have not been able to guarantee

the connectivity of the team in conditions JN and JV, as the peak thrust requested in those condition was higher than the one the robots can actuate. On the other hand, conditions providing haptic feedback demanded lower acceleration values. Of course, there are many commercially-available drones that cannot even accelerate more than  $10 \text{ m/s}^2$ . Therefore, it is very important to tune the gains of the haptic feedback to match the maximum applicable thrust. Higher gains will lead to higher connectivity forces provided to the user, which should in turn lead to a lower thrust requested to the drones. In the future, it could be interesting to study how to express the feedback control actions of Sec. VI-A3 in terms of the maximum thrust the drones can provide. Finally, we also measured the user's experience, evaluating the perceived effectiveness of each condition as well as the tiredness at the end of the task. Results are shown in Fig. 16. Users found the two conditions providing haptic feedback to be the most effective, but condition LC was found very tiring. This result is largely expected as the wearable device weighs 220 g. Reducing the weights of our wearable device is another priority for future work [50].

Looking at Figs. 15 and 16a, it seems quite clear that conditions OK and LC often perform very similarly. To prove this point, we ran an equivalence test for all the metrics,

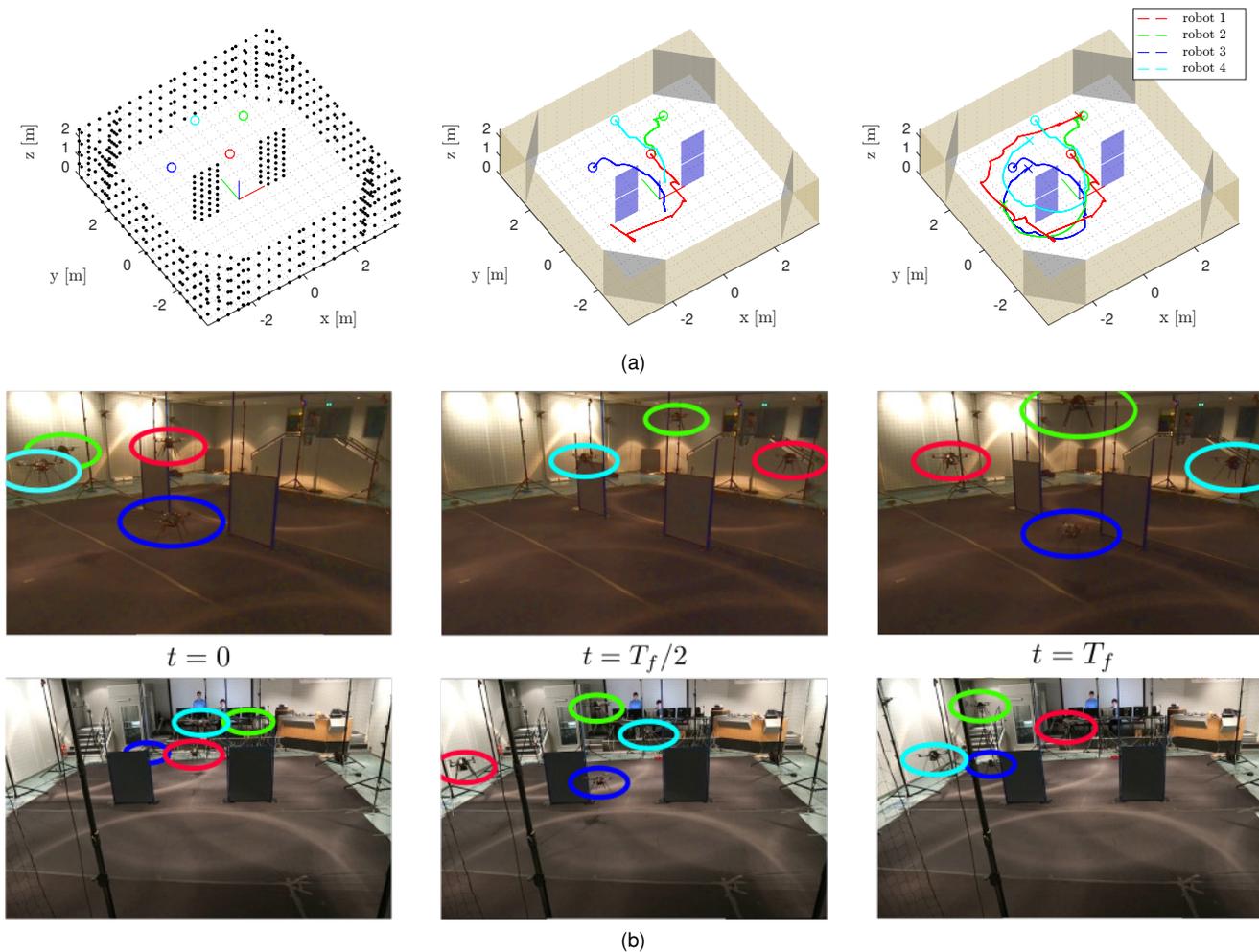


Fig. 18. Evaluation of our approach in a real-setting scenario, employing four quadrotors. Each robot is identified with a color: red is the leader robot (robot 1), green, blue, and cyan are robot 2, robot 3, and robot 4, respectively. (a) from left to right: robot initial positions at time  $t = 0$  and grid of obstacles; robot trajectories at time  $t = T_{\text{final}}/2$ ; robot trajectories and positions at time  $t = T_{\text{final}}$ . (b) shots of the experiment at time  $t = 0$ ,  $t = T_{\text{final}}/2$ , and  $t = T_{\text{final}} = 147$  s, from left to right, respectively, from two cameras.

showing that the performance of OK and LC are indeed statistically equivalent for most metrics. This result is very promising because the setup required by LC (Leap Motion + wearable device) is significantly less bulky and expensive than that required by OK (grounded interface). This result may open new interesting scenarios in the field of ubiquitous robotics and human-robot collaboration. The human operator may not need to stay in a fixed position, but instead can move and even share the *same* environment of the team of robots. A fully wearable and portable setup would indeed enable the human operator to become an integral part of the team, becoming literally its leader. This fascinating idea will be developed in future work.

