DISTRIBUTED COVERAGE BY A SWARM OF AUTONOMOUS QUADROTORS

NIKOLAOS BOUSIAS

Department of Electrical & Systems Engineering
General Robotics Automation Sensing & Perception Laboratory
University of Pennsylvania
Levine Hall 403 - 3330 Walnut Street
Philadelphia PA, 19104, USA

ABSTRACT. This work proposes a distributed, collaborative control framework for an autonomous swarm of quadrotors, tasked with the surveillance of a region of interest without obstacles. Each Mobile Aerial Agent (MAA) is equipped with a Pan-Tilt-Zoom camera of conical FOV. The suggested framework attempts to monotonically optimize the collective visual coverage, both in terms of total area and quality, by utilizing a Voronoi-free tessellation strategy while also accounting for the localization uncertainty induced in each node's GPS measurements. For this framework to accommodate heterogeneous teams of robots with a variety of dynamics, a hierarchical control structure is designed which incorporates distributed path planning with reduced order dynamics and a system specific controller aware of the dynamics of the node. Simulated results will be presented to verify the efficiency of the proposed scheme.

1. INTRODUCTION

THE distributed coverage problem is an application of multi-robot systems where robots are tasked with collectively sensing a subspace of their environment that causes interest. Each agent is outfitted with task-appropriate sensors and is capable of in-group inter-agent communication, thus forming a network of sensors collecting intelligence and exploring an environment. Such ad-hoc networks are robust and adaptive to single node failures, incorporation of additional resources or partial collapse of communications, and can accommodate heterogeneous teams. The heterogeneity of the swarm refers to the type of robots utilized (the deployed swarm may consist of mobile ground, underwater, or more commonly aerial robots) or the features of the on-board sensors and communication equipment. A number of uses have already been suggested in fields ranging from environmental monitoring (e.g. forest fire detection, population management of fauna/flora) and surveillance (e.g. border patrol, monitoring parades, urban security) to inspection of infrastructure, agricultural tasks (e.g. harvesting fruits, managing watering of plantation), or even de-mining, demonstrating the versatility and value of this problem, especially in dirty, dull or dangerous jobs where human involvement is prohibited.

Coverage problems for mobile robot teams can be broadly categorized as static or sweep coverage. In static or blanket coverage [1, 2] the objective of the mobile agents is to converge to a static configuration through which some performance criterion is optimized. In dynamic or sweep coverage problems [3, 4] the performance criterion is time–varying, resulting in the agents moving constantly. Other ways to categorize coverage problems are based on the properties of the region of interest [5, 6], of the dynamic model of the mobile agents [7, 8], or on the type of their on-board sensors [9, 10]. The most common approach to coverage

E-mail address: nbousias@seas.upenn.edu.

problems is geometric optimization [11] with other proposed approaches being event–triggered control [12], game theory [13], annealing algorithms [14], model predictive control [15], reinforcement learning [16], and potential fields [17].

To account for the uncertainty in measurements of most positioning systems, several proposed solutions have emerged, including probabilistic methods [18], safe trajectory planning [19], or the use of Voronoi–like partitioning schemes [20, 21]. In this article the positioning uncertainty model is similar to the one used in [20] but the approach followed differs. Instead of employing a Voronoi–like space partitioning, the positioning uncertainty is incorporated in the agents' sensing patterns and the sensed space is partitioned using a Voronoi–free technique similar to [22]. Furthermore, rather than using downwards facing cameras, this article extends the case to a complete PTZ-camera. A similar approach for visual area coverage has also appeared in [23] and an algorithm for information exchange has also been developed. In this work, however, the cameras are not allowed to zoom and the M.A.A.s' localization is precise. Moreover, regions covered by multiple agents contribute more to the objective, thus favoring overlapping between the agents' sensed regions. Previous works have examined downwards facing cameras [22, 24] and although positioning uncertainty has been successfully incorporated in these control schemes [25], it was achieved using a Voronoi–like partitioning.This is not tractable to generalize in the case of PTZ-cameras. PTZ-camera networks have been examined using Voronoi–like diagrams in [26] for stationary cameras and in Voronoi-free tesselation strategies as in [27, 28].

The aforementioned literature does not account for the dynamics of the nodes. A common practice is to assume simple low order dynamics (e.g. single-integrator) as it significantly simplifies the analysis and allows for heterogeneity in the node implementation. A team tasked with monitoring a region of interest may comprise of a variety of UAVs, depending on the availability and required dexterity. In the current work, we incorporate the control laws designed in [29] with plant-specific trajectory tracking controllers assuming that the team comprises only of quadrotors. Trajectory tracking controllers for quadrotors have been proposed in [30, 31].

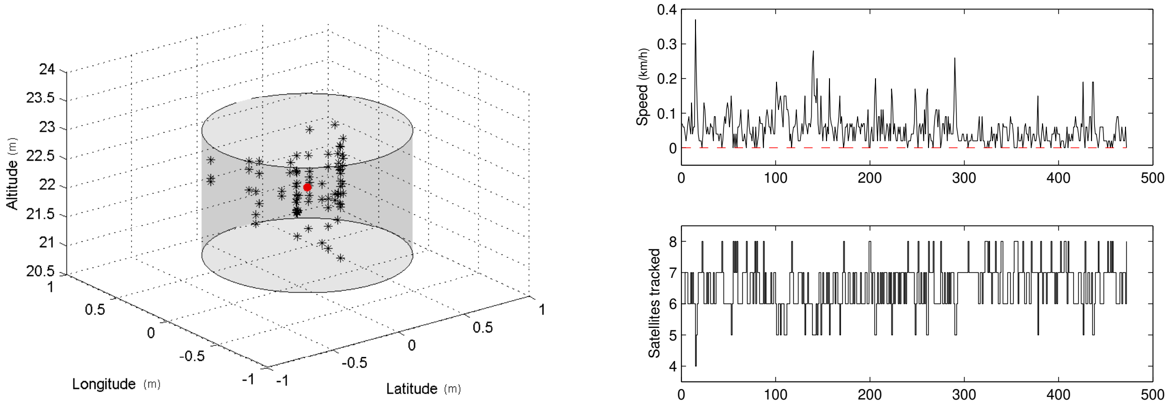
2. Problem Statement

A compact convex region of interest denoted as $\Omega \in \mathbb{R}^2$ is to be placed under surveillance by a swarm of n quadrotors equipped with Pan-Tilt-Zoom cameras. The problem is divided in two main tasks: designing the distributed control laws assuming an swarm of abstract MMAs (Mobile Aerial Agents) and implementing it using a swarm of quadrotors. The distributed collaborative framework, based on an approximation of the MMAs by point masses, is tasked will generating the nominal 3D path and PTZ angles that would monotonically optimize the collective visual surveillance over Ω, both in terms of total area covered and the quality of coverage. The collaborative control scheme formulated similarly to [27]. Introducing a swarm of quadrotors means designing the appropriate control laws to enable the physically-specified now system track the nominal trajectory created by the aforementioned laws.

3. Collaborative Control

3.1. **MMA & PTZ–camera model.** Each MAA is located at $\mathbf{X_i} = [x_i, y_i, z_i]^T, i \in I_n = \{1, \ldots, n\}$ and is restricted within a predefined altitude range. We define the vector $q_i = [x_i, y_i]^T \in \Omega$ denoting the projection of each MAA on Ω. Each MAA bears an on-board PTZ-capable visual sensor of conical FOV. We denote the sensor's pan and tilt angles h_i and θ_i respectively, while its zoom level $2\delta_i$ is represented by the angle of the cone of vision.

Given the conical field of view of the sensors, its intersection with the plane will be a conic section which we call the sensing pattern which ammounts to the region of the plane an MAA is able to cover. We define



(A) GPS latitude, longitude and altitude measurements. (B) Speed and number of satellites tracked.

FIGURE 1. Typical GPS measurements for static global position.

the center of the sensing footprint $q_{c,i} \in \Omega$ and denote the semi-major and semi minor axis of the elliptic sensing pattern a_i and b_i respectively. Therefore, each MAA's sensing pattern is

$$C_i^s(X_i, h_i, \theta_i, \delta_i) = \mathbf{R}(\theta_i) C_i^b + q_{i,c}, \quad i \in I_n, \tag{1}$$

where $\mathbf{R}$ is the 2×2 rotation matrix, $||.||$ is the Euclidean metric and

$$\begin{aligned}
C_i^b &= \left\{ q \in \mathbb{R}^2 : \left\| \begin{bmatrix} \frac{1}{a_i} & 0 \\ 0 & \frac{1}{b_i} \end{bmatrix} q \right\| \leq 1 \right\}, \\
a_i &= \frac{z_i}{2} \left[\tan(h_i + \delta_i) - \tan(h_i - \delta_i) \right], \\
b_i &= z_i \tan(\delta_i) \sqrt{1 + \left[\frac{\tan(h_i + \delta_i) + \tan(h_i - \delta_i)}{2} \right]^2}, \\
q_{i,c} &= q_i + w_i \frac{z_i}{2} \left[\tan(h_i + \delta_i) + \tan(h_i - \delta_i) \right] \\
w_i &= \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \end{bmatrix}^T .
\end{aligned}$$

The pan angle θ_i only affects the sensing pattern's orientation, while the tilt angle h_i affects the eccentricity of the elliptical sensing pattern.

A single integrator serves as the dynamic model, offering scalability and abstraction regarding the physical substance of the nodes of the swarm. To the extend of the collaborative strategy, the MAAs are approximated by point masses able of translational movement in $\mathbb{R}^3$. The visual sensors' PTZ rotations are controlled by on-board servos, thus their states are decoupled from those of the MAA. Thus, the kinematics of each MAA $i \in I_n$ is

$$\begin{aligned}
\dot{q}_i &= u_{i,q}, \quad q_i \in \Omega, \\
\dot{z}_i &= u_{i,z}, \quad z_i \in \left[z_i^{\min}, z_i^{\max} \right], \\
\dot{\theta}_i &= u_{i,\theta}, \quad \theta_i \in \mathbb{R}, \\
\dot{h}_i &= u_{i,h}, \quad h_i \in \left(-h_i^{\max}, h_i^{\max} \right), \\
\dot{\delta}_i &= u_{i,\delta}, \quad \delta_i \in \left[\delta_i^{\min}, \delta_i^{\max} \right],
\end{aligned} \tag{2}$$

where $u_{i,q} \in \mathbb{R}^2$, $u_{i,z}$, $u_{i,\theta}$, $u_{i,h}$, $u_{i,\delta} \in \mathbb{R}$.

