Distributed Control of a Limited Angular Field-of-View Multi-Robot System in Communication-Denied Scenarios: A Probabilistic Approach

Mattia Catellani, *Graduate Student Member, IEEE*, and Lorenzo Sabattini, *Senior Member, IEEE*

***Abstract*—Multi-robot systems are gaining popularity over single-agent systems for their advantages. Although they have been studied in agriculture, search and rescue, surveillance, and environmental exploration, real-world implementation is limited due to agent coordination complexities caused by communication and sensor limitations. In this work, we propose a probabilistic approach to allow coordination among robots in communication-denied scenarios, where agents can only rely on visual information from a camera with a limited angular field-of-view. Our solution utilizes a particle filter to analyze uncertainty in the location of neighbors, together with Control Barrier Functions to address the exploration-exploitation dilemma that arises when robots must balance the mission goal with seeking information on undetected neighbors. This technique was tested with virtual robots required to complete a coverage mission, analyzing how the number of deployed robots affects performances and making a comparison with the ideal case of isotropic sensors and communication. Despite an increase in the amount of time required to fulfill the task, results have shown to be comparable to the ideal scenario in terms of final configuration achieved by the system.**

***Index Terms*—Multi-robot systems, distributed robot systems.**

I. Introduction

MULTI-ROBOT systems are showing their potential in helping humans facing hard challenges nowadays, being extensively studied when employed in a wide range of applications, such as agriculture [1], search and rescue [2], target tracking [3], collaborative transportation [4], and monitoring and exploration of the environment, as we investigated in our previous work [5]. Yet their employment is still usually limited to laboratory environments, while only few real-world applications can be found [6], [7]. One major challenge limiting their adoption is dealing with non-ideal sensors. For instance, using cameras with a restricted angular field of view makes it difficult to perceive neighbors and coordinate effectively. Studies investigating anisotropic sensory have been made when applied to environmental perception [8], while communication among the agents is usually assumed to be isotropic. Limitations on communication capabilities lead to even more challenging problems in deploying a robotic team for real-world operations. This challenge is usually addressed only considering robots with a limited communication range designed as a sphere around the sensor, without taking into consideration other fundamental aspects such as the antenna radiation pattern, which is not isotropic, dropped packets, and delays. Moreover, sharing more information increases both hardware and software complexity, thus leading to scalability limitations [9]. In addition, extreme conditions may occur where communication is completely denied by environmental constraints, as in case of underwater or subterranean operations, or artificially jammed, e.g. in military missions.

The contribution of this work is the definition of an optimization-based visual control for decentralized multi-robot systems In particular, a probabilistic strategy is defined, in order to face the exploration-exploitation trade-off, i.e. choosing whether to search for neighbors and gaining information on their position or to exploit information to pursue the goal of the mission. A properly defined particle filter will be designed with the aim of gaining information on the location of each neighbor and the related uncertainty. Subsequently, Control Barrier Functions (CBFs) will be designed constraining the motion of the controlled robot to keep a target neighbor inside its own field of view. Finally, slack variables will be used to handle the exploration-exploitation dilemma, softening CBFs constraints, thus allowing the robot to lose visual contact with the target in order to pursue the global goal of the task. The proposed methodology is tested when applied to coverage control of a team of quadrotors in simulations.

II. Related Work

Only few research works can be found in the literature going as far as to investigate such extreme conditions. A solution is proposed in [10] for flocking control of unmanned aerial vehicles (UAVs). The team is characterized as a continuous fluid and

Manuscript received 21 July 2023; accepted 17 November 2023. Date of publication 29 November 2023; date of current version 8 December 2023. This letter was recommended for publication by Associate Editor G. Pereira and Editor M. A. Hsieh upon evaluation of the reviewers' comments. This work was supported by the Italian Ministry for University and Research through the COWBOT Project, under the PRIN 2020 program. *(Corresponding author: Mattia Catellani.)*

The authors are with the Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, 41121 Modena, Italy (e-mail: mattia.catellani@unimore.it; lorenzo.sabattini@unimore.it).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3337694, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3337694

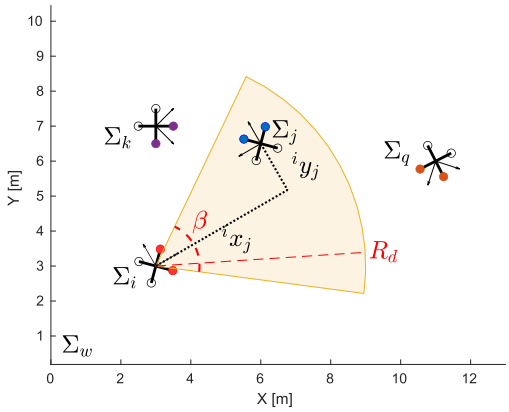

Fig. 1. Visualization of reference system and field of view of drone i, namely F_i, defined by a cone with amplitude β and radius R_d. In particular, only drone j is being detected, since k is on the left side of the visible space and q is too far away.

hydrodynamics laws are exploited to guarantee swarm cohesion, collision avoidance, and velocity consensus while taking into account latency within inter-agent communication and constraints resulting from the 3D antenna radiation pattern. Other interesting approaches make use of probabilistic information about the position of undetected neighbors, like Bayes multi-target filters and Probability Hypothesis Density (PHD) filters. A Gaussian Mixture PHD is proposed in [11] to build a localization system estimating positions of robots even when communication is lost for a long duration, evaluating the motion model of the system and selecting places where lost robots are more likely to be at each moment. Another probabilistic approach is presented in [12], where a collision avoidance method is proposed, accounting for both measurement and motion uncertainty. The authors exploit Probabilistic Safety Barrier Certificates (PrSBC) using Control Barrier Functions to define the space of probabilistically safe control actions.

