

Decentralized Connectivity Maintenance for Quadrotor UAVs with Field of View Constraints

Maxime Bernard, Claudio Pacchierotti, Paolo Robuffo Giordano

Abstract—We present a decentralized connectivity-maintenance algorithm for controlling a group of quadrotor UAVs with limited field of view (FOV) and not sharing a common reference frame for collectively expressing measurements and commands. This is in contrast to the vast majority of previous works on this topic which, instead, make the (simplifying) assumptions of omnidirectional sensing and presence of a common shared frame. For achieving this goal, we design a gradient-based connectivity-maintenance controller able to take into account the presence of a limited FOV. We also propose a novel (to our knowledge) decentralized estimator of the relative orientation among neighboring robots, which is a necessary quantity for correctly implementing the connectivity-maintenance action. We validate the framework in realistic simulations that show the effectiveness of our approach.

I. INTRODUCTION

Teams of coordinated robots are increasingly being employed in high-impact scenarios, including mapping, surveillance, disaster response, exploration, border control, and patrolling [1]–[3]. Their use has rapidly increased especially thanks to their higher expendability and flexibility with respect to standard human teams, as well as their ability to adapt to different situations and exploit onboard sensors for analyzing the surrounding environment. The action of such coordinated teams of robots is expected to result in higher redundancy, broader covered area, and better complementarity of features with respect to approaches employing single robots [4]. While grounded Urban Search-And-Rescue (USAR) robotic teams are already widely used in the field, aerial solutions are only recently being deployed [5].

As these applications are generally highly unstructured, it is often considered beneficial to enable a human operator to control the robotic team [3], [6]. For example, most currently-available USAR robotic systems are fully teleoperated [7], while autonomous solutions are still rare. On the other hand, autonomous surveillance robots are more common, but a remote human operator can usually still access them if needed. Of course, having a human user steer a multi-robot system raises several challenges, as she or he needs to intuitively and effectively control multiple robots at the same time. In this respect, Franchi et al. [8] proposed a “leader-follower” paradigm for the control of multi-robot systems, in which the formation can create and destroy links among agents at runtime according to the considered sensing constraints. The

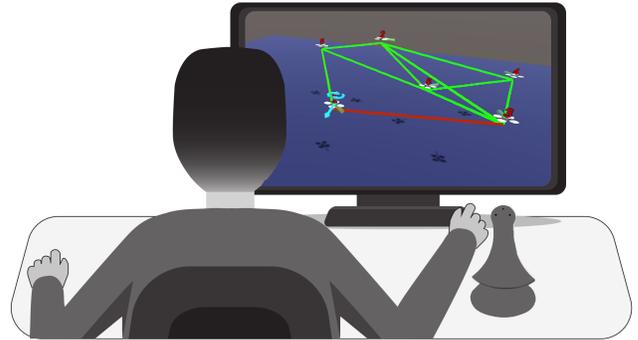


Fig. 1. Experimental setup for simulation with Connectivity Maintenance.

human operator only controls one robot (the leader), while the others reactively follow its movement by enforcing spring-like couplings among connected pairs of neighboring agents. This approach makes the control of the team very flexible and intuitive, as the human can just focus on controlling one leading robot; however, it also leaves very limited control over any other aspect of the team, e.g., the formation shape. Extensions of this algorithm have been proposed to enforce the maintenance of global properties such as connectivity [9] or rigidity [10] of the team, so as to increase the operator’s control over the formation. Other algorithms have explored the possibility of changing the leader at runtime for optimizing the tracking of the operator’s inputs [11], enabling the direct control of some collective parameter of the formation (e.g., the velocity of the group’s barycenter) [12], [13], or using consensus-based formation control [14].

Distributed connectivity-maintenance algorithms such as [9] provide a flexible yet intuitive solution for enabling a human user the control of a multi-robot systems while guaranteeing inter-agent communication throughout the considered mission. Connectivity maintenance has been successfully employed in applications of cooperative surveillance [15], search-and-rescue [16], and guidance [17] using teams of multiple drones controlled by a human operator. The maintenance of the team connectivity is enforced in presence of a series of sensing and interaction constraints, aiming at keeping the agents safe and able to communicate with the rest of the team. However, despite their success and broad applicability, the vast majority of previous works on connectivity maintenance makes several strong *simplifying assumptions* about the robot sensing capabilities and shared knowledge. For instance, they typically assume presence of an omnidirectional sensor for retrieving relative positions of neighboring robots, which are also assumed expressed in a

M. Bernard, C. Pacchierotti and P. Robuffo Giordano are with CNRS at IRISA Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France. {maxime.bernard, claudio.pacchierotti, prg}@irisa.fr

This work was supported by the ANR-20-CHIA-0017 project “MULTI-SHARED”

single common frame shared by the group. These assumptions simplify the control design but are not very realistic, since most onboard sensors have a limited field of view (cameras being a prominent example) and, in addition, every robot collects measurements and expresses command actions in its own *body-frame*, which in general does not need to be aligned with the body frames of the other robots.

The goal of this paper is to then propose a decentralized connectivity-maintenance algorithm able to overcome the two aforementioned issues and explicitly take into account presence of a limited FOV and lack of a common shared frame among the robots. For implementing the connectivity-maintenance action under these new constraints, we also develop a novel decentralized estimator of the relative orientation among neighboring robots, which is needed by the proposed scheme. Finally, we validate the proposed approach in a realistic simulation scenario employing a team of 6 quadrotors UAVs.

The rest of the paper is structured as follows. Sect. II introduces the modeling assumption of this work and then Sect. III details the proposed decentralized connectivity maintenance strategy by taking into account the FOV constraints. Section IV illustrates the decentralized estimator of the relative orientation among the robot body-frames. Section V then validates the approach in a realistic simulation involving 6 quadrotor UAVS, and Sect. VI finally concludes the paper.