After the virtual experiment of Sec. VI-A, we also carried out an experiment in a real environment, described in Sec. VI-B. We enrolled two expert operators to control a team composed either of three or four drones. For this experiment, we only considered condition LC. Both operators greatly appreciated the control interface and feedback modality, and they were both able to complete the task with minimal supervision and no practice trials. This result confirms the effectiveness of the proposed connectivity-maintenance algorithm as well as the viability of using wearable cutaneous feedback to provide relevant information about the robotic team.

In the future, we will extend the algorithm to consider drones as proper 3D volumes, as well as consider a more sophisticated dynamics model. We also want to carry out additional experiments, moving steadily toward more realistic environments. This includes carrying out experiments outdoor relying to onboard sensing only and addressing robot-to-robot and human-to-team communication issues. Robot-to-robot communication can be implemented using omnidirectional antennas, while human-to-team communication can be implemented equipping the user with a portable computing unit. As long-distance communication might suffer from delay, jitter, and information loss, it is important to consider stability-maintenance control techniques, such as those based on energy-bounding algorithms. We also want to enroll more subjects, and we plan to use the results observed in this paper to calculate the right sample size. Finally, we will also consider new ways and tools for providing wearable feedback information to the users, e.g., [42], [51].

#### APPENDIX A

Consider a group of  $N$  robots able to measure their relative positions and communicate according to the sensing/communication graph  $\mathcal{G}$ . Recalling Sec. IV-B, we let  $\hat{\mathbf{d}}_{c,i} \in \mathbb{R}^3$  denote the  $i$ -th robot estimation of the relative

position  $\mathbf{x}_i - \mathbf{1}_{N_3}^T \mathbf{x}/N$  between robot  $i$  and the formation centroid.

**Proposition 1.** *If the estimated  $\hat{\mathbf{d}}_{c,i}$  is updated as*

$$\dot{\hat{\mathbf{d}}}_{c,i} = k_c \sum_{j \in \mathcal{N}_i} (\hat{\mathbf{d}}_{c,j} - \hat{\mathbf{d}}_{c,i}) - (\mathbf{x}_j - \mathbf{x}_i), \quad k_c > 0, \quad (29)$$

and  $\dot{\mathbf{x}} = \mathbf{0}$  (stationary robots) and  $\hat{\mathbf{d}}_{c,i}(t_0) = \mathbf{0} \forall i$ , then  $\hat{\mathbf{d}}_{c,i}(t) \rightarrow \mathbf{x}_i - \mathbf{1}_{N_3}^T \mathbf{x}/N$ .