Our work will combine CBFs with probabilistic analysis in order to coordinate the team and complete the mission without collisions among the agents.

III. Preliminaries

A. Notation and Definitions

In the rest of the letter, we denote by $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ the set of natural, real, real non-negative, and real positive numbers, respectively. Given $x \in \mathbb{R}^n$, let $\|x\|$ be the Euclidean norm.

Let $\mathbb{F}(\mathbb{R}^2)$ be the collection of finite point sets in $\mathbb{R}^2$. We denote an element of $\mathbb{F}(\mathbb{R}^2)$ as $\mathcal{P} = \{p_1, \ldots, p_n\} \subset \mathbb{R}^2$, where $\{p_1, \ldots, p_n\}$ are points in $\mathbb{R}^2$.

In the letter, $Q \subset \mathbb{R}^2$ denotes a generic polygon: it will be used, in particular, to denote the environment where the robots move. An arbitrary point in Q is denoted by $q \in Q$. We consider robots moving in a closed environment free of obstacles. The inertial reference frame associated with the environment is denoted with Σ_w. The local reference frame attached to the body of each robot i in the environment is denoted with Σ_i. Fig. 1 provides a visualization of the considered reference frames. The notation ${}^w\Sigma_i$ denotes the transformation from the global reference frame Σ_w to the local frame Σ_i expressed in the global frame. Similarly, the notation ${}^i\boldsymbol{x}$ indicates a generic vector $\boldsymbol{x}$ expressed relatively to Σ_i. If not specified, a vector is expressed with respect to Σ_w.

B. Problem Description

Consider a team of N robots moving in a 2D obstacle-free bounded environment tasked with a coverage mission. The robots navigate only exploiting locally available information from an on-board camera and can not communicate with each other. Odometry data is not available for self-localization, but we assume robots are able to calculate their control input locally from a probability density function defining a target area to be reached in order to accomplish the global task. The above conditions can indistinctly apply to ground robots and drones assumed to fly at the same constant altitude and moving slowly enough that their pitch and roll angles are negligible. For this reason, we will mainly consider aerial robots in the rest of the letter, but the theoretical discussion can equally be applied to ground robots. We assume each robot can detect the position of neighbors that are in the field of view of its camera. The field of view can be expressed as a limited angular sector centered at local X axis with amplitude β and radius $R_d \in \mathbb{R}_{>0}$, as visualized in Fig. 1. We will indicate the area corresponding to the field of view of robot i as $F_i \subset \mathbb{R}^2$. Considering the i-th robot, the collection of all the other $N-1$ robots in the team will be indicated as $\mathcal{N}_i$. The 2D position of the neighbor j with respect to Σ_i is expressed as ${}^i\boldsymbol{p}_j = [{}^i x_j, {}^i y_j]^T$ (see Fig. 1). Furthermore, it is assumed that each robot has a unique ID and can visually recognize the ID of every detected neighbor. According to [13], ID recognition is possible when UAVs are equipped with blinking UV lights and a specific blinking frequency is assigned to each ID.

The state of each robot i is denoted with ${}^w\boldsymbol{\chi}_i = [{}^w x_i, {}^w y_i, {}^w \theta_i]^T$, where ${}^w x_i, {}^w y_i \in \mathbb{R}$ represent the coordinates of the 2D position of Σ_i in Σ_w while ${}^w\theta_i$ is the yaw angle. In case of a team of aerial robots, the altitude is excluded since we assume every robot flies at the same constant altitude. The robot is controlled according to the kinematic model ${}^i\dot{\chi}_i = {}^i\boldsymbol{u}_i = [{}^i v_x, {}^i v_y, {}^i \omega]^T$ where ${}^i v_x, {}^i v_y, w \in \mathbb{R}$ represent the linear and angular velocity input for the agent in Σ_i.

IV. Neighbors' Positions Estimation

In our work, we make use of particle filters to estimate the 2D position of neighbors. More in details, a particle filter is a recursive Bayesian state estimator that uses discrete particles to approximate the posterior distribution of an estimated state. Its algorithm is useful for online state estimation of a non-linear system according to the dynamic model of the robot and measurements taken (see [14]). Process and measurement noise distribution are also taken into account, resulting in the definition of a probability distribution of the real robot's state. The procedure for particle filter state estimation is presented in details in [15].

A. Multiple Particle Filters With Particles Deletion

In order to have an estimate on the position of its neighbors, each robot is required to run onboard a particle filtering algorithm for every other robot j in the team. As previously

stated, we assume every robot has a unique ID, and this ID can be recognized by other agents when detected, therefore no effort regarding association between a detected drone and state estimates is required. Since no information can be obtained from the i-th agent for robots lying outside its field of view F_i, we make use of an additional intermediate step to improve the estimation accuracy, while the other steps are the same as the traditional procedure.

1) *Initialization:* A specified number of particles $P \in \mathbb{N}$ is generated according to a known distribution or uniformly distributed within the environment. Each particle represents an hypothesis of the state variables.

$$\chi(0)_k \sim p\left(\chi(0)\right) \tag{1}$$

In the above equation, $\chi(0)_k$ is the initial state of the generic particle k, $p(\chi(0))$ is the initial distribution and the operator $\sim$ is used to denote that the particle is randomly obtained from the probability distribution.