II. PRELIMINARIES

A. Robot model

Let $\mathcal{W} : \{\mathbf{O}_{\mathcal{W}}, \mathbf{X}_{\mathcal{W}}, \mathbf{Y}_{\mathcal{W}}, \mathbf{Z}_{\mathcal{W}}\}$ represent a world frame with $\mathbf{Z}_{\mathcal{W}}$ aligned with the vertical (gravity) direction. Similarly to [10], [13], [18] and other related works, we consider a group of N “velocity-controlled” quadrotor UAVs with the following kinematic model

$$\begin{pmatrix} \dot{\mathbf{p}}_i \\ \dot{\psi}_i \end{pmatrix} = \begin{pmatrix} \mathbf{R}_i & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_i \\ w_i \end{pmatrix}, \quad (1)$$

where $\mathbf{p}_i \in \mathbb{R}^3$ is the robot 3D position in \mathcal{W} , $\psi_i \in S^1$ the yaw angle about $\mathbf{Z}_{\mathcal{W}}$ with $\mathbf{R}_i = \mathbf{R}_z(\psi_i) \in SO(3)$ the associated rotation matrix, and $\mathbf{v}_i \in \mathbb{R}^3$ and $w_i \in \mathbb{R}$ represents the *body-frame* linear velocity and yaw rate, which are assumed to be known and controllable.

The robots are assumed to be equipped with two kinds of sensors: a camera-like sensor with limited FOV and an omnidirectional distance sensor. The camera is used for detecting other UAVs in the group. In particular, when robot j is in visibility of robot i , we assume that robot i can obtain from its onboard camera a measurement of

$$\mathbf{p}_{ij} = \mathbf{R}_i^T (\mathbf{p}_j - \mathbf{p}_i), \quad (2)$$

that is, the relative position of j w.r.t. i expressed in the body-frame of i . The distance sensor is instead used for detecting any “obstacle point” $\mathbf{p}_{obs,i}$ within a given sensing range, including other robots as well.

Contrarily to most of the existing literature on connectivity-maintenance algorithms (as a small selection, see [3], [9],

[19], [20]), we note the following facts: we *do not* assume the presence of a common frame shared by all robots (as in all previous work) where all the measurements and commands are expressed. This is evident from model (1) (body-frame input commands) and the measurement (2) (body-frame relative position instead of a world-frame relative position). Furthermore, in this work, the onboard camera is modeled with a limited FOV: among others, this prevents the use of the camera for *obstacle avoidance* whenever an obstacle (or another UAV) is outside the FOV. This is why a second (typically much simpler) omnidirectional distance sensor is also assumed onboard the robots. This distance sensor is assumed able to only detect “non-free space” within a sensing range, but with no ability to discern actual obstacles from other robots (that are then just treated as obstacles for collision avoidance purposes, see Sect. III-D). The limited camera FOV also differentiates this work from the existing literature on connectivity maintenance, which instead simply assumes presence of an omnidirectional sensor able to measure relative positions w.r.t. robots and obstacles in all directions.

B. Sensing and Communication Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with $\mathcal{V} = \{1 \dots N\}$ being the vertex set and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the edge set. We also let $\mathcal{N}_i = \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ be the set of neighbors of robot i . Presence of an edge $e_k = (i, j)$ in \mathcal{E} models the possibility of robots i and j to sense and communicate among themselves. As usual, this is obtained by associating each edge with a state-dependent weight $A_{ij} \geq 0$ such that $A_{ij} > 0$ if the robots can sense/communicate and $A_{ij} = 0$ otherwise. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ associated to \mathcal{G} is then built out of the weights A_{ij} .

In the context of connectivity-maintenance algorithms, in general, we consider an undirected graph with $A_{ij} = A_{ji}$ and, thus, a symmetric adjacency matrix $\mathbf{A} = \mathbf{A}^T$. This is straightforward when assuming omnidirectional (and equal) sensing capabilities for all robots as in, e.g., [3], [9], [19], [20] and similar works. In our context, however, presence of individual (and not aligned) body-frames and of the limited camera FOV would in general result in non-symmetric weights: if $A_{ij} > 0$ (robot i measures robot j), A_{ji} could be 0 (robot j does not measure i because of the limited FOV). In the next Section we discuss how we address this issue for recovering in all cases the symmetry of the weights A_{ij} .

III. CONNECTIVITY MAINTENANCE CONTROL

We briefly recall the connectivity-maintenance algorithm presented in [9] and then discuss its extension to the setting considered in this work. The connectivity maintenance presented in [9] consists of a decentralized gradient descent of a cost function $V^\lambda(\lambda_2) \geq 0$ of the so-called *connectivity eigenvalue* λ_2 , the second smallest eigenvalue of the Laplacian matrix $\mathbf{L} = \text{diag}(\delta_i) - \mathbf{A}$ associated to the graph, where $\delta_i = \sum_{j=1}^N A_{ij}$. The cost function $V^\lambda(\lambda_2)$ is designed to have a vertical asymptote for a minimum connectivity value λ_2^{min} and to smoothly decrease towards 0 for $\lambda_2 > \lambda_2^{min}$ (see Fig. 2). Assume that the edge weights A_{ij} are smooth enough

functions of the state with some (mild) locality properties (in short, a weight A_{ij} may only depend on the state of robot i and of its neighbors).

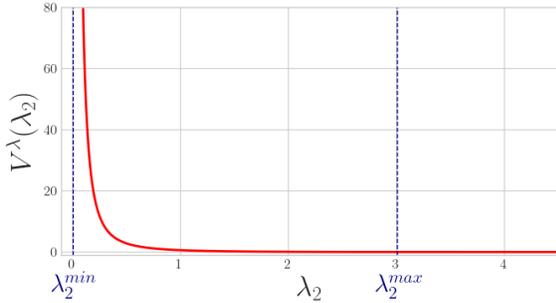


Fig. 2. The shape of $V^\lambda(\lambda_2)$ with $\lambda_2^{\min} = 0.01$ and $\lambda_2^{\max} = 3.01$.

The connectivity control action for a single integrator model $\dot{\mathbf{x}}_i = \mathbf{u}_i$ is the gradient descent

$$\mathbf{u}_i = -\frac{\partial V^\lambda}{\partial \mathbf{x}_i} = -\frac{\partial V^\lambda}{\partial \lambda_2} \sum_{j \in \mathcal{N}_i} \frac{\partial A_{ij}}{\partial \mathbf{x}_i} (v_{2_i} - v_{2_j})^2 \quad (3)$$

where v_{2_i} is the i -th component of the eigenvector \mathbf{v}_2 associated to the eigenvalue λ_2 . Evaluation of (3) requires knowledge of λ_2 and of the local components of \mathbf{v}_2 (of i and its neighbors). These can be estimated online in a decentralized way as explained in [9]. If the weights A_{ij} are properly designed so that they (and their gradients) can be evaluated from locally available quantities, then the law (3) can be implemented by robot i in a decentralized way.