*Proof.* Let  $\hat{\mathbf{d}}_c \in \mathbb{R}^{3N}$  and  $\mathbf{x} \in \mathbb{R}^{3N}$  be the stacks of vectors  $\hat{\mathbf{d}}_{c,i}$  and  $\mathbf{x}_i$ . We first note that that update law (29) can be rearranged as

$$\dot{\hat{\mathbf{d}}}_{c,i} = k_c \sum_{j \in \mathcal{N}_i} (\hat{\mathbf{d}}_{c,j} - \mathbf{x}_j) - (\hat{\mathbf{d}}_{c,i} - \mathbf{x}_i), \quad k_c > 0, \quad (30)$$

which is a consensus on the quantity  $\hat{\mathbf{d}}_c - \mathbf{x}$ , i.e., in compact form

$$\dot{\hat{\mathbf{d}}}_c = -k_c \mathbf{L}_3 (\hat{\mathbf{d}}_c - \mathbf{x}) \quad (31)$$

where  $\mathbf{L}_3 = \mathbf{L} \otimes \mathbf{I}_3$  and  $\mathbf{L}$  is the Laplacian matrix associated to graph  $\mathcal{G}$ . Let us now define the estimation error for the  $i$ -th robot as

$$\mathbf{e}_i = \hat{\mathbf{d}}_{c,i} - (\mathbf{x}_i - \mathbf{1}_{N_3}^T \mathbf{x}/N).$$

The stack  $\mathbf{e} \in \mathbb{R}^{3N}$  of the estimation errors for all robots can then be expressed as

$$\mathbf{e} = \hat{\mathbf{d}}_c - (\mathbf{x} - \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T \mathbf{x}/N) = \hat{\mathbf{d}}_c - (\mathbf{I}_{3N} - \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T / N) \mathbf{x}. \quad (32)$$

The goal is to show that the error  $\mathbf{e}(t) \rightarrow \mathbf{0}$  under (31) and the stated assumptions  $\dot{\mathbf{x}} = \mathbf{0}$  and  $\hat{\mathbf{d}}_c(t_0) = \mathbf{0}$ .

The dynamics of  $\mathbf{e}$  when implementing (31) and exploiting  $\dot{\mathbf{x}} = \mathbf{0}$  is given by

$$\dot{\mathbf{e}} = \dot{\hat{\mathbf{d}}}_c - (\mathbf{I}_{3N} - \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T / N) \dot{\mathbf{x}} = -k_c \mathbf{L}_3 (\hat{\mathbf{d}}_c - \mathbf{x}) \quad (33)$$

Since, as well-known,  $\mathbf{L} \mathbf{1}_N = \mathbf{0}$ , from the properties of the Kronecker product it follows that  $\mathbf{L}_3 \mathbf{1}_{N_3} = \mathbf{0}$ . Therefore, the error dynamics (33) can be also be written as

$$\dot{\mathbf{e}} = -k_c \mathbf{L}_3 (\hat{\mathbf{d}}_c - \mathbf{x} + \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T / N \mathbf{x}) = -k_c \mathbf{L}_3 \mathbf{e}, \quad (34)$$

showing that (31) imposes as well a consensus on the estimation error  $\mathbf{e}$ . The error  $\mathbf{e}(t)$  will then converge towards the average of its initial condition, i.e.,

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T \mathbf{e}(t_0) / N.$$

Using (32), this can be expanded as

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{1}_{N_3} \frac{\mathbf{1}_{N_3}^T}{N} \hat{\mathbf{d}}_c(t_0) - \mathbf{1}_{N_3} \frac{\mathbf{1}_{N_3}^T}{N} \left( \mathbf{I}_{3N} - \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T / N \right) \mathbf{x}(t_0). \quad (35)$$

Since  $\left( \mathbf{I}_{3N} - \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T / N \right)$  is a projector onto the space orthogonal to  $\mathbf{1}_{N_3}$ , (35) reduces to

$$\lim_{t \rightarrow \infty} \mathbf{e}(t) = \mathbf{1}_{N_3} \frac{\mathbf{1}_{N_3}^T}{N} \hat{\mathbf{d}}_c(t_0). \quad (36)$$

Exploiting the second assumption  $\hat{\mathbf{d}}_c(t_0) = \mathbf{0}$ , the proof then follows.  $\square$

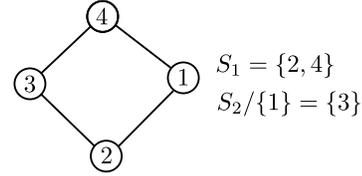


Fig. 19. Illustrative group of robots. The airflow weight  $\phi_{ij}$  is a function of the relative positions between robots  $i$  and  $j$  and their neighbors.