2) *Prediction:* The state of each particle is propagated following the state transition model of the system $f(\chi(t-1)_k, u(t))$, where $u(t)$ is the control input.The result is a new particles distribution.

$$\chi(t)_k = f\left(\chi\left(t-1\right)_k, u(t)\right) \tag{2}$$

Since the kinematic model requires a velocity input, an estimate is provided according to the common goal of the team. Otherwise, if an estimate on the input of drone j can not be derived, the worst case scenario is considered, i.e. robot j is moving towards robot i at maximum speed.

3) *Weight Update:* The likelihood of sensor measurements $\gamma(t)$ is exploited to update the weight of each particle.

$$w(t)_k = p\left(\gamma(t)|\chi(t)_k\right) \tag{3}$$

The measurement uncertainty was considered as a matrix $\Sigma_{MEAS} \in \mathbb{R}^{3\times3}$, which was assumed to be independent from the position of the robot inside the field of view in order to simplify the discussion.

4) *Particles Deletion:* In case robot i is not able to detect j, the only correction that can be made is deleting samples inside robot i's field of view F_i, since j would be detected if it was there. Particles inside the field of view are removed by assigning them a null weight:

$$w(t)_k = 0 \quad \text{if } j \notin F_i \text{ and } k \in F_i \tag{4}$$

5) *Resampling:* The particles are resampled according to their weights, in order to give more weight to particles that are more likely to match observed data. This means that the new set of particles will be more concentrated in regions of the state space that have the highest probability, and no samples will be placed inside F_i thanks to the particles deletion step.

6) *State Estimation:* The state estimate $\hat{\chi}(t)$ is calculated as a weighted sum of particles.

$$\hat{\chi}(t) = \sum_{k=1}^{P} \frac{w(t)_k \chi(t)_k}{w(t)_k} \tag{5}$$

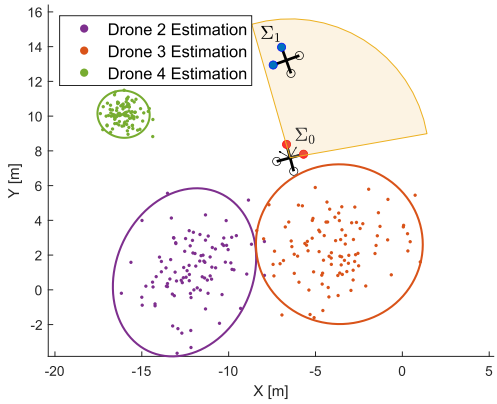

Fig. 2. 95% Confidence Ellipses of undetected neighbors.

It is interesting to note that the particle filter provides an estimate of both position and orientation, even if our control strategy will only exploit the position estimate. Orientation can be helpful in some situations, but for that to happen, measurements must have accurate information about it, which visual sensors may lack.

In Fig. 2, drone 0 visually gets drone 1's relative position, but can only estimate the location of undetected neighbors.

B. Samples Budget Re-Allocation

The adoption of several particle filters raises the problem of computational effort required to each robot, and makes the solution not scalable since a larger number of robots leads to the implementation of more filters. Our solution to overcome this problem and achieve a scalable architecture is defining a fixed total budget of samples P_T that have to be allocated to the neighbors at each time instant.

Let us assume, for now, that we can define a relative importance among the neighbors of robot i, and therefore we can sort them from the most to the least important. The importance parameter will be discussed in Section V. The key idea is to assign more particles to neighbors we are more interested in tracking based on the aforementioned importance, resulting in a more accurate evaluation. Conversely, the state estimation of less important robots can be performed with lower accuracy, so fewer particles are required. Therefore, after sorting neighbors, the number of required particles P_j is calculated according to their relative importance as a fraction of the total budget of particles P_T. Then, another resampling step is performed in order to draw a new samples set with P_j elements, without modifying the current probability distribution. In this way, the evaluation on neighbors' position can be considered scalable and only depends on P_T, regardless of the number of robots in the team.

The possibility of defining a fixed total number of samples, together with propagating particles only outside the field of view, are the main reasons for the adoption of particle filters for estimating the relative position of every other neighbors, instead of other solutions that would have been possibly less computationally expensive (e.g. Extended Kalman Filters).

V. Uncertainty Evaluation

In the rest of the letter, particles distribution will be investigated to gain information on the probability associated with the location of an undetected neighbor. Since our focus is on the 2D relative position of drone j with respect to Σ_i, namely ${}^i\boldsymbol{p}_j = [{}^i x_j, {}^i y_j]^T$, particles distribution can be addressed as a bivariate normal distribution, whose key parameters are the mean point ${}^i\overline{\mu}_j \in \mathbb{R}^2$ and the covariance matrix ${}^i\overline{\Sigma}_j \in \mathbb{R}^{2\times 2}$. Approximating the particles distribution as a Gaussian distribution is needed to lower the required computational effort. As a matter of fact, fitting the samples set to a non-uniform or a Gaussian mixture distribution would make the computational burden too heavy to be carried out by a generally low performance platform as an aerial robot. Moreover, considering particles as normally distributed reflects reality when all particles are placed outside drone i's field of view, therefore none of them is deleted. The distribution becomes non-uniform, instead, when some particles are deleted, but we can consider them as a small fraction of the samples set, thus not influencing the overall distribution significantly. Furthermore, a confidence interval can be derived from the distribution according to a desired confidence level drawn as an ellipse. Therefore, the real position of the undetected robot will be located inside the ellipse with the desired confidence level. Let us consider a generic ellipse defined by

$$\left(\frac{x - x_C}{a}\right)^2 + \left(\frac{y - y_C}{b}\right)^2 = \alpha \tag{6}$$