3.2. **MAA positional uncertainty.** Inherent inaccuracy in absolute positioning (GPS) sensors poses a critical issue for such systems since the control inputs are associated with the assumed position. As illustrated in Figure 1(a), the imprecision in altitude, longitude and latitude measurements can be considered bounded irrespectively of the number of satellites being tracked. A virtual cylinder that contains all possible positions of the agent can be used to characterize the localization uncertainty. In consideration of this, in the proposed scheme, each MAA is subject to a distinct 3D localization uncertainty. Each node of the swarm may be attributed different uncertainty boundaries. Given an upper bound $r_i = \left[r_i^q, r_i^z\right]^T$ for the localization uncertainty of each MAA, its location X_i may reside anywhere within a cylinder called the localization uncertainty region

$$C_i^u(\mathbf{X_i}, \mathbf{r_i}) = \left\{ \mathbf{X_i} \in \Omega \times \left[z_i^{min}, z_i^{max}\right] : \begin{bmatrix} \|q - q_i\| \\ \|z - z_i\| \end{bmatrix} \leq \begin{bmatrix} r_i^q \\ r_i^z \end{bmatrix} \right\}. \tag{3}$$

Furthermore, given the positioning uncertainty of each MAA, we also define the guaranteed sensed region $C_i^{gs} \subset \mathbb{R}^2$ as the region the MAA is guaranteed to cover for all its possible positions within C_i^u. The guaranteed sensed region is then defined as

$$C_i^{gs}(X_i, h_i, \theta_i, \delta_i, r_i) \triangleq \left\{ \bigcap_{X_i \in C_i^u} C_i^s \right\} = \left\{ \bigcap_{z \in \left[z_i - r_i^z, z_i + r_i^z\right]} C_i^p(z) \right\},$$

where

$$C_i^p(z) = \mathbf{R}(\theta_i) C_i^{b_{gs}} + q_{i,c},$$
$$C_i^{b_{gs}} = \left\{ q \in \Omega \colon \left\| \begin{bmatrix} \frac{1}{a_i - r_i^q} & 0 \\ 0 & \frac{1}{b_i - r_i^q} \end{bmatrix} q \right\| \leq 1 \right\}.$$

Figure 2 illustrates the visual coverage concept. The agent's guaranteed sensing pattern is shown in grey for both $h_i = 0$ and $h_i \in \left(0, \frac{\pi}{2} - \delta_i\right)$.

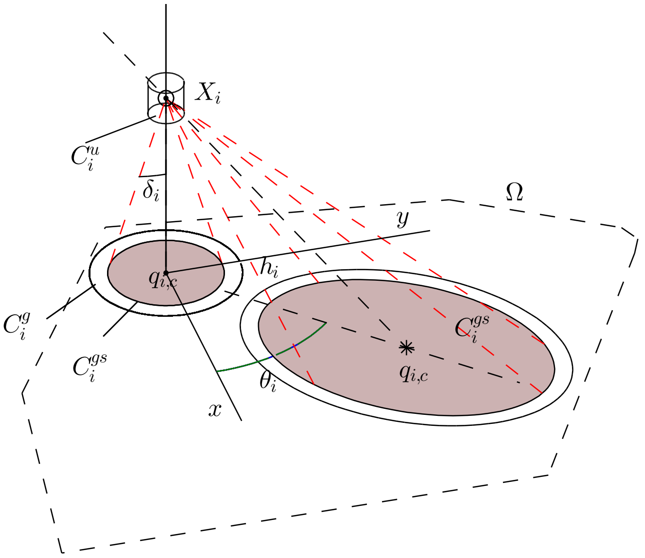


FIGURE 2. Visual coverage concept.

3.3. **Area coverage objective.** Due to the nature of visual sensors, objects further away from the sensor appear with lower quality than ones near the sensor. We model the coverage quality using a quality function

$f_i(z_i,h_i,\delta_i)\colon \left[z_i^{\min},+\infty\right] \to [0,1]$, with 0 and 1 corresponding to the lowest and the highest possible quality respectively. Each point $q \in \Omega$ is assigned an importance factor through a space density function $\phi\colon \Omega \to \mathbb{R}_+$ which expresses the a priori information regarding the importance of certain regions of Ω. We define the following joint coverage-quality objective

$$\mathscr{H} \overset{\triangle}{=} \int_{\Omega} \max_{i\in I_n} f_i(z_i,h_i,\delta_i)\ \phi\left(q\right) \mathrm{d}q. \tag{4}$$

This function accounts for both the area covered by the agents and the coverage quality over that area, while also taking into account the importance of points as encoded by $\phi(q)$. The goal of the MAA swarm is to maximize this objective.

3.4. **Area Partitioning Strategy.** The most common choice of partitioning scheme for area coverage problems is the Voronoi–like diagram [20, 25, 32]. However, in this work a partitioning of the sensed space is utilized similarly to [24]. This partitioning scheme assigns a region of responsibility to each agent based on the guaranteed sensed regions C_i^{gs} and the coverage quality over them. Each MAA is assigned a cell W_i as follows

$$W_i \overset{\triangle}{=} \left\{q \in \Omega\colon\ f_i > f_j,\ i \neq j\right\}, i \in I_n. \tag{5}$$

However the union of these cells does not comprise a complete tessellation of the total guaranteed sensed region $\bigcup\limits_{i\in I_n} C_i^{gs}$. This is due to the fact that regions sensed by multiple agents with the same coverage quality are left unassigned. These common regions still contribute towards the objective $\mathscr{H}$, so they must be taken into account. Let L be the number of agents with the same coverage quality f^l and overlapping guaranteed sensed regions and the set comprized by them be defined as

$$\mathscr{I}_l = \left\{i,j \in I_n, i \neq j\colon C_i^{gs} \cap C_j^{gs} \neq \emptyset\ \wedge\ f_i = f_j = f^l\right\}, l \in I_L$$

where $I_L = \{1,...,L\}$. Then, the common regions are

$$W_c^l = \left\{\exists i,j \in \mathscr{I}_l, i \neq j : q \in C_i^{gs} \cap C_j^{gs}\right\}, \quad l \in I_L. \tag{6}$$

Since only the guaranteed sensed regions $\bigcup\limits_{i\in I_n} C_i^{gs}$ are partitioned, a portion of Ω called neutral region is discarded

$$\mathscr{O} = \Omega \setminus \left\{\bigcup_{i\in I_n} W_i \cup \bigcup_{i\in I_l} W_c^l\right\} = \Omega \setminus \bigcup_{i\in I_n} C_i^{gs}.$$

By utilizing the partitioning strategy (5), (6), the coverage–quality objective (4) can be written as

$$\mathscr{H} = \sum_{i\in I_n} \int_{W_i} f_i\ \phi(q)\ \mathrm{d}q + \sum_{l=1}^{L} \int_{W_c^l} f^l\ \phi(q)\ \mathrm{d}q \tag{7}$$

3.5. **Collaborative Control Laws.**

Theorem 1. *For a swarm of MAAs with the aforementioned kinematics, sensing performance and localization uncertainty, the following control laws guarantee monotonic increase of the coverage–quality objective to a locally optimal configuration along the MAAs' trajectories.*

$$u_{i,q} = K_{i,q}\left[\int\limits_{\partial W_i \cap \partial \mathscr{O}} u_i^i\, n_i\ f_i\ \phi(q)\ \mathrm{d}q + \textstyle\sum_{\substack{j\in I_n\\ j\neq i}} \int\limits_{\partial W_i \cap \partial W_j} u_i^i\, n_i\left(f_i - f_j\right)\ \phi(q)\ \mathrm{d}q\right],$$

$$u_{i,z} = K_{i,z}\left[\int\limits_{\partial W_i \cap \partial \mathscr{O}} v_i^i\, n_i\ f_i\ \phi(q)\ \mathrm{d}q + \int\limits_{W_i} \frac{\partial f_i}{\partial z_i}\phi(q)\ \mathrm{d}q + \textstyle\sum_{\substack{j\in I_n\\ j\neq i}} \int\limits_{\partial W_i \cap \partial W_j} v_i^i\, n_i\left(f_i - f_j\right)\ \phi(q)\ \mathrm{d}q\right],$$

Algorithm 1: PARTITIONING ALGORITHM.

```
**Input:** $C_i^{gs}, S_i, S_j, C_j^{gs}\ \forall j \in N_i$
**Output:** $W_i$
$W_i \leftarrow C_i^{gs}$
**foreach** $j \in N_i$ **do**
    **if** $C_i^{gs} \cap C_j^{gs} \neq \emptyset$ **then**
        **if** $f(z_i, h_i, \delta_i) \leq f(z_j, h_j, \delta_j)$ **then**
            $W_i \leftarrow W_i \setminus \left\{C_i^{gs} \cap C_j^{gs}\right\}$
        **end**
    **end**
**end**
$W_i \leftarrow W_i \cap \Omega$
```

$$u_{i,\theta} = K_{i,\theta}\left[\int\limits_{\partial W_i \cap \partial \mathscr{O}} \tau_i^i\, n_i\, f_i\, \phi(q)\, \mathrm{d}q + \textstyle\sum_{\substack{j\in I_n\\ j\neq i}} \int\limits_{\partial W_i \cap \partial W_j} \tau_i^i\, n_i\,(f_i - f_j)\ \phi(q)\, \mathrm{d}q\right],$$

$$u_{i,h} = K_{i,h}\left[\int\limits_{\partial W_i \cap \partial \mathscr{O}} \sigma_i^i\, n_i\, f_i\, \phi(q)\, \mathrm{d}q + \int\limits_{W_i} \tfrac{\partial f_i}{\partial h_i}\ \phi(q)\, \mathrm{d}q + \textstyle\sum_{\substack{j\in I_n\\ j\neq i}} \int\limits_{\partial W_i \cap \partial W_j} \sigma_i^i\, n_i\,(f_i - f_j)\ \phi(q)\, \mathrm{d}q\right],$$

$$u_{i,\delta} = K_{i,\delta}\left[\int\limits_{\partial W_i \cap \partial \mathscr{O}} \mu_i^i\, n_i\, f_i\, \phi(q)\, \mathrm{d}q + \int\limits_{W_i} \tfrac{\partial f_i}{\partial \delta_i}\ \phi(q)\, \mathrm{d}q + \textstyle\sum_{\substack{j\in I_n\\ j\neq i}} \int\limits_{\partial W_i \cap \partial W_j} \mu_i^i\, n_i\,(f_i - f_j)\ \phi(q)\, \mathrm{d}q\right] \tag{8}$$

where $K_{i,q}, K_{i,z}, K_{i,\theta}, K_{i,h}, K_{i,\delta}$ are positive constants, n_i the outward pointing unit normal vector on W_i and $u_i^i, v_i^i, \tau_i^i, \sigma_i^i, \mu_i^i$ the Jacobian matrices

$$u_j^i \triangleq \frac{\partial q}{\partial q_i}, \quad v_j^i \triangleq \frac{\partial q}{\partial z_i}, \quad \tau_j^i \triangleq \frac{\partial q}{\partial \theta_i}, \quad \sigma_j^i \triangleq \frac{\partial q}{\partial h_i}, \mu_j^i \triangleq \frac{\partial q}{\partial \delta_i}, \quad q \in \partial W_j,\ i, j \in I_n.$$