We conclude with some remarks about the two assumptions  $\dot{\mathbf{x}} = \mathbf{0}$  and  $\hat{\mathbf{d}}_c(t_0) = \mathbf{0}$ . The condition  $\hat{\mathbf{d}}_c(t_0) = \mathbf{0}$  can be easily met by having each robot starting with its own estimated state  $\hat{\mathbf{d}}_{c,i}(t_0) = \mathbf{0}$ . As for the assumption  $\dot{\mathbf{x}} = \mathbf{0}$ , if the robots are moving slow enough compared to the estimation dynamics (as it is typically the case), then the estimation error will practically converge also for a  $\dot{\mathbf{x}} \neq \mathbf{0}$ . In case, however, the robot motion is not negligible, the estimation dynamics (31) can be extended as follows

$$\dot{\hat{\mathbf{d}}}_c = -k_c \mathbf{L}_3 (\hat{\mathbf{d}}_c - \mathbf{x}) + (\mathbf{I}_{3N} - \mathbf{1}_{N_3} \mathbf{1}_{N_3}^T / N) \dot{\mathbf{x}} \quad (37)$$

for fully compensating for this effect. When expanded for the  $i$ -th robot, the update law (37) becomes

$$\dot{\hat{\mathbf{d}}}_{c,i} = k_c \sum_{j \in \mathcal{N}_i} \left( (\hat{\mathbf{d}}_{c,j} - \hat{\mathbf{d}}_{c,i}) - (\mathbf{x}_j - \mathbf{x}_i) \right) + \dot{\mathbf{x}}_i - \mathbf{1}_{N_3}^T \dot{\mathbf{x}} / N \quad (38)$$

showing that each robot would need an estimation of the group average velocity  $\mathbf{1}_{N_3}^T \dot{\mathbf{x}} / N$  that can be retrieved by a consensus-like law run on  $\dot{\mathbf{x}}$  (e.g., the popular PI-ACE algorithm [52]).

## APPENDIX B

**Proposition 2.** *Given a group of  $N$  robots modeled and controlled as in Sec. III, the airflow weight  $\phi_{ij}$  is symmetric, that is,*

$$\phi_{ij} = \phi_{ji}, \quad \forall (i, j) \mid i \in V, j \in V, i \neq j$$

*Proof.* In order to show that  $\phi_{ij} = \phi_{ji}$  we will use w.l.o.g. the example shown in Fig. 19 for a simplified planar case. In the following, we will focus on the pair of robots (1, 2), but the reasoning can be applied to any pair of robots.

Recalling the definition of  $\phi_{ij}$  in (15), from Fig. 19 we have

$$\begin{cases} \phi_{12} = \prod_{k \in S_1} \phi_{1k}^* \left( \prod_{k \in S_2/\{1\}} \phi_{2k}^* \right) = \phi_{12}^* \phi_{14}^* \phi_{23}^* \\ \phi_{21} = \prod_{k \in S_2} \phi_{2k}^* \left( \prod_{k \in S_1/\{2\}} \phi_{1k}^* \right) = \phi_{21}^* \phi_{23}^* \phi_{14}^* \end{cases} \quad (39)$$

By looking at (39), showing that  $\phi_{12}^* = \phi_{21}^*$  would prove Proposition 2. Let us now recall the definition of  $\phi_{ij}^*(c_{ij}, d_{ij})$  from (14), and let us apply it to  $\phi_{12}^*(c_{12}, d_{12})$  and  $\phi_{21}^*(c_{21}, d_{21})$

$$\begin{cases} \phi_{12}^*(c_{12}, d_{12}) = \bar{\phi}_{12}(c_{12}) (1 - f(d_{12})) + f(d_{12}) \\ \phi_{21}^*(c_{21}, d_{21}) = \bar{\phi}_{21}(c_{21}) (1 - f(d_{21})) + f(d_{21}) \end{cases} \quad (40)$$

Since  $d_{12} = d_{21}$  by the definition of distance, we are only left to show that  $\bar{\phi}_{12}(c_{12}) = \bar{\phi}_{21}(c_{21})$ .

**Proposition 3.** Given Proposition 2, the partial airflow weight as  $\bar{\phi}_{ij}(c_{ij})$  is symmetric, that is,

$$\bar{\phi}_{ij}(c_{ij}) = \bar{\phi}_{ji}(c_{ji}), \quad \forall (i, j) \mid i \in \mathbb{V}, j \in \mathbb{V}, i \neq j$$