where $q_C = [x_C, y_C]^T$ is the central point, $a, b \in \mathbb{R}_{>0}$ are the major and minor semi-axis respectively, and $\alpha \in \mathbb{R}_{>0}$ is the scale of the ellipse. Since we are considering particles sampled from a multivariate Gaussian distribution, particles' x-coordinates and y-coordinates are normally distributed too. Therefore, the left hand side of (6) represents the sum of squares of normally distributed samples, also known as a Chi-Squared distribution. This allows us to evaluate the Chi-Squared likelihood to define the scale α of the ellipse, according to Pearson's Chi-Squared test [16]. Parameters defining the confidence ellipse can be easily calculated from the mean point and the covariance matrix. In particular, the center q_C corresponds to the mean point $\overline{\mu}$, and major and minor semi-axis, a and b respectively, can be calculated from the eigenvalues of the covariance matrix as $a = \sqrt{\alpha\lambda_1}$ and $b = \sqrt{\alpha\lambda_2}$, where $\lambda_1, \lambda_2 \in \mathbb{R}$, with $\lambda_1 \geq \lambda_2$, are the eigenvalues of the covariance matrix. Finally, the inclination of the major axis with respect to the x-axis is calculated as

$$\psi = \arctan\frac{v_1(y)}{v_1(x)} \tag{7}$$

where $v_1(x), v_1(y) \in \mathbb{R}$ are the x and y components of the eigenvector $\boldsymbol{v}_1$ corresponding to the largest eigenvalue λ_1, respectively. The calculated ellipses are shown in Fig. 2.

The distance $d_{ij} \in \mathbb{R}_{>0}$ between robot i and the elliptical area related to each neighbor j is measured to evaluate the relative importance of each robot as mentioned in Section IV-B. To calculate the distance from robot i to j-th ellipse, let us consider the parametric version of (6), generalized to a rotated ellipse:

$$\begin{aligned} {}^i x &= {}^i(\overline{\mu}_x)_j + a_j \cos\left({}^i\psi_j\right)\cos\theta - b_j \sin\left({}^i\psi_j\right)\sin\theta \\ {}^i y &= {}^i\left(\overline{\mu}_y\right)_j + a_j \sin\left({}^i\psi_j\right)\cos\theta + b_j \cos\left({}^i\psi_j\right)\sin\theta \end{aligned} \tag{8}$$

The nearest point to robot i lying on j-th ellipse, namely ${}^i p_j^{cr}$, can be obtained from the inclination of the line linking j to i:

$$\theta_j^{cr} = \arctan\frac{{}^i x_j}{{}^i y_j}. \tag{9}$$

Substituting (9) in (8), we obtain coordinates of ${}^i p_j^{cr}$. Finally, d_{ij} is calculated as the Euclidean distance between p_i and p_j^{cr}:

$$d_{ij} = \|{}^i p_j^{cr} - {}^i p_i\| \tag{10}$$

The distance d_{ij} will also be exploited in the next sections as a trade-off parameter for the exploration-exploitation problem, defining the behaviour of the controlled robot to search for undetected neighbors or to fulfill the global task.

VI. CBFs With Field of View Constraints

In this section we will briefly explain how Control Barrier Functions can be employed to constrain the motion of a controlled robot in order to keep a target neighbor inside its field of view. We refer interested readers to [17] for more details on CBFs. Let $\mathcal{C} = \{\chi \in \mathbb{R}^m : h(\chi) \geq 0\}$ be the safe set containing all the configurations that satisfy the requirements, i.e. allow the robot to keep the target inside its field of view. The safe set is defined as the super level of a smooth function $h : \mathbb{R}^m \to \mathbb{R}$ with $\frac{\partial h}{\partial \chi}(\chi) \neq 0 \ \forall \chi \in \partial\mathcal{C}$. Once h has been defined, an optimization-based controller can be derived in such a way to minimally modify the desired control input $\boldsymbol{u}^*$, solving the following optimization problem:

$$\begin{aligned} \boldsymbol{u}(\chi) = \arg\min_{\boldsymbol{u}\in\mathbb{R}^m} \ & \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{u}^*\|^2 \\ \text{s.t. } & \dot{h}(\chi, \boldsymbol{u}) \geq -\alpha(h(\chi)) \end{aligned} \tag{11}$$

where α is an extended class $\mathcal{K}$ function.

The problem of balancing between exploration and exploitation will then be converted into softening or hardening those constraints, in order to define if the agent must stay inside the safe set or it is allowed to go outside. We will use Control Lyapunov Functions (CLFs) to stabilize the system to an equilibrium point, or set, which we will make equivalent to the safe set $\mathcal{C}$ in order to bring the robot back into it. More details on CLFs can be found in [18].

A. Field of View Constraints

The set of constraint equations forcing a robot to keep a target neighbor inside its field of view is derived from our previous work [19] with the addition of a further constraint on the maximum distance, and only depends on the 2D relative

position of robot j, namely ${}^i\mathbf{p}_j$. In particular:

$$h({}^i\boldsymbol{p}_j) = \begin{bmatrix} \tan(\beta/2) & 1 \\ \tan(\beta/2) & -1 \\ {}^ix_j & {}^iy_j \\ -{}^ix_j & -{}^iy_j \end{bmatrix} {}^i\boldsymbol{p}_j + \begin{bmatrix} 0 \\ 0 \\ -D_s^2 \\ R_d^2 \end{bmatrix}. \tag{12}$$

The condition $h(\cdot) \geq 0$ is equivalent to constraining robot j to be on the left of the right border of the field of view and vice-versa, while maintaining a minimum safe distance D_s and a maximum distance R_d equal to the detection range of robot i. Considerations on how we can derive a control input for robot i go out of the scope of this work, therefore we refer the reader to [19] for a detailed explanation on the formulation of the optimization problem (11) from the constraint (12).