The robot model (1) considered in this work is, however, not a simple single integrator. Therefore, the gradient descent (3) needs to be adjusted accordingly. To this end, let us consider the time derivative of the cost V^λ

$$\begin{aligned} \dot{V}^\lambda &= \frac{\partial V^\lambda}{\partial \lambda_2} \dot{\lambda}_2 = \frac{\partial V^\lambda}{\partial \lambda_2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \left(\frac{\partial^T A_{ij}}{\partial \mathbf{p}_i} \dot{\mathbf{p}}_i + \frac{\partial A_{ij}}{\partial \psi_i} \dot{\psi}_i \right) (v_{2_i} - v_{2_j})^2 \\ &= \frac{\partial V^\lambda}{\partial \lambda_2} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \left(\frac{\partial^T A_{ij}}{\partial \mathbf{p}_i} \mathbf{R}_i \mathbf{v}_i + \frac{\partial A_{ij}}{\partial \psi_i} w_i \right) (v_{2_i} - v_{2_j})^2 \end{aligned} \quad (4)$$

where we used (1). By inspection, the i -th robot body-frame velocity commands for implementing a gradient descent of V^λ are then

$$\begin{cases} \mathbf{v}_i &= -\frac{\partial V^\lambda}{\partial \lambda_2} \sum_{j \in \mathcal{N}_i} \mathbf{R}_i^T \frac{\partial A_{ij}}{\partial \mathbf{p}_i} (v_{2_i} - v_{2_j})^2 \\ w_i &= -\frac{\partial V^\lambda}{\partial \lambda_2} \sum_{j \in \mathcal{N}_i} \frac{\partial A_{ij}}{\partial \psi_i} (v_{2_i} - v_{2_j})^2 \end{cases} \quad (5)$$

The goal of the next sections is to detail the design of weights A_{ij} for the particular sensing model considered in this work and to then analyze the implementability of (5) w.r.t. the quantities assumed available to robot i .

A. Design of the Weights

As discussed in Sect. II-A, we consider the presence of an onboard camera with limited FOV and of a (simpler) omnidirectional distance sensor. The camera has a minimum and maximum detection range, while the distance sensor has only a maximum range value to consider. For simplicity, we assume that the maximum range of the camera equals the maximum communication range of the robots. The weight A_{ij} is then designed as the product of two sub-weights

$$A_{ij} = \gamma_{ij} f_{ij}, \quad (6)$$

where γ_{ij} accounts for the min/max detection range and f_{ij} for the camera FOV. Similarly to [3], let us introduce two (classical) rising/falling activation functions

$$w_1^0(x, x^d, x^D) = \begin{cases} 0 & x \leq x^d \\ \frac{1}{2}(1 - \cos(ax + b)) & x^d < x \leq x^D \\ 1 & x^D < x \end{cases} \quad (7)$$

and

$$w_0^1(x, x^d, x^D) = \begin{cases} 1 & x \leq x^d \\ \frac{1}{2}(1 + \cos(ax + b)) & x^d < x \leq x^D \\ 0 & x^D < x \end{cases} \quad (8)$$

with $a = \pi/(x^D - x^d)$ and $b = -ax^d$, and lower/upper thresholds set at x^d and x^D (see Fig. 3 for an example).

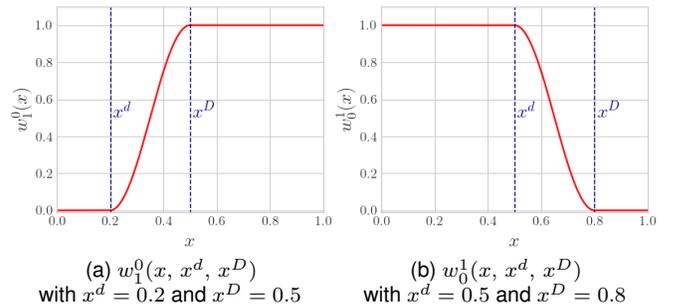


Fig. 3. The shape of the two considered rising/falling activation functions.

Weight γ_{ij} is then defined as

$$\gamma_{ij}(d_{ij}) = w_1^0(d_{ij}, d_{\min}, D_1) w_0^1(d_{ij}, D_2, d_{\max}), \quad (9)$$

where $d_{ij} = \|\mathbf{p}_{ij}\|$ is the distance among the robots and $d_{\min} < D_1 < D_2 < d_{\max}$ two suitable thresholds representing the min/max detection ranges.

As for the FOV weight, we define $\beta_{ij} = \frac{\mathbf{p}_{ij}}{d_{ij}}$ as the (unit-norm) bearing vector from i to j in the body-frame of robot i (note that β_{ij} can be directly computed from the measured \mathbf{p}_{ij}). Let also \mathbf{o}_C be the fixed body-frame direction of the camera optical axis. The quantity $a_{ij} = \mathbf{o}_C^T \beta_{ij}$ (the cosine of the angle between β_{ij} and \mathbf{o}_C) can therefore be leveraged for defining the FOV weight. Towards this end, we start by defining the function

$$f_{ij}^*(a_{ij}) = w_1^0(a_{ij}, a_{\min}, a_{\max}), \quad (10)$$

where a_{\max} is the cosine of the FOV limit (expressed as a maximum angular displacement from \mathbf{o}_C). Although tempting,

function f_{ij}^* cannot be directly used as the sought FOV weight, as it is not symmetric, i.e., in general, $f_{ij}^* \neq f_{ji}^*$. We then propose to define the FOV weight as

$$f_{ij} = f_{ij}^* + f_{ji}^* - f_{ij}^* f_{ji}^*. \quad (11)$$

This definition has two advantages: first of all, f_{ij} is now symmetric, as it can be easily verified. Furthermore, $f_{ij} = 1$ whenever $f_{ij}^* = 1$ or $f_{ji}^* = 1$ (or both), and $f_{ij} \rightarrow 0$ only when both $f_{ij}^* \rightarrow 0$ and $f_{ji}^* \rightarrow 0$. In other words, the FOV weight f_{ij} will not force robots i and j to simultaneously look at each other; instead, it will only enforce that at least one of the robots keeps the other one in sight. This is, in our opinion, an important feature for not *overconstraining* the robot group during motion.