Proof. In order to guarantee monotonic increase of $\mathscr{H}$, its time derivative is evaluated as

$$\frac{d\mathscr{H}}{dt} = \sum_{i\in I_n}\left[\frac{\partial \mathscr{H}}{\partial q_i}\dot{q}_i + \frac{\partial \mathscr{H}}{\partial z_i}\dot{z}_i + \frac{\partial \mathscr{H}}{\partial \theta_i}\dot{\theta}_i + \frac{\partial \mathscr{H}}{\partial h_i}\dot{h}_i + \frac{\partial \mathscr{H}}{\partial \delta_i}\dot{\delta}_i\right]$$

By selecting the following control inputs

$$u_{i,q} = K_{i,q}\frac{\partial \mathscr{H}}{\partial q_i},\ u_{i,z} = K_{i,z}\frac{\partial \mathscr{H}}{\partial z_i},\ u_{i,\theta} = K_{i,\theta}\frac{\partial \mathscr{H}}{\partial \theta_i}, u_{i,h} = K_{i,h}\frac{\partial \mathscr{H}}{\partial h_i},\ u_{i,\delta} = K_{i,\delta}\frac{\partial \mathscr{H}}{\partial \delta_i} \tag{9}$$

we guarantee, given the MAAs' dynamics and $K_{i,q}, K_{i,z}, K_{i,\theta}, K_{i,h}, K_{i,\delta} \geq 0$, that $\frac{\partial \mathscr{H}}{\partial t}$ is non-negative, thus, ensuring that the coverage–quality criterion increases in a monotonic manner.

The partial derivative $\frac{\partial \mathscr{H}}{\partial q_i}$ is

$$\frac{\partial \mathscr{H}}{\partial q_i} = \frac{\partial}{\partial q_i}\left\{\sum_{i\in I_n}\int\limits_{W_i} f_i\ \phi(q)\ \mathrm{d}q + \sum_{l=1}^{L}\int\limits_{W_c^l} f^l\ \phi(q)\ \mathrm{d}q\right\}.$$

By applying the Leibniz integral rule and since $\frac{\partial f_i(z_i,h_i,\delta_i)}{\partial q_i} = \frac{\partial f_j(z_j,h_j,\delta_j)}{\partial q_i} = 0$ the previous equation yields

$$\frac{\partial \mathscr{H}}{\partial q_i} = \int\limits_{\partial W_i} u_i^i\, n_i\, f_i\, \phi(q)\ \mathrm{d}q + \sum_{\substack{j\in I_n\\ j\neq i}}\ \int\limits_{\partial W_i \cap \partial W_j} u_j^i\, n_j\, f_j \phi(q)\ \mathrm{d}q \tag{10}$$

We use a boundary decomposition of ∂W_i into disjoint sets similarly to [24]

$$\partial W_i = \left\{ \{\partial W_i \cap \partial \Omega\} \cup \{\partial W_i \cap \partial \mathscr{O}\} \cap \left\{ \bigcup_{i \neq j} \partial W_i \cap \partial W_j \right\} \cap \left\{ \bigcup_{l=1}^{L} \partial W_i \cap \partial W_c^l \right\} \right\}$$

and assume a static region of interest Ω. Since $\partial W_i \cup \partial W_c^l$ are subsets of some sensed region boundary ∂C_j^{gs}, independent of the state of node i, at $q \in \left\{\partial \Omega \cap \partial W_i\right\} \cup \left\{\partial W_i \cup \partial W_c^l\right\}$, the Jacobian matrix is $u_i^i = \mathbf{0_{2\times 2}}$. In addition, due to the fact that the boundary $\partial W_i \cup \partial W_j$ is shared amongst both nodes i and j, it stands to reason that $u_j^i = u_i^j$ and $n_i = -n_j$ when they are evaluated over it. Therefore, (10) yields

$$\frac{\partial \mathscr{H}}{\partial q_i} = \int\limits_{\partial W_i \cap \partial \mathscr{O}} u_i^i \, n_i f_i \; \phi(q) \, \mathrm{d}q + \sum_{\substack{j \in I_n \\ j \neq i}} \int\limits_{\partial W_i \cap \partial W_j} u_i^i \, n_i \left[f_i - f_j\right] \; \phi(q) \, \mathrm{d}q.$$

Through a similar procedure and given that $\frac{\partial f_j}{\partial z_i} = \frac{\partial f_j}{\partial \theta_i} = \frac{\partial f_i}{\partial \theta_i} = \frac{\partial f_j}{\partial h_i} = \frac{\partial f_j}{\partial \delta_i} = 0$ we obtain the rest of the control laws $u_{i,z}, u_{i,\theta}, u_{i,h}$ and $u_{i,\delta}$.

□

Remark 1. *The control laws for the altitude, tilt and zoom angles include an extra term each, that is due to the dependence of the quality-coverage metric f_i from those variables, that is $\frac{\partial f_i}{\partial z_i}, \frac{\partial f_i}{\partial h_i}, \frac{\partial f_i}{\partial \delta_i} \neq 0$.*

Theorem 2. *For a swarm of n agents governed by the dynamics* (2) *and the control laws* (8)*, it stands that the system states $q_i, z_i, \theta_i, h_i, \delta_i$ $\forall i \in I_n$ ultimately converge to a final locally optimal configuration i.e*

$$\lim_{t\to\infty} \frac{\partial \mathscr{H}}{\partial q_i} = \lim_{t\to\infty} \frac{\partial \mathscr{H}}{\partial z_i} = \lim_{t\to\infty} \frac{\partial \mathscr{H}}{\partial \theta_i} = \lim_{t\to\infty} \frac{\partial \mathscr{H}}{\partial \delta_i} = \lim_{t\to\infty} \frac{\partial \mathscr{H}}{\partial h_i} = 0, \quad \forall i \in I_n$$

Proof. From the definition (4) and given that Ω is static, the maximum coverage objective a swarm may achieve is

$$M = arg_{q_i, z_i, \theta_i, h_i, \delta_i \forall i \in I_n} \max \mathscr{H} = \int_{\Omega} \phi\left(q\right) \mathrm{d}q$$

provided sufficient agents are deployed for the swarm to sense the entire region of interest with the maximum possible quality, that is

$$\left\{ \bigcup_{i \in I_n} W_i \,\cup\, \bigcup_{i \in I_l} W_c^l = \Omega \right\} \wedge \left\{ f_i(z_i, h_i, \delta_i) = 1 \; \forall i \in I_n \right\}.$$

Let us consider the Lyapunov-like function $\mathbb{V} = M + \alpha - \mathbb{H}$, where $\alpha > 0$. Since $M \geq \mathbb{H}$, then $\mathbb{V}$ is a positive, locally Lipschitz continuous, differentiable function with negative semi-definite time derivative along the trajectories of this autonomous system $\dot{\mathbb{V}} = -\dot{\mathbb{H}} \leq 0$. The evolution of all states of the system remains bounded. Assuming the dynamics of (2), should $\Omega \cap C_i^{gs} = \emptyset$ then $u_{i,q} = u_{i,z} = u_{i,\theta} = u_{i,h} = u_{i,\delta} = 0$ meaning that the robot is immobilized, thus $q_i, h_i, \delta_i, \theta_i$ are bounded. Given $\Omega \cap C_i^{gs} \neq \emptyset$, $\mathbb{V}$ is radially unbounded with respect to z_i, thus $\dot{\mathbb{V}} \leq 0$ guarantees boundedness. The equilibrium set $E = \{q_i \in \mathbb{R}^2, (z_i, \theta_i, h_i, \delta_i) \in \mathbb{R}, \forall i \in I_n : \dot{\mathbb{V}} = \dot{\mathbb{H}} = 0\}$ is in this case an invariant set since from (9) the robot ceases to move. Applying LaSalle's invariance principle, the trajectories of each robot asymptotically converge to E for $S^0 \equiv \{q_i^0, z_i^0, \theta_i^0, h_i^0, \delta_i^0\}$ such that $\bigcup_{i \in I_n} C_i^{gs}|_{S^0} \cap \Omega \neq \emptyset$. □

Theorem 3. *An equilibrium configuration for the aforementioned system $(q_i^*, z_i^*, \theta_i^*, h_i^*, \delta_i^*)$ $\forall i \in I_n$ defined by*

$$\left.\frac{\partial \mathscr{H}}{\partial q_i}\right|_{q_i = q_i^*} = \left.\frac{\partial \mathscr{H}}{\partial z_i}\right|_{z_i = z_i^*} = \left.\frac{\partial \mathscr{H}}{\partial \theta_i}\right|_{\theta_i = \theta_i^*} = \left.\frac{\partial \mathscr{H}}{\partial h_i}\right|_{h_i = h_i^*} = \left.\frac{\partial \mathscr{H}}{\partial \delta_i}\right|_{\delta_i = \delta_i^*} = 0$$

Algorithm 2: DISTRIBUTED CONTROL ALGORITHM.

```
Requirement: Robot $i$ has established communications with neighbours $N_i$ and has prior information
             of the geometry of $\Omega$ and the density function $\phi(q), \forall q \in \Omega$
Input: $S_i = \langle q_i, z_i, \theta_i, h_i, \delta_i \rangle$, $S_j = \langle q_j, z_j, \theta_j, h_j, \delta_j \rangle$, $\forall j \in N_i$, $\mathbb{K} \equiv [K_{i,q}, K_{i,z}, K_{i,\theta}, K_{i,h}, K_{i,\delta}]$,
       $r_i^q$, $r_i^z$, $z_i^{\max}$, $z_i^{\min}$, $\delta_i^{\min}$, $\delta_i^{\max}$, $h_i^{\max}$
Output: $u_{i,q}$, $u_{i,z}, u_{i,\theta}$, $u_{i,h}, u_{i,\delta}$
Initialization:
    $[u_{i,q}, u_{i,z}, u_{i,\theta}, u_{i,h}, u_{i,\delta}] \leftarrow \mathbf{0_{1\times 6}}$
    foreach $j \in N_i$ do
        receive $r_j^q$, $r_j^z$, $z_j^{\max}$, $z_j^{\min}$, $\delta_j^{\min}$, $\delta_j^{\max}$, $h_j^{\max}$
    end

while check_connectivity = True do
    estimate state $S_i$
    send $S_i$ to neighbours
    receive $S_j$, $\forall j \in N_i$
    calculate $a_i$, $b_i$, $q_{i,c}$, $f_i$, $C_i^{gs}$
    foreach $j \in N_i$ do
        calculate $a_j$, $b_j$, $q_{j,c}$, $f_j$, $C_j^{gs}$
    end
    $W_i \leftarrow partitioning\_algorithm(C_i^{gs}, C_j^{gs}, \Omega, f_i, f_j)$
    $[u_{i,q}, u_{i,z}, u_{i,\theta}, u_{i,h}, u_{i,\delta}] \leftarrow$
      $\mathbb{K}\left\{ integral(W_i, C_j^{gs}, f_i, f_j, \phi) + \begin{bmatrix} 0 & \frac{\partial f_i}{\partial z_i} & 0 & \frac{\partial f_i}{\partial h_i} & \frac{\partial f_i}{\partial \delta_i} \end{bmatrix} Area(W_i) \right\}$
end
```

is Lyapunov stable if and only if it is a local maxima of the objective function $\mathscr{H}$

Proof. See [33],[34]. □

3.6. **Resilience & Adaptability.** Most modern system designs focus on agility and intelligence implemented on a single robot. However, this strategy, efficiency and cost aside, offers limited resilience in the event of technical failure. If a single drone operating over a region of interest malfunctions, the entire surveillance operation collapses. Dropping the single-agent strategy in favor of a multi-agent approach guarantees than in case of failure -even multiple ones- at least some functionality/coverage is retained as the network self-reconfigures. The network would only collapse if all agents fail.