*Proof.* Let us now use the definition of  $\bar{\phi}_{ij}(c_{ij})$  from (12) in (40)

$$\begin{cases} \bar{\phi}_{12}(c_{12}) = \begin{cases} w_0^1(c_{12}, c_{\min}, c_{\max}) & \text{if } c_{12} \geq 0 \\ w_1^0(c_{12}, -c_{\max}, -c_{\min}) & \text{if } c_{12} < 0 \end{cases} \\ \bar{\phi}_{21}(c_{21}) = \begin{cases} w_0^1(c_{21}, c_{\min}, c_{\max}) & \text{if } c_{21} \geq 0 \\ w_1^0(c_{21}, -c_{\max}, -c_{\min}) & \text{if } c_{21} < 0 \end{cases} \end{cases} \quad (41)$$

From (11), one has  $c_{ij} = -c_{ji}$ , which therefore leads to only three feasible combinations for  $\bar{\phi}_{12}(c_{12})$  and  $\bar{\phi}_{21}(c_{21})$ , i.e.

$$\begin{cases} \begin{cases} \bar{\phi}_{12}(c_{12}) = 1 \\ \bar{\phi}_{21}(c_{21}) = 1 \end{cases} & \text{if } \begin{cases} c_{12} = 0, \\ c_{21} = 0 \end{cases} \\ \begin{cases} \bar{\phi}_{12}(c_{12}) = w_0^1(c_{12}, c_{\min}, c_{\max}) \\ \bar{\phi}_{21}(c_{21}) = w_1^0(c_{21}, -c_{\max}, -c_{\min}) \end{cases} & \text{if } \begin{cases} c_{12} > 0, \\ c_{21} < 0 \end{cases} \\ \begin{cases} \bar{\phi}_{12}(c_{12}) = w_1^0(c_{12}, -c_{\max}, -c_{\min}) \\ \bar{\phi}_{21}(c_{21}) = w_0^1(c_{21}, c_{\min}, c_{\max}) \end{cases} & \text{if } \begin{cases} c_{12} < 0, \\ c_{21} > 0 \end{cases} \end{cases} \quad (42)$$

The first row of (42) corresponds to the case when two quadrotors have the same altitude, and it produces a final weight  $\phi_{ij} = \phi_{ji} = 1$ , which solves Propositions 2 and 3 for the case  $c_{ij} = c_{ji}$ . We now consider the second combination in (42); for the third row, the proof procedure follows the same reasoning. In case of  $c_{12} > 0$  and  $c_{21} < 0$ , (42) can be further divided into three cases, depending on the magnitude of  $c_{12}$  w.r.t. thresholds  $c_{\min}$  and  $c_{\max}$ . When quadrotors (1, 2) altitudes are relative similar, that is, when  $0 < c_{12} \leq c_{\min}$  and  $-c_{\min} \leq c_{21} < 0$ , we have

$$\bar{\phi}_{12}(c_{12}) = \bar{\phi}_{21}(c_{21}) = 1.$$

For larger values of  $c_{12}$  ( $c_{\max} < c_{12}$ , and thus  $c_{21} < -c_{\max}$ ), we have

$$\bar{\phi}_{12}(c_{12}) = \bar{\phi}_{21}(c_{21}) = 0,$$

which results in  $\bar{\phi}_{12}(c_{12}) = \bar{\phi}_{21}(c_{21})$  when two quadrotors are one above each other (or close to this position). The third possibility, that is, when the two robots relative zenith angles are between the set thresholds  $c_{\min}$  and  $c_{\max}$ , yields (recalling (3) and (4))

$$\bar{\phi}_{12}(c_{12}) = \frac{1}{2} \left( 1 + \cos \left( \frac{\pi c_{12}}{c_{\max} - c_{\min}} - \frac{\pi c_{\min}}{c_{\max} - c_{\min}} \right) \right) \quad (43)$$

$$\bar{\phi}_{21}(c_{21}) = \frac{1}{2} \left( 1 - \cos \left( \frac{\pi c_{21}}{c_{\max} - c_{\min}} + \frac{\pi c_{\max}}{c_{\max} - c_{\min}} \right) \right). \quad (44)$$

As the last step, let us compare (43) and (44). By setting  $c_{\text{diff}} = c_{\max} - c_{\min}$ , which yields  $\pi = \pi(c_{\max} - c_{\min})/c_{\text{diff}}$ , and by remembering  $c_{ij} = -c_{ji}$  from (11), (44) can be rewritten

$$\begin{aligned} & \frac{1}{2} \left( 1 - \cos \left( \frac{\pi c_{21}}{c_{\text{diff}}} + \frac{\pi c_{\max}}{c_{\text{diff}}} \right) \right) = \\ & \frac{1}{2} \left( 1 + \cos \left( \pi - \frac{\pi c_{21}}{c_{\text{diff}}} - \frac{\pi c_{\max}}{c_{\text{diff}}} \right) \right) = \\ & \frac{1}{2} \left( 1 + \cos \left( \pi \left( \frac{c_{\max} - c_{\min} - c_{21} - c_{\max}}{c_{\text{diff}}} \right) \right) \right) = \\ & \frac{1}{2} \left( 1 + \cos \left( \frac{\pi c_{12} - \pi c_{\min}}{c_{\text{diff}}} \right) \right). \end{aligned} \quad (45)$$