B. Weight Gradient

The structure of the gradient of weights A_{ij} (6) needed for implementing (5) is as follows. From (6), one can obtain

$$\begin{aligned} \mathbf{R}_i^T \frac{\partial A_{ij}}{\partial \mathbf{p}_i} = & -f_{ij} \frac{\partial \gamma_{ij}}{\partial d_{ij}} \boldsymbol{\beta}_{ij} + \gamma_{ij} \left(-(1 - f_{ji}^*) \frac{\partial f_{ij}^*}{\partial a_{ij}} \frac{\mathbf{P}_{ij}}{d_{ij}} \mathbf{o}_C \right. \\ & \left. + (1 - f_{ij}^*) \frac{\partial f_{ji}^*}{\partial a_{ji}} {}^i \mathbf{R}_j \frac{\mathbf{P}_{ji}}{d_{ij}} \mathbf{o}_C \right) \end{aligned} \quad (12)$$

and

$$\frac{\partial A_{ij}}{\partial \psi_i} = -\gamma_{ij} (1 - f_{ji}^*) \frac{\partial f_{ij}^*}{\partial a_{ij}} \mathbf{o}_C^T \mathbf{S} \boldsymbol{\beta}_{ij}, \quad (13)$$

where ${}^i \mathbf{R}_j = \mathbf{R}_i^T \mathbf{R}_j = \mathbf{R}_z(\psi_j - \psi_i)$ depends on the relative rotation among the body frames of robot j and i . The detailed expression of the various terms in (12–13) is reported in the Appendix.

C. Discussion

We now discuss the maintenance controller (5)–(12–13) in terms of the quantities actually needed by robot i for implementing it. From inspection of the various terms (see also the Appendix), one can conclude that evaluation of (12) requires knowledge of d_{ij} , $\boldsymbol{\beta}_{ij}$, the relative angle $\psi_{ij} = \psi_j - \psi_i$ (for evaluating ${}^i \mathbf{R}_j$), and $\boldsymbol{\beta}_{ji} = -{}^i \mathbf{R}_j \boldsymbol{\beta}_{ij}$. Evaluation of (13) requires again knowledge of d_{ij} , $\boldsymbol{\beta}_{ij}$, and $\boldsymbol{\beta}_{ji}$. Sect. IV will discuss how an estimation of the relative angle ψ_{ij} can be maintained by robot i during motion. We then distinguish the following cases:

- 1) robot i can measure robot j : in this case, robot i has access to all the needed quantities, since d_{ij} , $\boldsymbol{\beta}_{ij}$ and $\boldsymbol{\beta}_{ji}$ can be obtained from the measured \mathbf{p}_{ij} and the estimated angle ψ_{ij} ;
- 2) robot i cannot measure robot j but it is measured by robot j : in this case, robot j needs to communicate to robot i the measured \mathbf{p}_{ji} and estimated ψ_{ji} from which robot i can again compute all the needed quantities;
- 3) robot i and robot j do not measure themselves: in this case the terms $\partial A_{ij} / \partial \mathbf{p}_i = 0$ and $\partial A_{ij} / \partial \psi_i = 0$ (and alike for robot j).

Concerning the symmetry of weights A_{ij} in (6), we note that the weight $f_{ij} = f_{ji}$ is symmetric by construction, thanks to

the definition (11), and that $\gamma_{ij} = \gamma_{ji}$ since $d_{ij} = d_{ji}$. The only remark is that, in the case 2) listed above robot, i still needs to compute the weight γ_{ij} , even if robot j is not in visibility, so as to ensure symmetry of the total weight A_{ij} . This can of course be done using the quantity \mathbf{p}_{ji} received from robot j .

In summary, the weights A_{ij} can be made symmetric and the maintenance controller (5)–(12–13) can be implemented by the robots in the group with the requirement that, if a robot i measures another robot j , then it needs to communicate its measured relative position \mathbf{p}_{ij} and current estimation of ψ_{ij} to robot j . Since this amount of information is constant per neighbor, we can conclude that the overall maintenance scheme is decentralized and scalable.

D. Collision Avoidance

Collision avoidance with obstacles or other robots is dealt with in a simple yet practical way. Let \mathbf{o}_i^k be the position of a k -th obstacle point in the frame of robot i , detected by the onboard sensor. For each obstacle point inside the sensing range d_{max} , we define a repulsive velocity command $\mathbf{v}_{i,obs}^k = -k_{obs}(\|\mathbf{o}_i^k\|) \mathbf{o}_i^k$ where the gain $k_{obs}(\|\mathbf{o}_i^k\|) \geq 0$ is any monotonic function with a vertical asymptote at $\|\mathbf{o}_i^k\| = d_{min}^{obs}$ and vanishing (with vanishing slope) for $\|\mathbf{o}_i^k\| \rightarrow d_{max}^{obs}$, where $0 < d_{min}^{obs} < d_{max}^{obs}$ is a safety distance. The i -th robot velocity command \mathbf{v}_i from (5) is then augmented with the additional term $\mathbf{v}_{i,obs} = \sum_k \mathbf{v}_{i,obs}^k$ for all detected obstacle points \mathbf{o}_i^k . This simple reactive strategy is sufficient to handle collision avoidance in most practical situations. Stability of the coupling between connectivity maintenance and obstacle repulsion can be shown by following the same arguments as in [12]: in short, for any bounded external action (in our case, a bounded external command from the human operator), the closed-loop system is stable besides practically improbable (zero-measure) situations of perfect (and unlikely) alignments among the robots.

IV. ESTIMATION OF THE RELATIVE ORIENTATION

As explained in the previous section, for implementing the maintenance controller, each robot i needs to know the relative orientation $\psi_{ij} = \psi_j - \psi_i$ w.r.t. the other robots it is interacting with. This is a typical requirement common to previous works sharing the same robot model (1), see, e.g., [10], [13], [18] and references therein. In general, in these prior works the problem of obtaining ψ_{ij} is addressed by assuming special structures of the interaction graph \mathcal{G} . In [18], any interacting pair of robots is always assumed in mutual visibility so that both $\boldsymbol{\beta}_{ij}$ and $\boldsymbol{\beta}_{ji}$ are available. This, as well-known, allows for an algebraic computation of ψ_{ij} . In [10], [13], constant mutual visibility is not required, but the graph is assumed to be (*bearing*) *rigid*. The rigidity condition allows implementing a simple decentralized localization algorithm that allows estimating ψ_{ij} from the measured bearings. These solutions however cannot be used in the setting considered in this work as we *do not* require mutual visibility nor maintenance of graph rigidity (which is a much stronger requirement than just graph connectivity). Therefore, we detail

here an alternative (and, to our knowledge, novel) scheme for retrieving an estimation of ψ_{ij} .