Other than resilience to component failures, adaptability in changes in the synthesis of the network may also imply scalability. A swarm consisting of $n+1$ agents will always provide superior coverage to that of a swarm of n agents (unless for the n case, the number of agents suffice to cover the entire area with all agents at minimum altitude, minimum tilt angle and maximum zoom so extra agents would have empty W_j cells). Assuming that the region of interest requires enhanced coverage/detail and the maximum possible coverage is not yet achieved, extra agents can be deployed in the operating swarm without the need to ground the up and running fleet. The control strategy, also, considers refuelling in the midst of a mission. This observation means that the control design allows dynamic resources allocation for the visual surveillance of an area, thus, the system is flexible and robust to failures.

3.7. **Computational complexity.** To determine the computational complexity of the distributed control, let us assume that all visual pattern-related sets comprise of a total of m points. Let us, also, assume that all agents are neighbours with each other, that is $N_i \equiv I_n \setminus i$. Then, regarding the partitioning strategy of Algorithm 1

$$T_{partition}(n,m) \in \left[\sum_{k=1}^{m} O(1) + \sum_{j=1}^{n-1}\sum_{k=1}^{m} O(1) + \sum_{k=1}^{m} O(1)\right] \tag{11}$$

so $T_{partition}(n,m) = O(n \cdot m)$ as each operation between sets is $O(m)$. The overall system's control of Algorithm 2 requires

$$T_c^i(n,m) \in \left[\sum_{j=1}^{n-1} O(1) + O(1) + \sum_{k=1}^{m} O(1) + \sum_{j=1}^{n-1}\left(O(1) + \sum_{k=1}^{m} O(1)\right) + \sum_{j=1}^{n-1}\sum_{k=1}^{m} O(1) + O(1)\right] = O(n \cdot m) \tag{12}$$

When calculating the control laws for all robots on a centralized processor (as implemented for the simulations subsection), the time complexity becomes $O(n^2 \cdot m)$.

Remark 2. *The complexity estimations stress the trade-off between time complexity and number of points used for the discrete pattern approximation. The more precision required, the greater the time interval between each iteration, thus, the collaborative control no longer guarantees monotonicity.*

Remark 3. *Despite not being equally accurate, utilizing a uniform quality-coverage function f instead of a decreasing one leads to a significant decline in computational complexity ($O(n \cdot m)$ instead of $O(n \cdot m^2)$). This observation is even more important when considering the case of central processing or distributed processing but on platforms with limited computation power.*

3.8. **Position constraints.** In the problem statement section, it was declared that the projection of all agents 3D position on the planar must always be constrained inside the region of interest, that is $q_i \in \Omega$, $\forall i \in \imath_n$ (this way convexity constraints were introduced in the collaborative control development). When implementing the aforementioned decentralized control laws, there can be no guarantee that all of the agents projection on the planar reside in Ω. It is possible that an agent leaves Ω despite the fact that its guaranteed sensing pattern can never entirely quit Ω (i.e. $C_i^{gs} \cap \Omega \neq \emptyset$, $\forall i \in I_n$). Still, even if the agents' measurements regarding its 3D position are inside Ω, there is no guarantee that the agent is indeed inside Ω due to the aforementioned localization uncertainty. To guarantee constrained projection on the planar $q_i \in \Omega$, $\forall i \in I_n$, the entire localization uncertainty region C_i^u must reside in it since the agent's real location may be anywhere in C_i^u, that is $C_i^u \subseteq \Omega \Rightarrow C_i^u \cap \Omega \equiv C_i^u$. This is achieved by replacing Ω with a subset $\Omega_i^s \subseteq \Omega$, $\forall i \in I_n$ defined as the Minkowski difference of Ω with the planar disk $D(r_i^q) = q \in \Omega : \ \|q\| \leq r_i^q$, as depicted in Figure 3. Formally,

$$\Omega_i^s := q \in \Omega : \ q + D(r_i^q) \subseteq \Omega, \quad i \in I_n.$$

This definition accounts for the different uncertainty radii r_i^q among agents, leading to different notions of the region of interest throughout the network. It also saves on computational effort by constraining only the projection q_i of each agent on the planar inside Ω_i^s instead of the whole positioning uncertainty regions. In case that the 2D projection of the agent on the planar reaches the borders of the redefined region of interest, i.e. $q_i \in \partial\Omega^s$, and the control input u_i^q leads the agent to quit Ω_i^s, then the computed control input u_i^q is replaced by its projection on the boundaries of Ω_i^s. Consequently, the extended control law guaranteeing that all agents remain inside Ω at all times is

$$\tilde{u}_{i,q} = \begin{cases} proj_{\partial\Omega_i^s} u_i^q, & if \quad q_i \in \partial\Omega_i^s \ \wedge \ q_i + \varepsilon u_i^q \notin \Omega_i^s \\ u_{i,q}, & otherwise \end{cases} \tag{13}$$

where ε an infinitesimally small positive value. This extension does not jeopardize the monotonicity or stability of the network's overall performance.

Should the constraints of the agents' projection on Ω be lifted, the aforementioned control laws would be feasible even in cases of non-convex regions of interest.

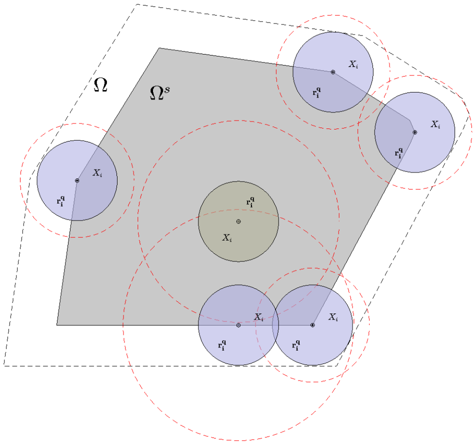


FIGURE 3. Concept of Ω^s. All nodes are constrained inside Ω^s (grey area), so the projection of their uncertainty region (blue circles) on the planar indicated by Ω^s will always reside inside Ω (dashed black line). On the contrary, the sensing patterns may contain areas outside of Ω, that is $C_i^g s \setminus \Omega \neq \emptyset$.

3.9. **Collision avoidance.** In an aerial robotic swarm that operates in a decentralized fashion, meaning that all nodes of the network make decisions regarding their movement on their own based on information of their environment, no central overseer exists to ensure that the designed trajectories do not lead in collisions. The natural tendency of the controllers is for robots to push away from one another. However, this is not a guarantee of collision avoidance. In a practical set up, the controllers would have to be combined with an obstacle/collision avoidance extension. The control laws $u_{i,q}, u_{i,z}$ (8) that correspond to the 3D position control of each MAA, collectively fail to take into consideration the possibility of a collision between two or more MAAs, formally $\left[q_i \equiv q_j \ \wedge \ z_i \equiv z_j\right], i \neq j,\ i,j \in I_n$, where q_i, q_j, z_i, z_j the true position of the MAAs. This way, though, we do not account for the localization uncertainty as described by (3). Since each agent can not precisely estimate its position, two or more agents may collide while each considers being in safe distance from other MAAs. Therefore a possible collision may occur when

$$\left[q_i \pm k_1 \cdot r_i^q \equiv q_j \pm k_2 \cdot r_j^q \ \wedge \ z_i \pm k_3 \cdot r_i^z \equiv z_j \pm k_4 \cdot r_j^z\right]$$

for $k_1, k_2, k_3, k_4 \in [0,1]$, $i \neq j,\ i,j \in I_n$. To address this issue, we expand the control laws to cover for this case.

Definition 1. *We define the minimum safety distances d_{ij}^q, d_{ij}^z that MAA i is allowed to approach j horizontally and vertically respectively as*

$$\begin{aligned} d_{ij}^q &:= \left\| r_i^q + r_j^q \right\| + (\beta_i^q + \beta_j^q) + \varepsilon^q, \quad r_i^q, r_j^q \in \mathbb{R}^2 \\ d_{ij}^z &:= r_i^z + r_j^z + (\beta_i^z + \beta_j^z) + \varepsilon^z, \quad r_i^z, r_j^z \in \mathbb{R},\ i \neq j,\ i,j \in I_n \end{aligned}$$

where $\varepsilon^q, \varepsilon^z$ infinitely small, real positive constants and $\beta_i^q, \beta_j^q, \beta_i^z, \beta_j^z$ real positive values providing information about the 3D size of the objects-agents. Assuming all MAAs are point-masses, $\beta_i^q, \beta_j^q, \beta_i^z, \beta_j^z \to 0$.