It is important to highlight that the chosen CBF (12) can also force the controlled robot to go back into the safe set if outside. This means that the input $\boldsymbol{u}$ resulting from the QP problem will carry the agent inside the set $\mathcal{C}$, i.e. in a configuration where robot j lies inside F_i.

Theorem 1: Let $V(\chi) : \mathbb{R}^m \to \mathbb{R}$ be:

$$V(\chi) = \begin{Bmatrix} 0 & \text{if } \chi \in \mathcal{C} \\ -h(\cdot) & \text{if } \chi \notin \mathcal{C} \end{Bmatrix} \tag{13}$$

Consider the CBF $h({}^i\boldsymbol{p}_j)$ defined in (12). Then, $V(\chi)$ is a CLF for the system.

Proof: The candidate CLF (13) is continuously differentiable and positive definite by construction. We should demonstrate that the derivative $\dot{V}(\chi)$ is negative definite or semidefinite. Let us focus on the case $\chi \notin \mathcal{C}$:

$$\dot{V}(\chi) = -\dot{h}(\chi) \leq \alpha(h(\chi)) \tag{14}$$

From [19] we have $\alpha(h(\chi)) = \lambda h(\chi)$, where $\lambda \in \mathbb{R}_{\geq 0}$ is a positive constant. Since we know that $h(\chi) < 0$ if $\chi \notin \mathcal{C}$, we can combine the chain of inequalities and finally get

$$\dot{V}(\chi) < 0 \tag{15}$$

which ensures that $V(\chi)$ is a CLF and asymptotically stabilizes the set $\mathcal{C}$. ■

B. Slack Variables for Exploration-Exploitation

The task of keeping every robot $j \in \mathcal{N}_i$ inside F_i can potentially become impossible to solve, especially in case of teams with a high number of robots. Moreover, even if this condition could be satisfied, the set of feasible control actions would become very restricted and achieving the global goal of the mission could be highly complicated. For this reason, we face this problem as an exploration-exploitation trade-off, where robot i is required to search for j only when the distance to the j-th ellipse is low, indicating that a collision could potentially happen. Information on the estimated position and uncertainty ellipses are gained from the particle filter as described in Section V.

The implementation of this solution is carried out integrating slack variables into the optimization problem, softening CBFs constraints related to neighbors placed far away from i, therefore there is no danger of collision. We define the slack variable

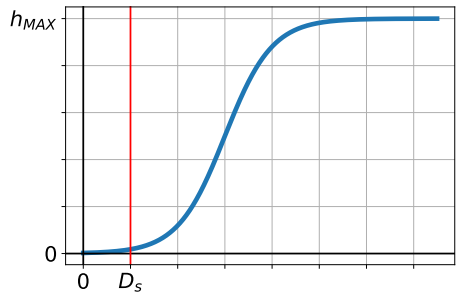

Fig. 3. Value of the slack variable ϵ.

$\epsilon(d_{ij}) \in \mathbb{R}_{\geq 0}$ as a sigmoid function of the distance d_{ij} calculated in (10). Consider the following sigmoid function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{16}$$

We define the slack variable as $\epsilon(d_{ij}) = \sigma(d_{ij} - 3D_s)$ where D_s is the safety distance. As we can see from Fig. 3, $\epsilon \to 0$ as d_{ij} decreases to D_s, while assuming a high value when the distance is greater and there is no danger of collision. In addition, the slope of the function contributes to balancing exploration and exploitation, assigning more weight to exploitation as the slope increases. The set of saturation values h_{MAX} is defined as the modulus of the lowest value that can be assumed by (12) according to the environment size, except for the safety constraint defined by the third row of $h({}^i\boldsymbol{p}_j)$, which is always kept as a hard constraint in order to ensure collision avoidance, thus $\epsilon_3 = 0$.

Inserting slack variables into the optimization problem (11), it becomes:

$$\begin{aligned} \boldsymbol{u}(\chi) = \arg\min_{\boldsymbol{u} \in \mathbb{R}^m} \ & \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{u}^*\|^2 \\ \text{s.t. } & \dot{h}(\chi, \boldsymbol{u}) + \epsilon \geq -\alpha(h(\chi)). \end{aligned} \tag{17}$$

Thus, the integration of ϵ has the effect of softening constraints related to robot j, therefore i can move towards the goal of the global mission when d_{ij} is safe $\forall j \in \mathcal{N}_i$. Otherwise, when the j-th ellipse indicates a possibly dangerous position, the j-th set of constraints becomes hard and forces the robot back into the safe set thanks to Theorem 1, in order to gain information on j-th position and increase the accuracy of the particle filter.

C. Hierarchical Optimization

One possible drawback of stacking several inequality constraints into the optimization problem (17) could be infeasibility. As a matter of fact, especially in case of large teams, robot i could be in a condition where the distance to more than one ellipses is critical, thus having several hard constraints in the optimization problem. To solve this issue, we assign a priority to each set of constraints in the same way as we sorted neighbors for the particles budget re-assignment in Section IV-B, i.e. based on the distance d_{ij} to the j-th ellipse calculated in (10). The adopted approach to hierarchically solve the optimization problem is the one proposed in [20], whose goal is to hierarchically minimize the violation $\|\kappa_j\|$ of the j-th level of constraints. More in details, a cascade of QP problems is performed, starting from minimizing $\|\kappa_1\|$ which gives an optimal value κ_1^*, then proceeding in

minimizing the violation of the following levels of constraints:

$$\begin{aligned}\kappa_j = \arg\min_{\chi,\boldsymbol{u}} \ & \|\kappa_j\|^2 \\ \text{s.t. } & \dot{h}(\chi,\boldsymbol{u}) + \kappa_1^* \quad \geq -\alpha(h(\chi)) \\ & \vdots \\ & \dot{h}(\chi,\boldsymbol{u}) + \kappa_{j-1}^* \quad \geq -\alpha(h(\chi)) \\ & \dot{h}(\chi,\boldsymbol{u}) + \kappa_j \quad \geq -\alpha(h(\chi)) \end{aligned} \tag{18}$$

This formulation ensures that, for each new level of constraints, the new value κ_j will not affect the prior constraint levels and therefore ensures a strict hierarchy.