Remark 1. *Note that we do not (purposely) assume the relative yaw angle $\psi_{ij} = \psi_j - \psi_i$ to be directly measurable from the onboard camera: indeed, while extracting a relative position is (relatively) feasible, obtaining a relative orientation information from vision is in practice much harder because of the typical rotational symmetries of the UAVs. Therefore we consider \mathbf{p}_{ij} as the only information that can be reliably extracted from the images. Also, while from the onboard IMUs one could obtain a (noisy) measurement of $\dot{\psi}_{ij}$, a plain forward integration of $\dot{\psi}_{ij}$ would clearly quickly drift over time. These considerations then motivate the estimation algorithm described hereafter.*

We start by considering the dynamics of the measurement (2)

$$\dot{\hat{\mathbf{p}}}_{ij} = -\mathbf{v}_i - \mathbf{S}\mathbf{p}_{ij}w_i + {}^i\mathbf{R}_j\mathbf{v}_j. \quad (14)$$

We note that in (14) the only unknown term is ${}^i\mathbf{R}_j\mathbf{v}_j$, since \mathbf{v}_i , w_i , and \mathbf{p}_{ij} are assumed available to robot i . This motivates to consider the use of a “disturbance observer” for estimating the unknown ${}^i\mathbf{R}_j\mathbf{v}_j$, seen as a disturbance acting on the dynamics of \mathbf{p}_{ij} . In particular, let $\hat{\mathbf{p}}_{ij}$ be an estimation of \mathbf{p}_{ij} maintained by robot i and updated with the following law

$$\dot{\hat{\mathbf{p}}}_{ij} = -\mathbf{v}_i - \mathbf{S}\mathbf{p}_{ij}w_i + k_1(\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}) \quad (15)$$

with $k_1 > 0$. Let us also define the quantity

$$\mathbf{e} = k_1(\mathbf{p}_{ij} - \hat{\mathbf{p}}_{ij}) \quad (16)$$

which, using (14–15), has dynamics

$$\dot{\mathbf{e}} = k_1({}^i\mathbf{R}_j\mathbf{v}_j - \mathbf{e}). \quad (17)$$

In Laplace domain, this is equivalent to

$$\mathbf{e}(s) = \frac{k_1}{s + k_1} {}^i\mathbf{R}_j\mathbf{v}_j. \quad (18)$$

Therefore, the quantity $\mathbf{e}(t)$ (which can be evaluated by robot i) acts as a first-order low-pass filtered version of the unknown ${}^i\mathbf{R}_j\mathbf{v}_j$ with a cut-off frequency given by gain k_1 . Robot i can then (algebraically) obtain an estimation $\hat{\psi}_{ij}$ by computing the angle around the vertical axis among the vectors \mathbf{e} (approximating ${}^i\mathbf{R}_j\mathbf{v}_j$) and \mathbf{v}_j (communicated by robot j).

This estimation strategy works reasonably well, but it has some shortcomings. First of all, the estimated $\hat{\psi}_{ij}$ is reliable only if $\mathbf{v}_j \neq \mathbf{0}$ and if \mathbf{v}_j is not aligned with the vertical direction (since, in this case, ${}^i\mathbf{R}_j\mathbf{v}_j = \mathbf{v}_j$). This latter situation is, in practice, very rare, but the former situation ($\mathbf{v}_j \approx \mathbf{0}$) can frequently occur during the group motion. Second, the estimation dynamics (15) does not exploit knowledge of the angular velocity w_j , which is instead an available quantity. Third, when robots i and j are in mutual visibility, the relative angle ψ_{ij} can be algebraically computed from the two measured relative positions \mathbf{p}_{ij} and \mathbf{p}_{ji} , see, e.g., [18]. The possible availability of a direct measurement of ψ_{ij} , although possibly noisy, is also not exploited by (15).

To address these points, we propose an improved estimation strategy based on the previous one. Let $\tilde{\psi}_{ij}$ be the state of the second estimator maintained by robot i . Since $\dot{\psi}_{ij} = w_j - w_i$, we choose the following estimation dynamics

$$\begin{aligned} \dot{\tilde{\psi}}_{ij} = w_j - w_i + k_3 f_{ij}^* f_{ji}^* \sin(\psi_{ij} - \tilde{\psi}_{ij}) \\ + k_2(\|\mathbf{v}_j\|)(1 - f_{ij}^* f_{ji}^*) \sin(\hat{\psi}_{ij} - \tilde{\psi}_{ij}) \end{aligned} \quad (19)$$

where $k_2(\|\mathbf{v}_j\|) = \bar{k}_2 w_1^0(\|\mathbf{v}_j\|, v_{min}, v_{max})$, $0 < v_{min} < v_{max}$ are lower/upper threshold for the norm of \mathbf{v}_j and $\bar{k}_2 > 0$ and $k_3 > 0$ are estimation gains. The rationale behind (19) is as follows: if robots i and j are in mutual visibility, then $f_{ij}^* f_{ji}^* = 1$ and (19) reduces to

$$\dot{\tilde{\psi}}_{ij} = w_j - w_i + k_3 \sin(\psi_{ij} - \tilde{\psi}_{ij}), \quad (20)$$

i.e., a forward odometry propagation with a correction term on the “measured” (but possibly noisy) relative angle ψ_{ij} (the use of the $\sin(\cdot)$ allows dealing with possible jumps of $\pm 2\pi$ in ψ_{ij}). Note that, in this case, both robots i and j will implement their own instance of (20): they can then exchange their estimated $\tilde{\psi}_{ij}$ and $\tilde{\psi}_{ji}$ for obtaining an average of the two values (exploiting the fact that $\psi_{ij} = -\psi_{ji}$).

If, instead, the two robots are not in mutual visibility but robot i can measure robot j (the other case being symmetrical), then $f_{ij}^* f_{ji}^* = 0$ and (19) reduces to

$$\dot{\tilde{\psi}}_{ij} = w_j - w_i + k_2(\|\mathbf{v}_j\|) \sin(\hat{\psi}_{ij} - \tilde{\psi}_{ij}), \quad (21)$$

which is again a forward odometry propagation but now using the estimated $\hat{\psi}_{ij}$ in the correction term. Since, as explained above, the reliability of the estimated $\hat{\psi}_{ij}$ decreases with the norm of \mathbf{v}_j , the gain k_2 is designed so as to vanish for a $\|\mathbf{v}_j\| \rightarrow v_{min}$ so that, when $\|\mathbf{v}_j\|$ is too small, (19) further reduces to $\dot{\tilde{\psi}}_{ij} = w_j - w_i$.