Remark 4. *It should be noted that examining vertical and horizontal distance between two agents to test for collision must take place separately instead of simply calculating the 3D Euclidean norm. The reason is that there exists no correlation between altitude and position measurements for the assumptions of subsection III-B. If one assumed only one metric for safe distance, i.e.* $\sqrt{\left\|q_i - q_j\right\|^2 + \left\|z_i - z_j\right\|^2} \leq \left\|[d_{ij}^q, d_{ij}^z]^T\right\|$*, the control law extension could in some cases falsely indicate possible collisions.*

Remark 5. *It is obvious from the above remark that* $[d_{ij}^q, d_{ij}^z]^T = [d_{ji}^q, d_{ji}^z]^T$*. If uncertainty bounds are common throughout the network and assuming dimensionless or same-size agents, safe distance is also common, reducing computational complexity from* $O(n-1)$ *to* $O(1)$*. Generally, in smaller networks this computational complexity may be insignificant but for large swarms, dynamic collision avoidance extensions could lead to strenuous computations.*

When robot i moves towards robot j, i.e. $u_{i,q}^T(q_j - q_i) \geq 0 \quad \wedge\!\!\!\vee \quad u_{i,z}(z_j - z_i) \geq 0$, and the distance between them drops below the joint threshold $\left\|q_i - q_j\right\| \leq d_{ij}^q \wedge \left\|z_i - z_j\right\| \leq d_{ij}^z$, the respective distributed control laws of the involved robots are nullified to prevent any further approach. Once the control laws no longer force robot i to approach robot j or the distance between them tends to increase, movement restrictions are lifted. The enhanced localization controllers for each robot i are

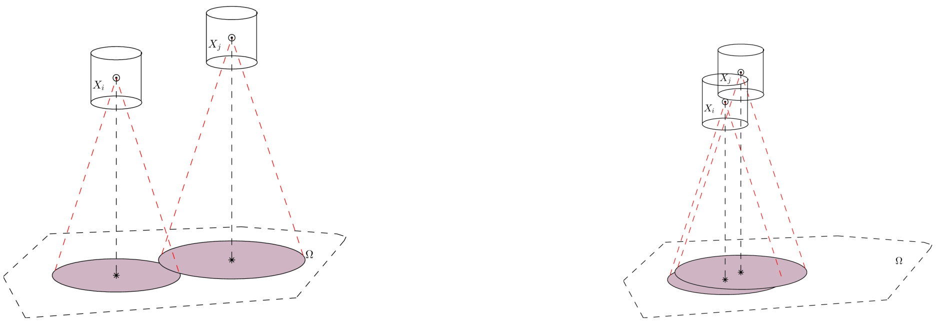
$$\hat{u}_{i,q} = \begin{cases} 0, & if\ \exists j \in I_n: \ u_{i,q}^T(q_j - q_i) \geq 0 \ \wedge \\ & \quad \left\|q_i - q_j\right\| \leq d_{ij}^q \wedge \left\|z_i - z_j\right\| \leq d_{ij}^z \\ u_{i,q}, & \quad otherwise \end{cases} \tag{14}$$

$$\hat{u}_{i,z} = \begin{cases} 0, & if \ \ \exists j \in I_n: \ \ u_{i,z}(z_j - z_i) \geq 0 \ \wedge \\ & \quad \left\|q_i - q_j\right\| \leq d_{ij}^q \wedge \left\|z_i - z_j\right\| \leq d_{ij}^z \\ u_{i,z}, & \quad otherwise \end{cases} \tag{15}$$

This modification in the decentralized control loop does not affect the monotonicity of the coverage-quality objective $\mathscr{H}$ since it either renders the value indicated by the decentralized control laws (monotonicity proven) or zero in the case of a possible collision which results in $\mathscr{H}$ being unaffected by this agent, that is the agents contribution at $t + dt$ is the same as at t. Moreover, this extension only acts on the translational movement control signals as they may cause the collision, leaving unchanged the control signals corresponding to the PTZ-camera (i.e. $\hat{u}_{i,\theta} = u_{i,\theta}$, $\hat{u}_{i,h} = u_{i,h}$, $\hat{u}_{i,\delta} = u_{i,\delta}$).

Remark 6. *In spite of the collision avoidance extensions, the derivative of* $\mathscr{H}$ *remains locally Lipschitz continuous. Contrary to the collision-ignoring controllers that were linearly dependant on the derivatives of the coverage-quality objective, the same is not true regarding the control laws. Still, the stability of the network stands.*

Two or more agents may be in peril of collision even though they do not consider each other a neighbour. That is due to the sensing pattern-based definition of neighbours coupled with the PTZ capability of the camera. When using downwards/nadir facing cameras physical proximity and neighbouring characteristics are linked. However, this is no longer the case when using PTZ-cameras as two very distant agents may be considered neighbours and two agents in the verge of collision may not be so. Therefore, in this case, to implement the aforementioned approach for collision avoidance, each agent requires complete knowledge of the current state of the network. This remark pauses a major obstacle when attempting a real-world implementation. To tackle it, a multi-hop communications algorithm has been proposed in [23].



(A) Collision-free.

(B) Possible collision.

FIGURE 4. Collision concept.

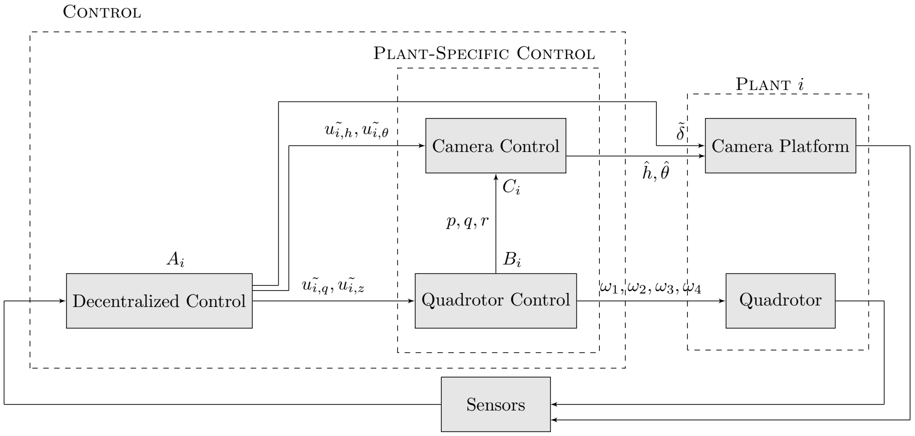


FIGURE 5. Control structure

4. SWARM OF QUADROTORS

Let's consider the case of utilizing a swarm consisting of n quadrotors for visual surveillance of the region of interest Ω. Implementing the aforementioned collaborative control strategy would formulate the desired trajectories in a distributed manner for each node of the network, indicating the desired tranlational/angular velocities. However, the dynamic model of the MMAs assumed for that strategy was capable of only translational movement as the MAAs were abstractly approximated by point masses to accommodate the possibility of groups of various types of UAVs. By determining the swarm to comprise of quadrotors, the new dynamics of the team need to be accounted for. Keeping the abstract character of the collaborative control strategy, the dynamics (2) remain unchanged. Instead, a secondary control level, aware of the node's specific dynamics, is introduced to enable the quadrotor to follow the trajectory indicated by the collaborative control strategy. The diagram of Figure 5 illustrates the proposed control structure. The controller A_i implementing the aforementioned self-organizing strategy receives the estimation of the current states of the quadrotors and their cameras and calculates in every iteration the nominal time derivatives of all states which can be translated in a waypoint to move towards and the desired Pan-Tilt-Zoom angles. They are the inputs of the trajectory tracking controller B_i which controls the angular velocities of the four motors and, thus, the rotational and translational movements of the quadrotor. Controller C_i is responsible for the rotation of the platform upon which the camera is mounted. It is obvious that the Pan-Tilt angles derived from the collaborative control strategy are affected by the roll, pitch and yaw angles of the quadrotor.

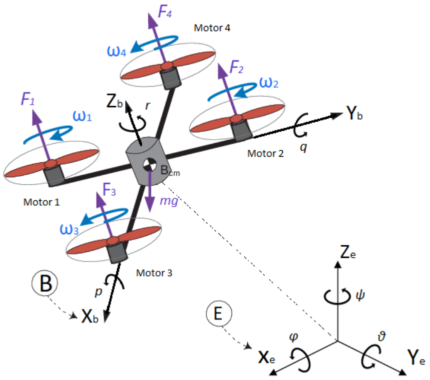


FIGURE 6. Quadrotor concept.

4.1. **Quadrotor Dynamics and Control.** We adopt the formulation for quadrotor dynamics and geometric nonlinear controller of [31], which we summarize in brief. First, the dynamics of the quadrotor's center can be written as

$$m^A\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} +^A R_B \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} . \tag{16}$$

Where m is the mass of the quadrotor, ${}^A r$ is the position of the quadrotor in the inertial frame A, g is the gravitational constant, ${}^A R_B$ is the rotation matrix from the body from B to the inertial frame A, and F_i is the force applied by the i-th propeller. Let the input u_1 be defined as

$$u_1 = F_1 + F_2 + F_3 + F_4 \tag{17}$$

For the angular velocity, let p, q, r be the angular velocities about each of the quadrotor's three basis vectors. Then the dynamics evolve (in the body frame) according to

$$I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \mathbf{u}_2 - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times I \begin{bmatrix} p \\ q \\ r \end{bmatrix} , \tag{18}$$

where I is the inertia tensor, and u_2 is defined by

$$\mathbf{u}_2 = \begin{bmatrix} 0 & l & 0 & -l \\ -l & 0 & l & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} . \tag{19}$$

In the above, we define l to be the distance from the propeller to the center of the robot, and γ is a constant ratio depending on our lift and drag (from our motor model).

To derive our controller, define

$$\ddot{\mathbf{r}}^{\text{des}} = \ddot{\mathbf{r}}_T - K_d(\dot{\mathbf{r}} - \dot{\mathbf{r}}_T) - K_p(\mathbf{r} - \mathbf{r}_T) , \tag{20}$$

where $\mathbf{r}_T$ is our desired position. Then, define

$$\mathbf{F}^{\text{des}} = m\ddot{\mathbf{r}}^{\text{des}} + [0, 0, mg]^T \tag{21}$$

And calculate u_1 as

$$u_1 = [0,0,1]^B R_A \mathbf{F}^{\text{des}} \tag{22}$$

To calculate u_2, define

$$\mathbf{b}_3{}^{\text{des}} = \frac{\mathbf{F}^{\text{des}}}{||\mathbf{F}^{\text{des}}||} \tag{23}$$

$$\mathbf{a}_\psi = [\cos\psi_T, \sin\psi_T, 0]^T \tag{24}$$

$$\mathbf{b}_2{}^{\text{des}} = \frac{\mathbf{b}_3{}^{\text{des}} \times \mathbf{a}_\psi}{||\mathbf{b}_3{}^{\text{des}} \times \mathbf{a}_\psi||} \tag{25}$$

$$R^{\text{des}} = [\mathbf{b}_2{}^{\text{des}} \times \mathbf{b}_3{}^{\text{des}}, \mathbf{b}_2{}^{\text{des}}, \mathbf{b}_3{}^{\text{des}}] \tag{26}$$

$$\mathbf{e}_R = \frac{1}{2}(R^{\text{des}^T} R - R^T R^{\text{des}})^\vee \tag{27}$$

where ψ is the yaw angle, the $\vee$ operator maps elements of SO(3) to $\mathbb{R}^3$, K_R and K_ω are diagonal gain matrices, and $\mathbf{e}_\omega = \omega - \omega^{\text{des}}$ is the angular velocity error. Finally, we can calculate u_2 as

$$\mathbf{u}_2 = I(-K_R\mathbf{e}_R - K_\omega\mathbf{e}_\omega)\ , \tag{28}$$

4.2. **Correction of PTZ angles.** It is obvious that the quadrotor's Euler angles temper with the camera's nominal Pan-Tilt angles, computed from the time integral of their respective derivatives

$$h_i' = \int u_{\tilde{i},h}\, dt, \quad \theta_i' = \int u_{\tilde{i},\theta}\, dt,$$

and thus, need to be adjusted accordingly. It should be noted that all three Euler angles interfere with the pan angle but only roll and pitch interfere with the tilt angle.