The calculated values of minimized violations $\kappa = [\kappa_1, \ldots, k_{N-1}]^T$ is then inserted into (17) as a further set of slack variables. In conclusion, the main optimization problem becomes:

$$\begin{aligned}\boldsymbol{u}(\chi) = \arg\min_{\boldsymbol{u}\in\mathbb{R}^m} \ & \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{u}^*\|^2 + \frac{1}{2}\|\kappa\|^2 \\ \text{s.t. } & \dot{h}(\chi,\boldsymbol{u}) + \epsilon + \kappa \geq -\alpha(h(\chi)). \end{aligned} \tag{19}$$

The integration of κ into the optimization problem affects the robot's behaviour enforcing it to hierarchically search for undetected neighbors, aiming at first detecting neighbors who are expected to be in more dangerous positions.

VII. Experimental Evaluation

In this section, we present an application of our solution in the widely studied use case of coverage control. Then, we apply our methodology to coverage control of a team of aerial robots in a virtual environment and compare our results with those obtained employing robots equipped with isotropic sensors and ideal communication and we analyze how the number of robots in the team affects the performances of our solution and check its capability of handling large teams. Finally, we test the effectiveness of our solution when the team is instructed to follow intersecting trajectories. Robots are considered to be equipped with two kinds of sensors. The first one is an anisotropic sensor and is exploited to detect other robots. Its limited field of view corresponds to the aforementioned angular sector F_i. The second one, instead, is an isotropic sensor exploited for retrieving information from the environment. Its sensing area is considered to be a circle of radius equal to $R_s \in \mathbb{R}_{>0}$ centered in robot i's position. A graphical representation is provided in Fig. 4 for an aerial robot.

A. I Case Study: Coverage Control

The probabilistic approach proposed in this letter can be employed along a wide range of decentralized control algorithms. As a case study we investigate how the presented solution can guarantee the safe execution of the coverage task. Briefly, the coverage control exploits Lloyd's algorithm to reach the system configuration that maximizes the covered surface [21]. The Voronoi partitioning algorithm is used to divide the overall area of the environment into cells. In our work, we make use of estimated positions to locally build a rough approximation of the real Voronoi partitioning of the environment based on

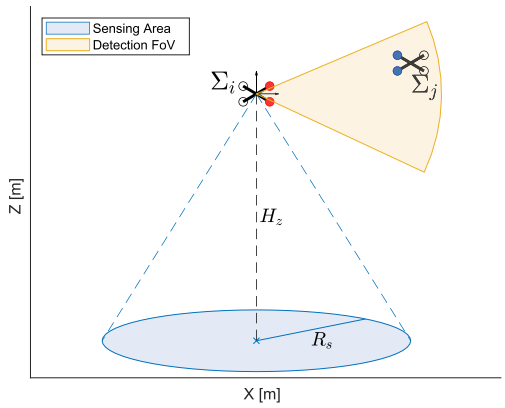

Fig. 4. Setup for experimental evaluation.

information from particle filters. The desired control action $\boldsymbol{u}^*$ is calculated with the coverage control algorithm as described in [21] with the only difference of the angular velocity calculated to face the centroid of its Voronoi cell, in order to move the robot towards the centroid while rotating along the z-axis until facing the target location. Finally, the desired velocity is inserted into the optimization problem (19). It is interesting to note that self-localization is not strictly necessary to control the team, since all the data needed to calculate the control input for the i-th robot is expressed relative to Σ_i.

B. Comparison With Ideal Scenario

Tests were performed in virtual environments. We employed a team of 6 aerial robots in the RotorS Gazebo simulator framework [22], controlled exploiting the ROS middleware. Drones were considered to fly at a constant altitude $H_z = 3.5$ m sensing a circular area with radius $R_s = 6$ m on the ground thanks to a camera facing downwards with a $120°$ field of view amplitude. Their maximum linear velocity is limited to 1.0 m/s, thus their roll and pitch angles can be neglected. Detection and relative localization of neighbors was emulated communicating to the i-th drone the relative position of j only when $j \in F_i$, with F_i defined as the angular sector of radius $R_d = 10$ m and amplitude $\beta = 120°$. Measurement uncertainty was modeled as a diagonal matrix $\Sigma_{MEAS} = diag(0.1, 0.1, 0.5)$, indicating a uniform and accurate measure of the position, but a lower accuracy on the orientation. A safety distance $D_s = 2.0$ m was defined for the safety constraints of the CBFs. We assumed every robot knows the starting position of each one of its neighbors. The team was tasked with a coverage operation in an obstacle-free large-scale environment of 40×40 m^2. A probability function $\phi(q)$ was defined as a Gaussian distribution with mean point $q_m \in \mathbb{R}^2$, highlighting the higher importance of areas near q_m. The control input for robot j needed in the prediction step (2) of the particle filter was estimated by robot i as a proportional law driving j towards q_m:

$${}^i\boldsymbol{u}_j^{est} = -K_p\left({}^ip_j - {}^iq_m\right) \tag{20}$$

The chosen control input roughly represents the expected behaviour of robot j, which, according to the coverage control strategy, should move towards the area of interest whose mean