Robot i can then communicate $\tilde{\psi}_{ji} = -\tilde{\psi}_{ij}$ to robot j , who needs this information for implementing the maintenance controller. Finally, in any intermediate case, i.e., when $0 < f_{ij}^* f_{ji}^* < 1$, the estimator (19) will blend the two cases in a continuous way.

V. RESULTS

This section presents the results of a simulation in which $N = 6$ UAVs were controlled by the connectivity-maintenance algorithm and obstacle avoidance actions discussed in the previous sections.¹ We implemented the proposed control and estimation algorithms using decentralized ROS 2 instances, a custom Unity scene to visualize the UAV behavior, and *robotpkg* from LAAS’ Openrobots project [21] to simulate their dynamics². The parameters used in the algorithm are reported in Table I.

¹Experiments presented in this paper were carried out thanks to a platform of the Robotex 2.0 French research infrastructure.

²Note that the employed simulation environment simulates the full 3D quadrotor dynamics in presence of actuation noise and not the simplified model (1) used only for control design. Therefore, the reported results are also meant to validate the portability of the proposed approach to a more realistic case closer to real-world conditions.

$d_{min} = 1 \text{ m}$	$a_{min} = 0.4$	$d_{min}^{obs} = 1.0 \text{ m}$	$k_1 = 10$
$D_1 = 3 \text{ m}$	$a_{max} = 0.95$	$d_{max}^{obs} = 3.5 \text{ m}$	$\bar{k}_2 = 10$
$D_2 = 7 \text{ m}$	$\mathbf{o}_C =$	$\lambda_2^{min} = 0.01$	$k_3 = 10$
$d_{max} = 9 \text{ m}$	$[1, 1, 0]/\sqrt{2}$	$\lambda_2^{max} = 3.01$	

TABLE I

ALGORITHM PARAMETERS USED IN OUR VALIDATION.

A human operator was asked to control the multi-robot team in a 3-dimensional space to reach a predefined target, employing the “leader-follower” paradigm presented in [9]. Using an analog joystick, the human operator imparts an additional input velocity to one robot (the leader) for controlling its motion. As all the robots are subject to the connectivity-maintenance action of (3), the team follows the movement of the leader so as to keep the team connected³. A discussion on the effectiveness of this approach in controlling multi-robot teams is presented in [3].

The task begins with the team in a sparse random configuration. At the very beginning, the connectivity-maintenance algorithm makes the robots reaching a (local) maximum for the group connectivity by creating new links whenever two robots start sensing each other. Then, as the human user imparts external commands to the robot leader $i = 1$ to reach the target, the algorithm keeps adjusting the robot formation so as to satisfy the constraints described in Sect. III, including the newly-introduced FOV weight.

We strongly encourage the reader to watch the full simulation in the video available at: <https://youtu.be/CdnAmIrdBQU>, where the various details of our approach can be better appreciated. Trajectories of the experiment present in both the video attachment and at 43 seconds in the online video (<https://youtu.be/CdnAmIrdBQU?t=43>) are shown on Fig. 4 and Fig. 5.

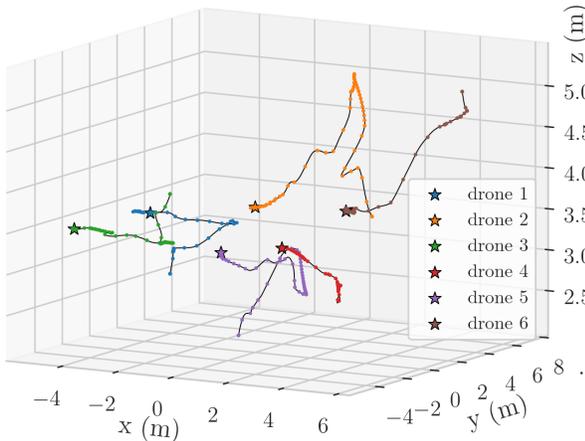


Fig. 4. Drones trajectories during the experiment. Drone 1 is the drone controlled by the human user. Each dot in the trajectory corresponds to the position of the drone at a 5 seconds interval.

³Since the external human input is bounded, the connectivity maintenance action will always be able to overcome it and prevent disconnection from the group.

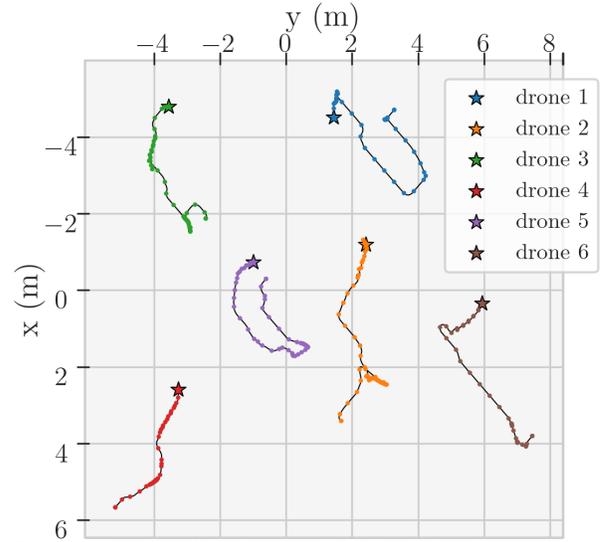


Fig. 5. Top view of the drones trajectories during the experiment.

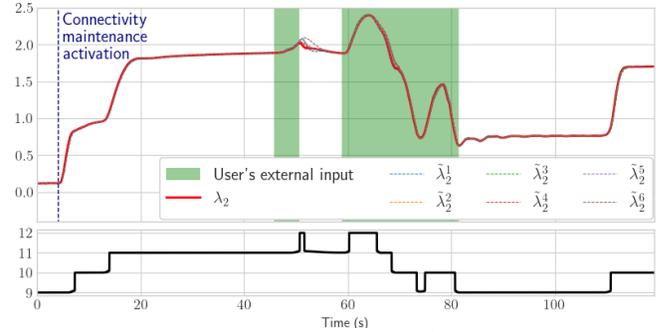


Fig. 6. Validation in simulation. (Top) True λ_2 vs. the six estimations $\tilde{\lambda}_2^i$ calculated by the robots over time; the green phases indicate when the human user is providing an external input to the robot leader (robot 1). (Bottom) Number of edges in the graph, which changes dynamically as the task evolves.

Figure 6 (top) reports the behavior of the connectivity eigenvalue $\lambda_2(t)$ (solid red line) and the N estimations $\tilde{\lambda}_2^i$ maintained by the N robots during motion. One can note how the estimation error remains quite small despite the (sometimes) rapid changes in $\lambda_2(t)$ also due to the action of the human operator (green phases). Fig. 6 (bottom) reports the number of edges during motion which is not constant, as expected.