Since (45) is equal to (43),  $\bar{\phi}_{12}(c_{12}) = \bar{\phi}_{21}(c_{21})$  also for the case when the two robots relative zenith angles are between the set thresholds  $c_{\min}$  and  $c_{\max}$ , thus proving Proposition 3 and Proposition 2.  $\square$

## REFERENCES

- [1] Y. Liu and G. Nejat, "Robotic urban search and rescue: A survey from the control perspective," *Journal of Intelligent & Robotic Systems*, vol. 72, no. 2, pp. 147–165, 2013.
- [2] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Trans. Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 3, pp. 367–385, 2003.
- [3] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic, R. Siegart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty *et al.*, "The SHERPA project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments," in *Proc. IEEE Int. Symp. on Safety, Security, and Rescue Robotics (SSRR)*, 2012, pp. 1–4.
- [4] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.
- [5] C. Secchi, A. Franchi, H. H. Bülthoff, and P. Robuffo Giordano, "Bilateral Control of the Degree of Connectivity in Multiple Mobile-robot Teleoperation," in *IEEE ICRA*, 2013, pp. 3645–3652.
- [6] M. Silvagni, A. Tonoli, E. Zenerino, and M. Chiaberge, "Multipurpose uav for search and rescue operations in mountain avalanche events," *Geomatics, Natural Hazards and Risk*, vol. 8, no. 1, pp. 18–33, 2017.
- [7] B. Custers, *Future of Drone Use*, 2016.
- [8] W. S. Ng and E. Sharlin, "Collocated interaction with flying robots," in *IEEE Int. Symp. on Robots and Human Interactive Communications*, 2011, pp. 143–149.
- [9] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A Kinect-based natural interface for quadrotor control," *Entertainment Computing*, vol. 4, no. 3, pp. 179–186, 2013.
- [10] K. Pfeil, S. L. Koh, and J. LaViola, "Exploring 3D gesture metaphors for interaction with unmanned aerial vehicles," in *Proc. Intl. Conf. Intelligent User Interfaces*, 2013, pp. 257–266.
- [11] J. P. Hansen, A. Alapetite, I. S. MacKenzie, and E. Møllenbach, "The use of gaze to control drones," in *Proc. ACM Symp. on Eye Tracking Research and Applications*, 2014, pp. 27–34.
- [12] Y. Yu, D. He, W. Hua, S. Li, Y. Qi, Y. Wang, and G. Pan, "FlyingBuddy2: a brain-controlled assistant for the handicapped," in *Int. Conf. on Ubiquitous Computing*, 2012, pp. 669–670.
- [13] C. Rognon, S. Mintchev, F. Dell'Agnola, A. Cherpillod, D. Atienza, and D. Floreano, "Flyjacket: An upper body soft exoskeleton for immersive drone control," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2362–2369, 2018.
- [14] Somniacs SA, "Birdly," <https://web.archive.org/web/20190530103144/https://www.somniacs.co/>, [Online; accessed May-2019].
- [15] HYPERSUIT, "Hypersuit," <https://web.archive.org/web/20190530103302/https://www.hypersuit.fr/>, [Online; accessed May-2019].
- [16] L. A. Sandru, M. F. Crainic, D. Savu, C. Moldovan, V. Dolga, and S. Preitl, "Automatic Control of a Quadcopter, AR. Drone, Using a Smart Glove," in *Proc. Int. Conf. on Control, Mechatronics and Automation*, 2016, pp. 92–98.
- [17] A. Sarkar, K. A. Patel, R. G. Ram, and G. K. Kapoor, "Gesture control of drone using a motion controller," in *Proc. Int. Conf. on Industrial Informatics and Computer Systems (IIICS)*, 2016, pp. 1–5.