Let's consider a point $S = [0,0,a]^T$ residing on the yaw axis O_z. If the quadrotor rotates by ϕ around O_x (roll) and then θ around O_y (pitch) but but does not yaw, the respective tilt angle will be the one between the new z'–axis and the previous z–axis. Then, given the rotation matrix

$$R_y(\phi)R_x(\theta) = \begin{bmatrix} c\theta & s\theta s\phi & -c\phi s\theta \\ 0 & c\phi & -s\phi \\ -s\phi & s\phi c\theta & -c\phi c\theta \end{bmatrix},$$

point S is $[-a\cos\phi\sin\theta,\ -a\sin\phi,\ -a\cos\phi\cos\theta]^T$ in the new coordinates. Switching to spherical coordinates, S is

$$\rho' = ||S|| = a,\ \lambda' = \arccos(\cos\phi\cos\theta),\ \gamma' = \arctan\frac{\tan\phi}{\sin\theta}.$$

Assuming ψ to be the yaw angle of the quadrotor, the desired Pan-Tilt angles are

$$\bar{h}_i = h_i' - \gamma' - \psi, \quad \bar{\theta}_i = \theta_i - \lambda'. \tag{29}$$

4.3. **Adaptive gains of the distributed controller.** The coupling of the aforementioned controllers bears no synchronization assurances as the distributed controller responsible for path planning have no embedded information regarding the dexterity of the system. A quadrotor may be unable to reach the indicated target point in the designated time frame, thus some feedback on the progress made is necessary to inform the distributed controller that the system-specific controller is unable to keep up. Naturally, the indicated metric of progress would be the evolution of the coverage-quality objective of the network. However, since that would contradict the distributed nature of the system's design, the tracking errors are instead used. The

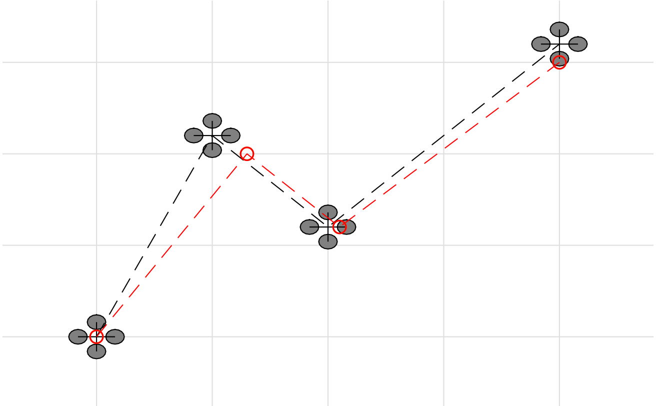


FIGURE 7. A quadrotor may be unable to reach the indicated target point in time.

updated distributed controller's gains are:

$$K_{i,q} = K_{i,q}^{0}\frac{2}{e^{\lambda|q-q_d|}+1}, \quad K_{i,z} = K_{i,z}^{0}\frac{2}{e^{\lambda|z-z_d|}+1}$$

for some $\lambda > 0$. When the errors are significant, the distributed controller produces smaller desired inputs (smaller euclidean distance till the next point, though still in the direction that increases $\mathscr{H}$) allowing the system's implementation to meet these targets.

5. SIMULATION STUDIES

Simulation studies were conducted in order to evaluate the efficiency of the proposed control strategy when implemented using a swarm of Crazyflie 2.0 quadrotors (`https://www.bitcraze.io/products/old-products/crazyflie-2-0/`). The Crazyflie is an open source quadrotor platform with a characteristic length of 92-mm and a mass, including battery, of 30g. The Crazyflie has an on-board microcontroller and Inertial Measurement Unit (IMU) which enables low-level control and partial state estimation. For consistency, the region of interest Ω was selected to be the same as in [25]. The space density function was assumed to be $\phi(q) = 1, \forall q \in \Omega$, assigning equal importance to all points inside the region of interest. All agents are considered to have various localization uncertainties. The camera state limits were $h_i^{\max} = 50^\circ$, $\delta_i^{\min} = 15^\circ$ and $\delta_i^{\max} = 35^\circ$, $\forall i \in I_n$. The MAAs' cells are shown filled in grey with solid black boundaries while the boundaries of guaranteed sensed regions are shown in dashed red lines. Case studies I,III assume a swarm of quadrotors, whereas case studies II and IV assume single integrator dynamics to check directly the distributed controller. Videos and data of the simulations are available.

5.1. **Case study I.**

This case study simulates the integration of the quadrotor dynamics in the collaborative framework as depicted in 5 and explores the effect the aggressiveness of the quadrotor's control has on the overall performance of the system. This simulation compares the evolution of the system if the nominal control indicated by the decentralized algorithm was achieved (MAAs) with the one a quadrotor is capable of achieving when utilizing a geometric controller tuned in the lab for limited velocities and accelerations. For the sake of consistency, the initial state of the network used in Simulation I of [27] also serves as initial configuration here with slightly augmented control gains, though, in the decentralized algorithm to force the system converge faster to the final optimal configuration and, thus, highlight any weakness caused by the underactuation of the quadrotor that makes it impossible to follow a trajectory designed for single integrator dynamics. For the geometric controller, the gains were set:

$$K_p = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix} \quad [m^{-1}]$$

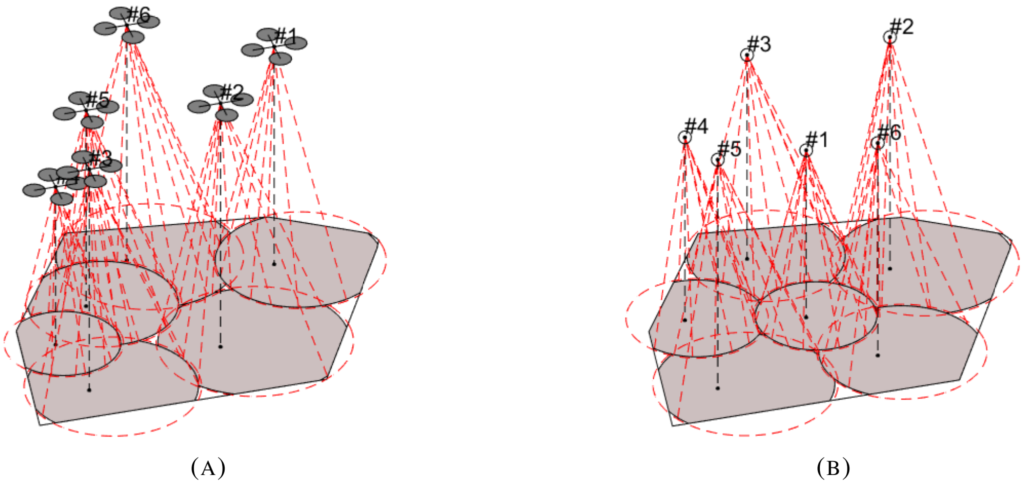
$$K_d = 2 \cdot \sqrt{K_p} \quad [s - m^{-1}]$$

$$K_R = 500 \cdot \mathbb{1} \quad [rad^{-1}]$$

$$K_\omega = 70 \cdot \mathbb{1} \quad [s - rad^{-1}]$$

For the quadrotor implementation we consider a time step of 2ms. For the decentralized controller, the time step is set at 100ms. This choice is not random as this is approximately the time frame for an MAA to implement each iteration of the algorithm (localize, communicate and compute control laws).

Figures 8 and 9 illustrate the swarm's final state and the evolution of the coverage–quality objective $\mathscr{H}$ and collective area under surveillance $\mathscr{A}$ over time respectively. Though the final configuration of the MAAs is different from that of the Crazyflies, ultimately both achieve similar metrics $\mathscr{H}$ and $\mathscr{A} \approx 97\%$. As designed, the coverage–quality objective is monotonically increasing over time till it reaches its maximum. The Crazyflie swarm, however, presents a transient response during which the system struggles to follow the nominal trajectory indicated by the distributed controller. Figure 11 illustrates these localization errors. The errors in all three dimensions are less than 0.2m and converge to zero. Notice that the transient period coincides with significant positional errors. Finally, Figure 13 depicts the position of each Crazyflie over time. It is worth noting, that even though the system is designed with strict constraints $q \in \Omega, z \in [z^{min}, z^{max}]$, when transferring from single integrator dynamics to real-world systems, these bounds may be violated as no guarantees exist for the positional error. Should a bound exist, the Portyagin difference between the feasible set and the error set could provide offline conservative bounds for the distributed controller.



(A)

(B)

FIGURE 8. Simulation I: Final configuration for swarm of Crazyflies (a) and MAAs (b) with integrated dynamics and control.

5.2. **Case study II.** In this simulation, the effect of unaccounted localization uncertainties of each node of the system on the perceived quality of coverage is explored. The distributed controller is tested alone, without the geometric controller (assume MAAs instead of Crazyflie quadrotors). A similar set up with Case study I

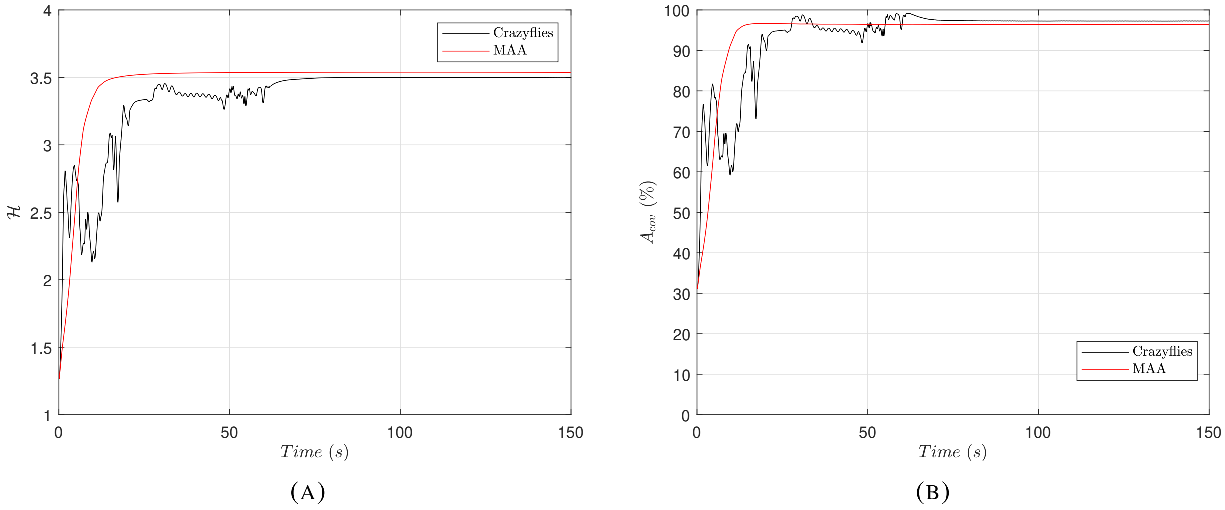


FIGURE 9. Simulation I: Comparison of $\mathscr{H}$ evolution and total covered area.