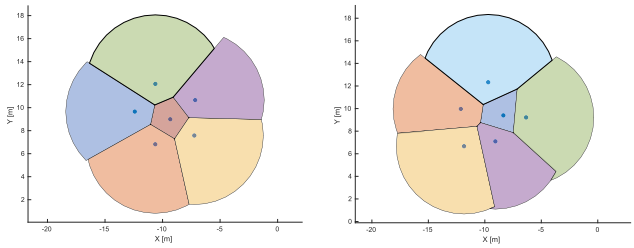

Fig. 5. Comparison between final configuration reached when performing coverage control with isotropic (left) and anisotropic (right) sensors adopting the proposed solution.

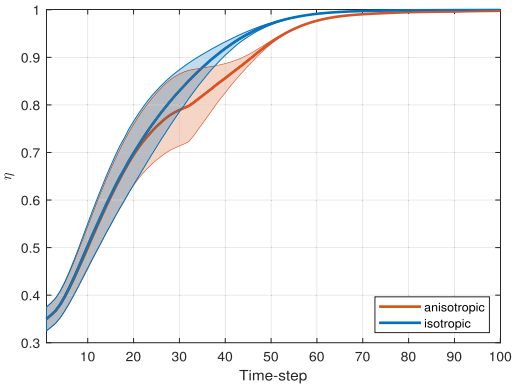

Fig. 6. Comparison in coverage performance.

point is ${}^i q_m$. Since ${}^i \boldsymbol{u}_j^{est}$ will not be accurate, the uncertainty on prediction will be modeled accordingly in order to exploit probabilistic analysis to avoid collisions. The aim of this first set of experiments was to compare performances in terms of covered surface and time required to accomplish the mission with respect to the ideal scenario of robots equipped with isotropic sensors.

The effectiveness was evaluated running 5 sets of simulations comparing performances of our solution with the ideal scenario of a coverage operation employing isotropic sensors. In particular, for each run, a random starting position was generated for each drone, then the mission was performed a first time with $\beta = 120°$ and afterwards with $\beta = 360°$. In both cases, log data was retrieved about the time required to reach the final configuration and area coverage efficiency $\eta \in [0, 1]$ calculated as:

$$\eta = \int_A \phi(q)\, dq \tag{21}$$

where $A \subset \mathbb{R}^2$ is the total surface covered by the team as represented in Fig. 5.

First of all, results demonstrated that the robotic team successfully accomplished the task without any collision being reported. Furthermore, results shown in Fig. 6 show that in both cases the team reaches a configuration where $\eta = 1$, i.e. the area of most interest according to $\phi(q)$ is completely covered. In addition, the time required to reach a value of $\eta = 0.99$ was evaluated, in order to have quantitative data indicating how much slower the fulfillment of the task was. Results show an average increase of 18.6% in task duration when adopting our strategy with anisotropic sensors in comparison with the ideal case of isotropic sensors.

From the obtained results we can state that our methodology can be successfully employed to overcome limitations on communication among the agents and their limited sensory capability, with the only drawback of a small increase in the duration of the mission due to the necessity of each robot to possibly stop and search for information.

TABLE I
CPU TIME VS NUMBER OF ROBOTS

Number of Robots	AVG CPU Time	Standard Deviation
6	138.8 ms	15.83
9	126.0 ms	16.26
12	125.7 ms	14.18

C. Analysis of Performances With Large Teams

This further set of tests focused on the analysis of the computational effort required to each agent depending on the total number of robots employed in the mission. The aim of this investigation is evaluating the scalability of the proposed solution, and check if the fixed total budget of particles is capable of making the algorithm independent from the number of robots. Teams with 6, 9 and 12 robots were employed, and the time required by the single agents for computation was measured. The simulation setup and robot's parameters were the same as the previous set of experiments, in particular F_i has been defined as an angular sector with amplitude $\beta = 120°$ and radius $R_d = 10$ m, a reasonable value for considering a precise neighbors localization (the authors in [13] define a maximum detection range of 15 m). Performances were evaluated meaning the average time required by every robot along the entire duration of the operation and the standard deviation among these values. Results are reported in Table I. The analysis of these results focuses on the relative values of average CPU time more than the absolute ones. As a matter of fact, the simulation can not reflect the real computation time since the detection and relative localization of neighbors was only emulated, in addition to hardware differences. For this reason, the analysis has been performed over a comparison of data related to teams with different number of agents, highlighting no significant differences in the average computation time required, even when numerous teams were employed. From these results, we can prove that the computational complexity of the particle filtering algorithm does not depend on the number of robots thanks to the fixed budget of particles to be continuously re-allocated, while the increasing number of cascaded QP problems (18) implies a negligible increase in the total computation time. In conclusion, the analysis suggests the scalability of the control strategy and indicates that it is nearly independent from the number of robots in the team.