- [18] R. A. S. Fernández, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina, and P. Campoy, "Natural user interfaces for human-drone multi-modal interaction," in *Proc. Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 1013–1022.
- [19] C. Rognon, A. R. Wu, S. Mintchev, A. Ijspeert, and D. Floreano, "Haptic Guidance with a Soft Exoskeleton Reduces Error in Drone Teleoperation," in *Proc. Int. Conf. on Human Haptic Sensing and Touch Enabled Computer Applications*, 2018, pp. 404–415.
- [20] J. Cacace, A. Finzi, and V. Lippiello, "Multimodal interaction with multiple co-located drones in search and rescue missions," *arXiv preprint arXiv:1605.07316*, 2016.
- [21] E. Tsykunov, R. Agishev, R. Ibrahimov, L. Labazanova, A. Tleugazy, and D. Tsetserukou, "Swarmtouch: Guiding a swarm of micro-quadrotors with impedance control using a wearable tactile interface," *IEEE Trans. Haptics*, vol. 12, no. 3, pp. 363–374, 2019.
- [22] D. Lee, A. Franchi, H. I. Son, C. Ha, H. H. Bühlhoff, and P. Robuffo Giordano, "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.
- [23] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bühlhoff, "Experiments of Passivity-Based Bilateral Aerial Teleoperation of a Group of UAVs with Decentralized Velocity Synchronization," in *IEEE IROS*, 2011, pp. 163–170.
- [24] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. Robuffo Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor UAVs," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 57–68, 2012.
- [25] J.-A. Claret, I. Zaplana, and L. Basañez, "Teleoperating a mobile manipulator and a free-flying camera from a single haptic device," in *Proc. IEEE Int. Symp. on Safety, Security, and Rescue Robotics*, 2016, pp. 291–296.
- [26] N. Staub, M. Mohammadi, D. Bicego, Q. Delamare, H. Yang, D. Prattichizzo, P. Robuffo Giordano, D. Lee, and A. Franchi, "The TeleMAGMaS: An Aerial-Ground Comanipulator System," *IEEE Robotics & Automation Magazine*, 2017.
- [27] M. Aggravi, F. Pausé, P. Robuffo Giordano, and C. Pacchierotti, "Design and Evaluation of a Wearable Haptic Device for Skin Stretch, Pressure, and Vibrotactile Stimuli," in *IEEE Robotics and Automation Letters*, vol. 3, no. 3, 2018, pp. 2166–2173.
- [28] A. Franchi, C. Secchi, H. I. Son, H. H. Bühlhoff, and P. Robuffo Giordano, "Bilateral Teleoperation of Groups of Mobile Robots with Time-Varying Topology," *IEEE Trans. Robotics*, vol. 28, no. 5, pp. 1019–1033, 2012.
- [29] D. Zelazo, A. Franchi, H. H. Bühlhoff, and P. Robuffo Giordano, "Decentralized Rigidity Maintenance Control with Range Measurements for Multi-Robot Systems," *International Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2015.
- [30] F. Schiano and P. Robuffo Giordano, "Bearing rigidity maintenance for formations of quadrotor uavs," in *IEEE Intl. Conf. Robotics and Automation (ICRA)*, 2017.
- [31] A. Franchi and P. Robuffo Giordano, "Online leader selection for improved collective tracking and formation maintenance," *IEEE Trans. Control of Network Systems*, vol. 5, no. 1, pp. 3–13, 2018.
- [32] A. Franchi, P. Robuffo Giordano, and G. Michieletto, "Online Leader Selection for Collective Tracking and Formation: the Second Order Case," *IEEE Trans. Control of Network Systems*, 2018.
- [33] F. Schiano, A. Franchi, D. Zelazo, and P. Robuffo Giordano, "A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 5099–5106.
- [34] E. Soria, F. Schiano, and D. Floreano, "The influence of limited visual sensing on the reynolds flocking algorithm," in *Proc. IEEE Intl. Conf. Robotic Computing (IRC)*, 2019, pp. 138–145.
- [35] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.
- [36] E. Canepa, "An overview about the study of downwash effects on dispersion of airborne pollutants," *Environmental Modelling & Software*, vol. 19, no. 12, pp. 1077–1087, 2004.
- [37] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," in *IEEE ICRA*, 2017, pp. 3293–3298.
- [38] D. W. Yeo, N. Sydney, and D. A. Paley, "Onboard flow sensing for multi-rotor pitch control in wind," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 5, pp. 1196–1201, 2018.
- [39] D. Yeo, E. Shrestha, D. Paley, and E. Atkins, "An Empirical Model of Rotorcraft UAV Downwash Model for Disturbance Localization and Avoidance," in *Proc. AIAA Atmospheric Flight Mechanics Conference*, 2015.
- [40] D. W. Yeo, N. Sydney, D. Paley, and D. Sofge, "Onboard Flow Sensing For Downwash Detection and Avoidance On Small Quadrotor Helicopters," in *Proc. AIAA Guidance Navigation and Control Conference*, 2015, pp. 2015–1769.
- [41] J. Bimbo, C. Pacchierotti, M. Aggravi, N. Tsagarakis, and D. Prattichizzo, "Teleoperation in cluttered environments using wearable haptic feedback," in *IEEE IROS*, 2017.
- [42] F. Chinello, C. Pacchierotti, J. Bimbo, N. G. Tsagarakis, and D. Prattichizzo, "Design and evaluation of a wearable skin stretch device for haptic guidance," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 524–531, 2017.
- [43] F. Abi-Farraj, C. Pacchierotti, O. Arenz, G. Neumann, and P. R. Giordano, "A haptic shared-control architecture for guided multi-target robotic grasping," *IEEE Trans. Haptics*, 2019.
- [44] M. Aggravi, G. Salvietti, and D. Prattichizzo, "Haptic wrist guidance using vibrations for Human-Robot teams," in *IEEE Int. Symp. on Robots and Human Interactive Communications*, 2016, pp. 113–118.
- [45] G. B. Limentani, M. C. Ringo, F. Ye, M. L. Bergquist, and E. O. McSorley, "Beyond the t-test: statistical equivalence testing," *Analytical chemistry*, vol. 77, no. 11, pp. 221–226, 2005.
- [46] C. Pacchierotti, D. Prattichizzo, and K. J. Kuchenbecker, "Displaying sensed tactile cues with a fingertip haptic device," *IEEE Trans. Haptics*, vol. 8, no. 4, pp. 384–396, 2015.
- [47] C. Pacchierotti, *Cutaneous haptic feedback in robotic teleoperation*, 2015.
- [48] D. Brunner, "Extreme high power drone with highest climb rate," [https://web.archive.org/web/20180814195214/https://www.research-drone.com/en/extreme\\_climb\\_rate.html](https://web.archive.org/web/20180814195214/https://www.research-drone.com/en/extreme_climb_rate.html), 2018, [Online; accessed May-2019].
- [49] T. H. C. Team, "Drone thrust testing," <https://web.archive.org/web/20180925134358/https://www.halfchrome.com/drone-thrust-testing>, 2018, [Online; accessed May-2019].
- [50] C. Pacchierotti, S. Sinclair, M. Solazzi, A. Frisoli, V. Hayward, and D. Prattichizzo, "Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives," *IEEE Trans. Haptics*, 2017.
- [51] F. Chinello, C. Pacchierotti, M. Malvezzi, and D. Prattichizzo, "A three revolute-revolute-spherical wearable fingertip cutaneous device for stiffness rendering," *IEEE Trans. Haptics*, vol. 11, no. 1, pp. 39–50, 2017.
- [52] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," in *IEEE CDC*, 2006, pp. 338–343.