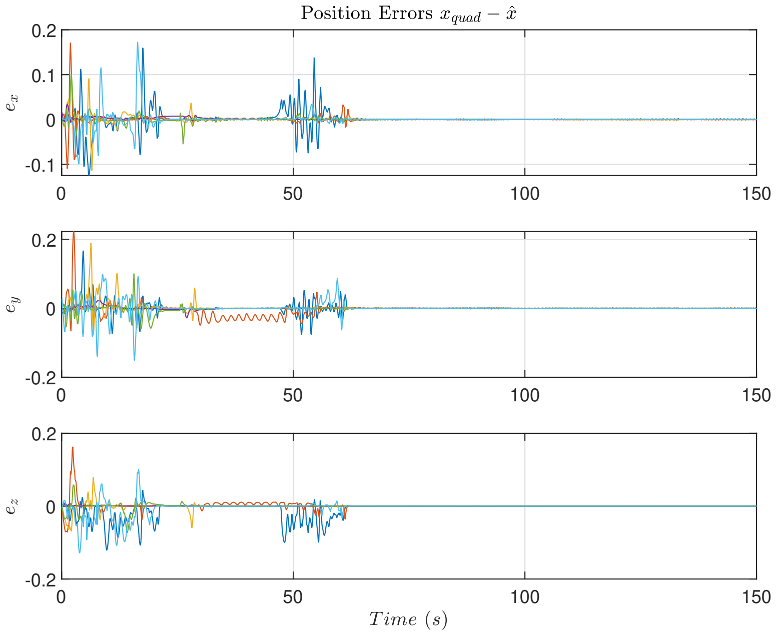


FIGURE 10

FIGURE 11. Simulation I: Localization errors.

is considered. The 6 MAAs, equipped with PTZ-cameras, utilize the aforementioned distributed control laws assuming their location estimation to be precise. Figure 15 illustrates the collective coverage objective that the swarm considers achieved when ignoring uncertainties in solid black compared to the actual one achieved for the real locations of the nodes in dashed red. Figures 14 (a) and (b) depict the virtual final configuration of the swarm compared to the real one respectively. This simulation emphasizes the importance of accounting for the localization uncertainty of each node of the team when attempting collaborative visual coverage of a region of interest.

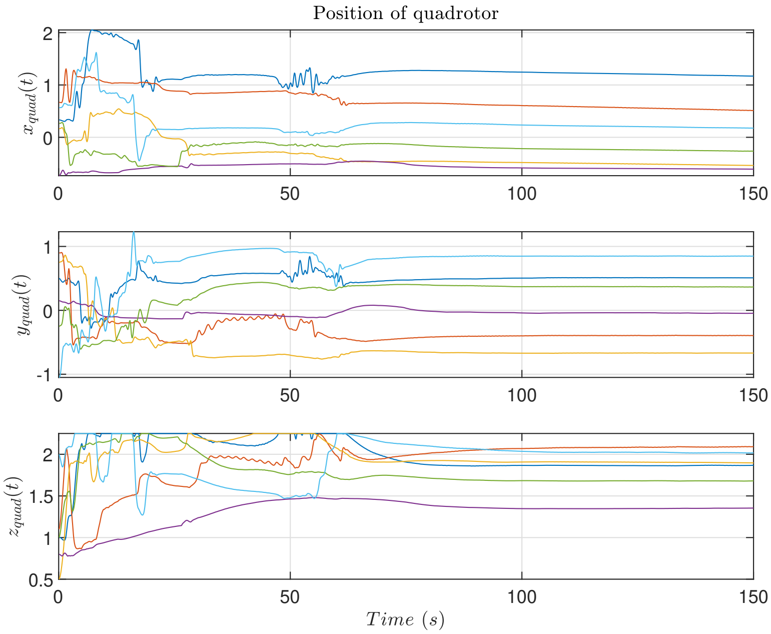
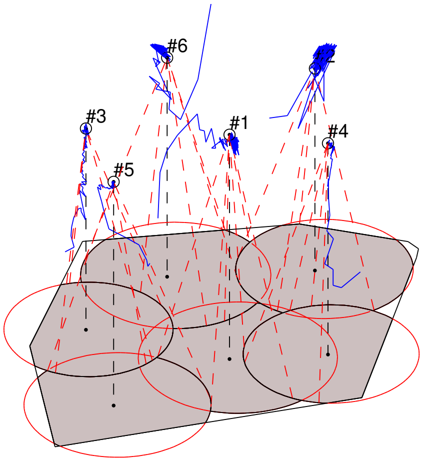
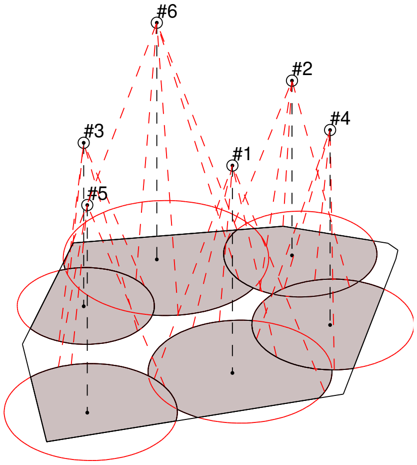


FIGURE 12

FIGURE 13. Simulation I: Position of quadrotors over time.



(A)



(B)

FIGURE 14. Simulation II: Virtual (a) and real (b) final configurations.

5.3. **Case study III.** In this case study, we test the ability of the suggested scheme to explore non-convex regions of interest. Most control strategies for distributed coverage are trapped in local optima when dealing with non-convex Ω. Solutions have been proposed in [35, 36, 37]. The proposed gradient-based method of this project is able to escape local optima and find a globally optimal configuration if we relax the constraints on the position of the quadrotors as in [23]. Figure 16 illustrates the final configuration of the team offering coverage of Ω. The scheme achieves 98% coverage and the same $\mathscr{H}$ as the swarm of MAAs, though the rise time is considerably more, as seen in Fig. 17. The positional errors depicted in Fig. 19 converge quickly to

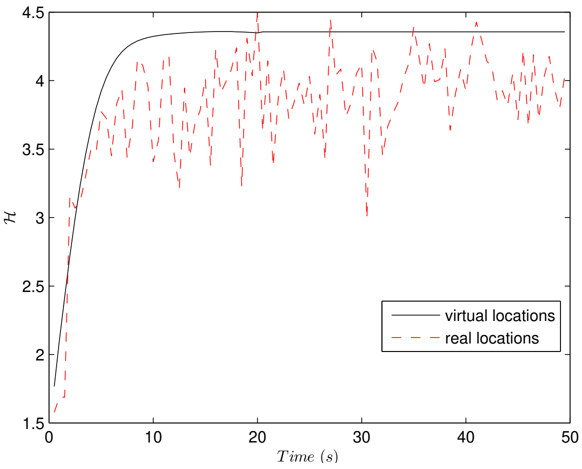
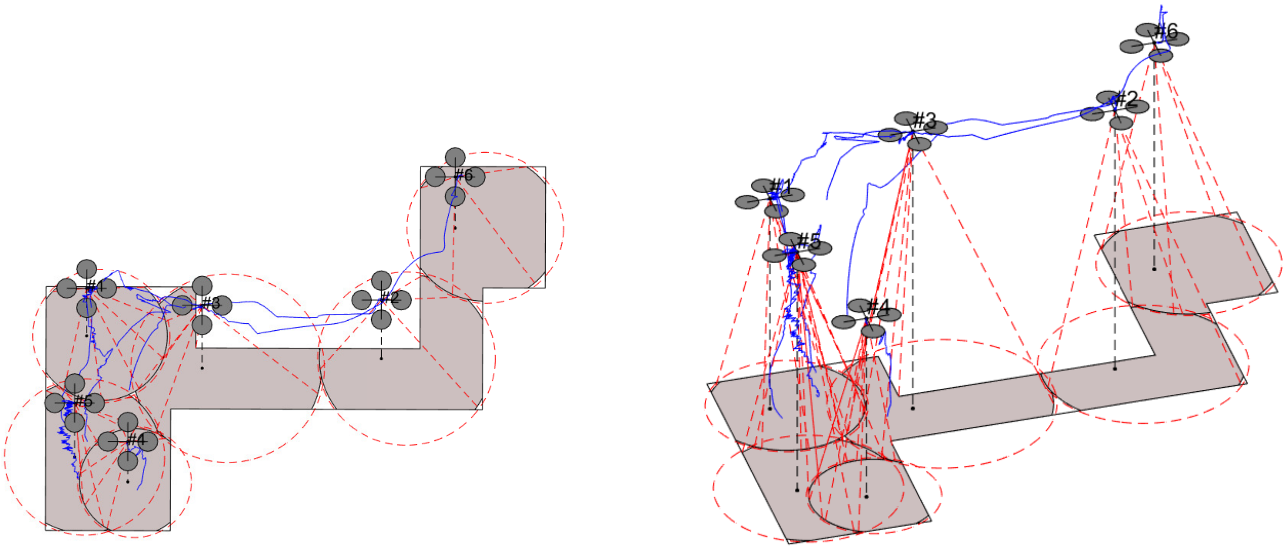


FIGURE 15. Simulation II: Coverage–quality objective comparison of virtual and actual configuration.

zero for y, z directions and with a minor oscillation of less than 0.005m in the x direction of quadrotors 2 and 4.



(A) (B)

FIGURE 16. Simulation III: Final configuration for swarm of Crazyflies over non-convex environment Ω.