Fig. 7 reports the error in estimating the relative orientation among robot 1 and robot 2 and 3. For the reader’s convenience, this estimation error is quantified by the quantity $\sin((\psi_{1j} - \tilde{\psi}_{1j})/2)$, which vanishes for $\psi_{1j} - \tilde{\psi}_{1j} = \pm 2k\pi$. The plot also shows, with colored bands, the four possible visibility combinations among robot 1 and the other robots (mutual visibility, one robot measuring the other but not being measured by it, the two robots not in visibility of each other). We can note that the estimation error is always limited. Moreover, when the two robots are in mutual visibility (green phase in Fig. 7) the error is close to zero as, in this case, a direct measurement of ψ_{1j} is available. When the two robots are instead completely disconnected (white phases), no measurement is available. In this case, the estimations of ψ_{1j} are not updated anymore, thus leading to an (unavoidable) increased estimation error, which is however

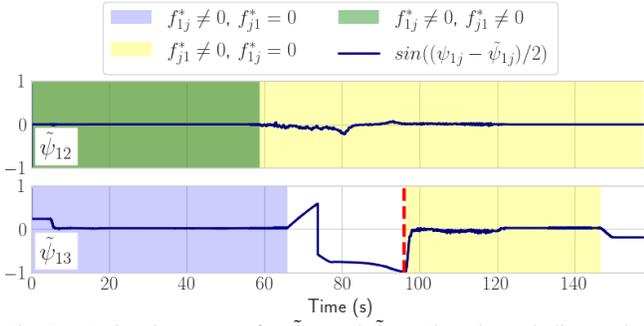


Fig. 7. Estimations errors for $\tilde{\psi}_{12}$ and $\tilde{\psi}_{13}$. Blue phases indicate when robot 1 is measuring robot j ($f_{1j}^* \neq 0$, see Sect. III-A) but robot j does not measure robot 1 ($f_{j1}^* = 0$); yellow phases indicate when robot j is measuring robot 1 but robot 1 does not measure robot j ; green phases indicate when both robot 1 and j are measuring each other. Note how the estimation errors always remain very small apart from the white phase (bottom plot) since, in this case, the two robots are completely disconnected. The error however quickly converges back to zero as soon as one robot starts measuring the other one (vertical red dashed line).

quickly recovered when one of the robots (or both) become visible again (red dashed line). Estimation errors during blue and yellow phases can also temporarily increase when $w_i \neq 0$ or $w_j \neq 0$ but $v_j = 0$ (blue phase) or $v_i = 0$ (yellow phases) (see Sect. IV).

VI. DISCUSSION AND CONCLUSIONS

This paper presented a decentralized connectivity-maintenance control algorithm for a team of quadrotor UAVs having limited FOV and not sharing a common reference frame for collectively expressing measurements and commands. To the best of our knowledge, these contributions advance the existing literature on this subject which, instead, has been based on the (simplifying) assumptions of omnidirectional sensing and presence of a common frame for all the robots in the group. The proposed maintenance control action is therefore able to address a more realistic setting, closer to real-world conditions. The approach has been successfully validated in realistic simulations involving 6 quadrotor UAVs.

We are now working towards an experimental validation with real quadrotors in our arena. A limitation of the algorithm that we would like to address concerns model (1), which does not consider the tilting (roll/pitch) motion of a quadrotor needed to accelerate in space. Indeed, if this tilting is non-negligible (i.e., when far from near-hovering regimes), then the proposed FOV weight may not be able to accurately model the sensing limitations of an onboard camera, which would also depend on the quadrotor tilting angle. We are then considering how to extend model (1) (and the FOV weight design) for correctly modeling this effect. Another major challenge in vision based relative localization is determining the identity of the observed neighbors. For solving this challenge, we plan to use an addressable LED ring mounted on the drones and a RGBD camera with color filtering to retrieve both the relative position and the identity of the neighbors.

APPENDIX

We detail here the computation of the weight gradients presented in Sect. III-B. From (6), one clearly has

$$\frac{\partial A_{ij}}{\partial \mathbf{p}_i} = f_{ij} \frac{\partial \gamma_{ij}}{\partial \mathbf{p}_i} + \gamma_{ij} \frac{\partial f_{ij}}{\partial \mathbf{p}_i} \quad (22)$$

and

$$\frac{\partial A_{ij}}{\partial \psi_i} = f_{ij} \frac{\partial \gamma_{ij}}{\partial \psi_i} + \gamma_{ij} \frac{\partial f_{ij}}{\partial \psi_i}, \quad (23)$$

since γ_{ij} is only a function of the distance d_{ij} , it follows that

$$\frac{\partial \gamma_{ij}}{\partial \mathbf{p}_i} = -\frac{\partial \gamma_{ij}}{\partial d_{ij}} \frac{\mathbf{p}_j - \mathbf{p}_i}{d_{ij}} \quad (24)$$

and $\partial \gamma_{ij} / \partial \psi_i = 0$. As for $\partial f_{ij} / \partial \mathbf{p}_i$, from (11) one has

$$\frac{\partial f_{ij}}{\partial \mathbf{p}_i} = (1 - f_{ji}^*) \frac{\partial f_{ij}^*}{\partial \mathbf{p}_i} + (1 - f_{ij}^*) \frac{\partial f_{ji}^*}{\partial \mathbf{p}_i}. \quad (25)$$

and analogously for $\partial f_{ij} / \partial \psi_i$. Recalling the definition (10) and that $a_{ij} = \mathbf{o}_C^T \boldsymbol{\beta}_{ij}$, one obtains

$$\frac{\partial f_{ij}^*}{\partial \mathbf{p}_i} = \frac{\partial f_{ij}^*}{\partial a_{ij}} \frac{\partial^T \boldsymbol{\beta}_{ij}}{\partial \mathbf{p}_i} \mathbf{o}_C = -\frac{\partial f_{ij}^*}{\partial a_{ij}} \frac{\mathbf{R}_i \mathbf{P}_{ij}}{d_{ij}} \mathbf{o}_C \quad (26)$$

and

$$\frac{\partial f_{ji}^*}{\partial \mathbf{p}_i} = \frac{\partial f_{ji}^*}{\partial a_{ji}} \frac{\partial^T \boldsymbol{\beta}_{ji}}{\partial \mathbf{p}_i} \mathbf{o}_C = \frac{\partial f_{ji}^*}{\partial a_{ji}} \frac{\mathbf{R}_j \mathbf{P}_{ji}}{d_{ij}} \mathbf{o}_C, \quad (27)$$