D. II Case Study: Intersecting Trajectories

Finally, we tested the behaviour of a team of 4 UAVs when instructed to reach a desired configuration, which could lead to

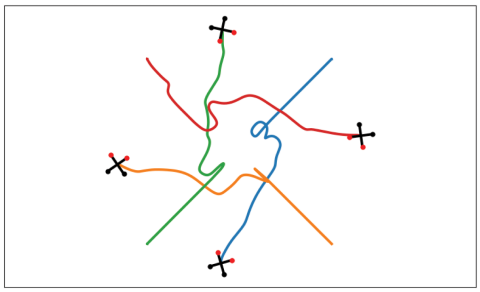

Fig. 7. Intersecting trajectories for 4 UAVs.

intersecting trajectories being followed by each robot. The aim of this last set of experiments was to check whether the agents were able to avoid collisions even without any information about the target position of their teammates. The simulation setup was the same as previous tests, with the only difference on the estimated control input for the prediction step (2). Since the behaviour of robot j is unknown, we considered the dangerous case of j moving towards i at half the maximum speed:

$$ {}^{i}\boldsymbol{u}_j^{est} = -\frac{v_{MAX}}{2}\frac{{}^{i}p_j}{\|{}^{i}p_j\|} \tag{22} $$

The desired control law ${}^{i}\boldsymbol{u}_i^*$ is again calculated as a proportional law towards the target. As we can see from Fig. 7, each UAV is capable of deviating from $\boldsymbol{u}^*$ in order to avoid collisions, then proceed to the target location occasionally searching for neighbors. The successful behaviour of the team confirmed the effectiveness of our solution even in challenging scenarios with potential collisions.

VIII. Conclusion

This letter presents a distributed approach for controlling a multi-robot system in communication-denied scenarios, particularly those with robots having a limited camera field of view for neighbor position retrieval. The key contribution is an optimization-based control ensuring collision avoidance while accomplishing a global task. The exploration-exploitation dilemma is addressed, requiring robots to balance information seeking about neighbors' positions with achieving the mission's global goal. Each robot predicts neighbor positions using a particle filter and defines confidence ellipses to address uncertainty. Control Barrier Functions constrain robot motion to keep a target in its field of view, and confidence ellipse distances are used to assess the opportunity to soften constraints. Simulations demonstrate successful collision-free task completion, and scalability is confirmed with varying robot numbers. Future work will focus on eliminating the need for IDs, evaluating overall area probabilities for robot presence, implementing an error model for neighbor localization, and addressing obstacles in the environment.

References

[1] C. Ju, J. Kim, J. Seol, and H. I. Son, "A review on multirobot systems in agriculture," *Comput. Electron. Agriculture*, vol. 202, 2022, Art. no. 107336.

[2] J. P. Queralta et al., "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191617–191643, 2020.

[3] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: Taxonomy and survey," *Auton. Robots*, vol. 40, no. 4, pp. 729–760, 2016.

[4] G. Li, R. Ge, and G. Loianno, "Cooperative transportation of cable suspended payloads with MAVs using monocular vision and inertial sensing," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5316–5323, Jul. 2021.

[5] M. Catellani, F. Pratissoli, F. Bertoncelli, and L. Sabattini, "Coverage control for exploration of unknown non-convex environments with limited range multi-robot systems," in *Proc. 16th Int. Symp. Distrib. Auton. Robotic Syst.*, 2022.

[6] C. Zhang and N. Noguchi, "Development of a multi-robot tractor system for agriculture field work," *Comput. Electron. Agriculture*, vol. 142, pp. 79–90, 2017.

[7] M. Iida et al., "Multi-combine robot system for rice harvesting operation," in *Proc. ASABE Annu. Int. Meeting. Amer. Soc. Agricultural Biol. Engineers*, 2017, p. 1.

[8] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed dynamic coverage and avoidance control under anisotropic sensing," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 4, pp. 850–862, Dec. 2017.

[9] E. Klavins, "Communication complexity of multi-robot systems," in *Algorithmic Foundations of Robotics V*, Berlin, Germany: Springer, 2004, pp. 275–291.

[10] M. B. Silic, Z. Song, and K. Mohseni, "Anisotropic flocking control of distributed multi-agent systems using fluid abstraction," in *Proc. AIAA Inf. Syst.-AIAA Infotech, Aerosp.*, 2018, Art. no. 2262.

[11] A. Wasik, P. U. Lima, and A. Martinoli, "A robust localization system for multi-robot formations based on an extension of a gaussian mixture probability hypothesis density filter," *Auton. Robots*, vol. 44, no. 3, pp. 395–414, 2020.

[12] W. Luo, W. Sun, and A. Kapoor, "Multi-robot collision avoidance under uncertainty with probabilistic safety barrier certificates," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 372–383.

[13] J. Horyna, V. Walter, and M. Saska, "UVDAR-COM: Uv-based relative localization of UAVs with integrated optical communication," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, 2022, pp. 1302–1308.

[14] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Norwood, MA, USA: Artech House, 2003.

[15] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 174–188, Feb. 2002.

[16] K. Pearson, "X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *London, Edinburgh, Dublin Philos. Mag. J. Sci.*, vol. 50, no. 302, pp. 157–175, Jul. 1900.

[17] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[18] P. G. Drazin and P. D. Drazin, *Nonlinear Systems*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

[19] F. Bertoncelli, V. Radhakrishnan, M. Catellani, G. Loianno, and L. Sabattini, "Directed graph topology preservation in multi-robot systems with limited field of view using control barrier functions," submitted to IEEEAccess.

[20] O. Kanoun, F. Lamiraux, and P.-B. Wieber, "Kinematic control of redundant manipulators: Generalizing the task-priority framework to inequality task," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 785–792, Aug. 2011.

[21] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Automat.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[22] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham, Switzerland: Springer Int. Publishing, 2016, doi: 10.1007/978-3-319-26054-9_23.

Open Access funding provided by 'Universit? degli Studi di Modena e Reggio Emilia' within the CRUI CARE Agreement