**Marco Aggravi** (S'11, M'16) received his M.Sc. degree in Computer Science Engineering in 2011 and his Ph.D. degree in Information Engineering and Science in 2016, both from the University of Siena, Italy. In 2011, he was a Visiting Researcher at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), Germany. In 2016, he was a Visiting Researcher at the Georgia Institute of Technology, USA. Since 2017, he has been a Post-doctoral Researcher with the Rainbow group at Irisa and Inria in Rennes, France.



**Claudio Pacchierotti** (S'12, M'15, SM'20) is a researcher of the CNRS at IRISA and Inria RBA, Rennes, France. He earned his Ph.D. degree at the Univ. Siena in 2014, and then he was a PostDoc at the Italian Institute of Technology, Genova. He visited the Penn Haptics Group at the University of Pennsylvania in 2014, the DIMEG of the University of Padua in 2013, and the MIRA of the University of Twente in 2014. Pacchierotti received the 2014 EuroHaptics Best PhD Thesis Award, and the 2015 IEEE Trans. Haptics Meritorious Service Award. He is Chair of the IEEE Technical Committee on Haptics and Secretary of the Eurohaptics Society.



**Paolo Robuffo Giordano** (M'08, SM'16) received his M.Sc. degree in Computer Science Engineering in 2001, and his Ph.D. degree in Systems Engineering in 2008, both from the University of Rome "La Sapienza". In 2007 and 2008 he spent one year as a PostDoc at the Institute of Robotics and Mechatronics of the German Aerospace Center (DLR), and from 2008 to 2012 he was Senior Research Scientist at the Max Planck Institute for Biological Cybernetics in Tübingen, Germany. He is currently a senior CNRS researcher head of the Rainbow group at Irisa and

Inria in Rennes, France.