where $\mathbf{P}_{ij} = \mathbf{I}_3 - \boldsymbol{\beta}_{ij} \boldsymbol{\beta}_{ij}^T$. See [13], [22] for further details about this derivation. Similar considerations also yield

$$\frac{\partial f_{ij}^*}{\partial \psi_i} = -\frac{\partial f_{ij}^*}{\partial a_{ij}} \mathbf{o}_C^T \mathbf{S} \boldsymbol{\beta}_{ij} \quad (28)$$

and $\partial f_{ji}^* / \partial \psi_i = 0$ with $\mathbf{S} = [[0 \ 0 \ 1]^T]_{\times}$ and $[\cdot]_{\times}$ being the usual skew-symmetric matrix operator. Therefore, we can conclude that, for what concerns (5), one has

$$\begin{aligned} \mathbf{R}_i^T \frac{\partial A_{ij}}{\partial \mathbf{p}_i} &= -f_{ij} \frac{\partial \gamma_{ij}}{\partial d_{ij}} \boldsymbol{\beta}_{ij} + \gamma_{ij} \left(-(1 - f_{ji}^*) \frac{\partial f_{ij}^*}{\partial a_{ij}} \frac{\mathbf{P}_{ij}}{d_{ij}} \mathbf{o}_C \right. \\ &\quad \left. + (1 - f_{ij}^*) \frac{\partial f_{ji}^*}{\partial a_{ji}} \mathbf{R}_j \frac{\mathbf{P}_{ji}}{d_{ij}} \mathbf{o}_C \right) \end{aligned} \quad (29)$$

and

$$\frac{\partial A_{ij}}{\partial \psi_i} = -\gamma_{ij} (1 - f_{ji}^*) \frac{\partial f_{ij}^*}{\partial a_{ij}} \mathbf{o}_C^T \mathbf{S} \boldsymbol{\beta}_{ij}, \quad (30)$$

where ${}^i \mathbf{R}_j = \mathbf{R}_i^T \mathbf{R}_j = \mathbf{R}_z(\psi_j - \psi_i)$ depends on the relative rotation among the body frames of robot j and i .

REFERENCES

- [1] R. Mendonça, M. M. Marques, F. Marques, A. Lourenco, E. Pinto, P. Santana, F. Coito, V. Lobo, and J. Barata, "A cooperative multi-robot team for the surveillance of shipwreck survivors at sea," in *Proc. IEEE OCEANS*, 2016, pp. 1–6.
- [2] R. R. Murphy, K. L. Dreger, S. Newsome, J. Roderick, E. Steimle, T. Kimura, K. Makabe, F. Matsuno, S. Tadokoro, and K. Kon, "Use of remotely operated marine vehicles at minamisanriku and rikuzentakata japan for disaster recovery," in *Proc. IEEE International symposium on safety, security, and rescue robotics*, 2011, pp. 19–25.

- [3] M. Aggravi, C. Pacchierotti, and P. Robuffo Giordano, "Connectivity-Maintenance Teleoperation of a UAV Fleet With Wearable Haptic Feedback," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1243–1262, Jul. 2021.
- [4] J. Cortés and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.
- [5] S. J. Kim and G. J. Lim, "Drone-aided border surveillance with an electrification line battery charging system," *Journal of Intelligent & Robotic Systems*, vol. 92, no. 3, pp. 657–670, 2018.
- [6] C. Pacchierotti, "Cutaneous haptic feedback for robotics and virtual reality," Habilitation à diriger des recherches (HDR), Université de Rennes 1 (UR1), 2022.
- [7] M. Zucker, S. Joo, M. X. Grey, C. Rasmussen, E. Huang, M. Stilman, and A. Bobick, "A general-purpose system for teleoperation of the drc-hubo humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 336–351, 2015.
- [8] 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 Transactions on Robotics*, vol. 28, no. 5, pp. 1019–1033, 2012.
- [9] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bühlhoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, Mar. 2013.
- [10] F. Schiano and P. Robuffo Giordano, "Bearing rigidity maintenance for formations of quadrotor UAVs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1467–1474.
- [11] A. Franchi and P. Robuffo Giordano, "Online Leader Selection for Improved Collective Tracking and Formation Maintenance," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 1, pp. 3–13, Mar. 2018.
- [12] 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 Transactions on Mechatronics*, vol. 18, no. 4, pp. 1334–1345, Aug. 2013.
- [13] F. Schiano, A. Franchi, D. Zelazo, and P. Robuffo Giordano, "A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 5099–5106.
- [14] Y. Kuriki and T. Namerikawa, "Consensus-based cooperative formation control with collision avoidance for a multi-uav system," in *Proc. American Control Conference*, 2014, pp. 2077–2082.
- [15] M. Aggravi, G. Sirignano, P. Robuffo Giordano, and C. Pacchierotti, "Decentralized Control of a Heterogeneous Human-Robot Team for Exploration and Patrolling," *IEEE Transactions on Automation Science and Engineering*, pp. 1–17, 2021.
- [16] M. Aggravi, A. A. S. Elsherif, P. Robuffo Giordano, and C. Pacchierotti, "Haptic-enabled decentralized control of a heterogeneous human-robot team for search and rescue in partially-known environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4843–4850, 2021.
- [17] 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," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2166–2173, 2018.
- [18] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bühlhoff, and P. Robuffo Giordano, "Modeling and Control of UAV Bearing Formations with Bilateral High-level Steering," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1504–1525, 2012.
- [19] A. Gasparri, L. Sabattini, and G. Ulivi, "Bounded control law for global connectivity maintenance in cooperative multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 700–717, 2017.
- [20] B. Capelli, H. Fouad, G. Beltrame, and L. Sabattini, "Decentralized connectivity maintenance with time delays using control barrier functions," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2021, pp. 1586–1592.
- [21] A. Mallet, F. Kanehiro, S. Fleury, and M. Herrb, "Reusable robotics software collection," in *Proc. Int. Workshop on Software Development and Integration in Robotics (SDIR)*, 2007.
- [22] D. Zelazo, P. Robuffo Giordano, and A. Franchi, "Bearing-Only Formation Control Using an $SE(2)$ Rigidity Theory," in *Proc. IEEE Conf. on Decision and Control*, 2015.