5.4. **Case study IV.** This simulation emphasizes that the distributed control design is resilient to failures of any number of agents. Once the swarm of MAAs (no quadrotors used in this simulation) has converged to a local maxima, 2 agents suddenly fail/collapse leaving the remaining agents to cover for this loss. The proposed strategy is adaptive and re–configures to once again monotonically increase the quality-coverage criterion. As expected, the remaining agents are unable to reach the previous maxima of $\mathscr{H}$. Figures 22 (a)

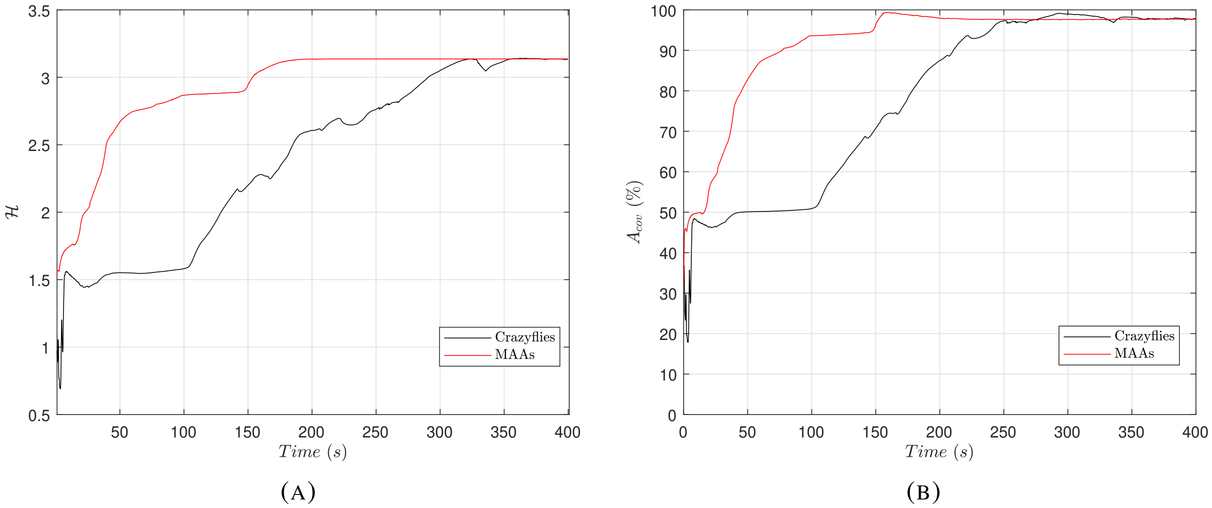


FIGURE 17. Simulation III: Coverage–quality function $\mathscr{H}$ evolution and total covered area state for swarm of Crazyflies over non-convex environment Ω.

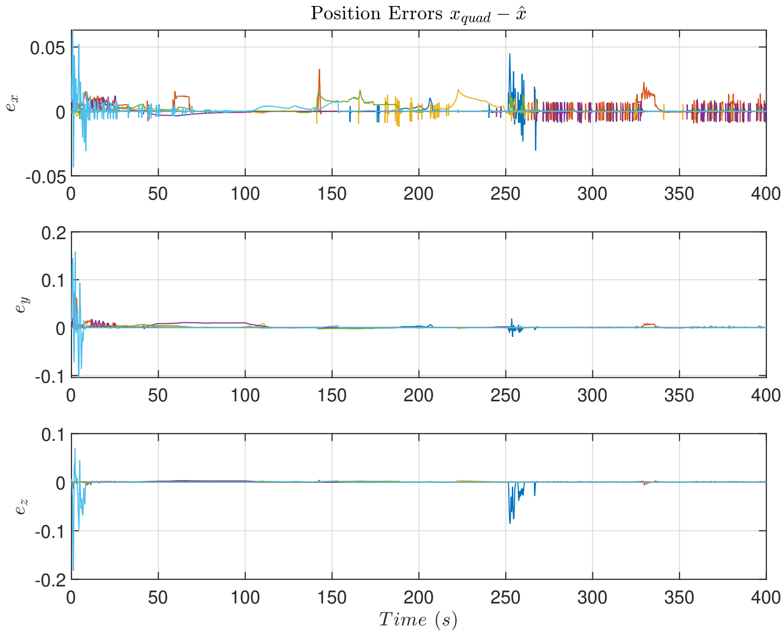


FIGURE 18

FIGURE 19. Simulation III: Localization errors.

and (d) depict the initial and final configurations of the swarm respectively. Figures 22 (b) and (c) illustrate the network's configuration before and after the loss of 2 MAAs. Figure 23a illustrates the achieved coverage objective $\mathscr{H}$. Notice the radical drop (around 30% was what those MAAs contributed in total sensed area and $\mathscr{H}$) at t=60s. The system, however, is able to bounce back and once again optimize the visual coverage of Ω with the resources it has left. There is a small deviation in total area coverage between the final configuration

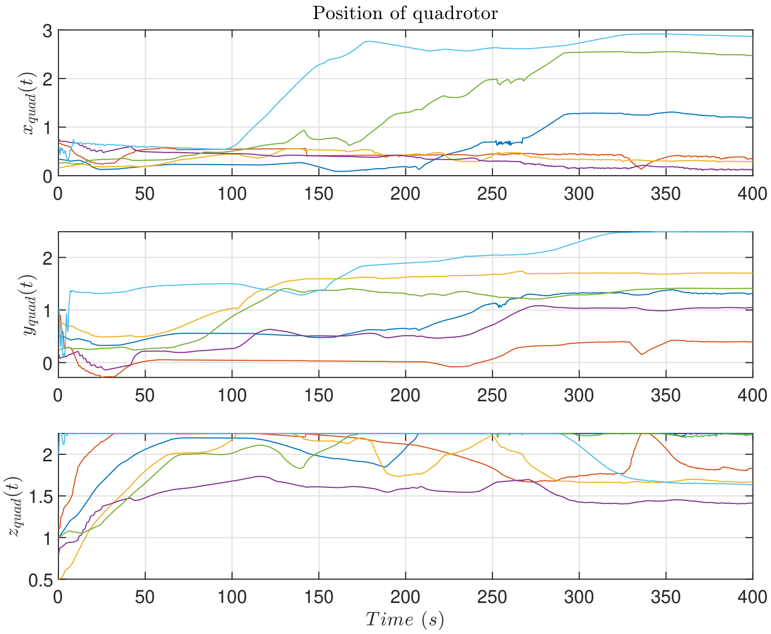


FIGURE 20

FIGURE 21. Simulation III: Position of quadrotors over time.

for N=6 and N=4 (around 3% less area is sensed), contrary to the significant deviation in $\mathscr{H}$. In other words, the swarm attempts to make up for the loss in sensed area by sacrificing the quality of coverage.

6. CONCLUSIONS

This article examines the efficiency of the collaborative visual aerial algorithm when implemented on a swarm of quadrotors, equipped with PTZ–cameras while operating under localization uncertainty. The system's control design is hierarchically structured with a decentralized control algorithm generating the nominal path and a geometric nonlinear controller, aware of the agent's dynamics, tracking that trajectory. Simulation studies were offered to evaluate the feasibility of the proposed configuration.

REFERENCES

[1] F. Abbasi, A. Mesbahi, J. M. Velni, A team-based approach for coverage control of moving sensor networks, Automatica 81 (2017) 342–349.

[2] A. Pierson, L. C. Figueiredo, L. C. A. Pimenta, M. Schwager, Adapting to sensing and actuation variations in multi-robot coverage, The International Journal of Robotics Research 36 (3) (2017) 337–354.

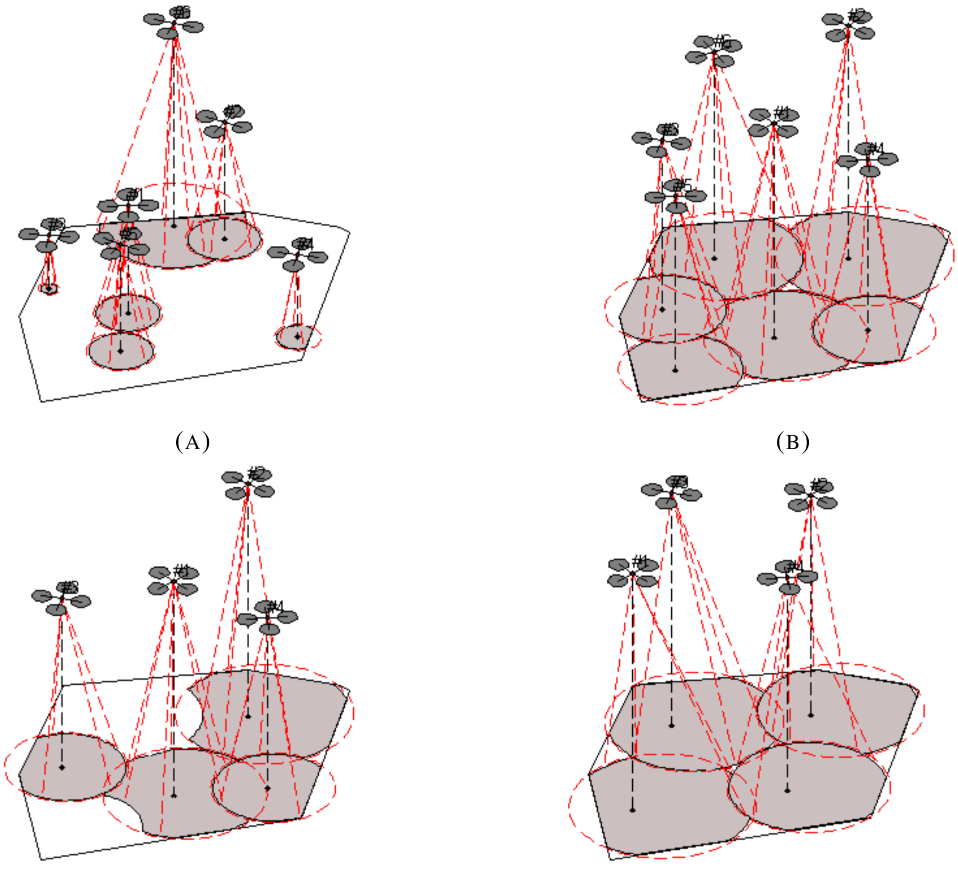
[3] J. M. Palacios-Gasós, E. Montijano, C. Sagüés, S. Llorente, Distributed coverage estimation and control for multirobot persistent tasks, IEEE Transactions on Robotics PP (99) (2016) 1–17.

[4] C. Franco, D. M. Stipanović, G. López-Nicolás, C. Sagüés, S. Llorente, Persistent coverage control for a team of agents with collision avoidance, European Journal of Control 22 (2015) 30–45.

[5] Y. Kantaros, M. Thanou, A. Tzes, Distributed coverage control for concave areas by a heterogeneous robot-swarm with visibility sensing constraints, Automatica 53 (2015) 195–207.

[6] R. J. Alitappeh, K. Jeddisaravi, F. G. Guimarães, Multi-objective multi-robot deployment in a dynamic environment, Soft Computing 21 (21) (2016) 1–17.

[7] F. Sharifi, A. Chamseddine, H. Mahboubi, Y. Zhang, A. G. Aghdam, A distributed deployment strategy for a network of cooperative autonomous vehicles, IEEE Transactions on Control Systems Technology 23 (2) (2015) 737–745.



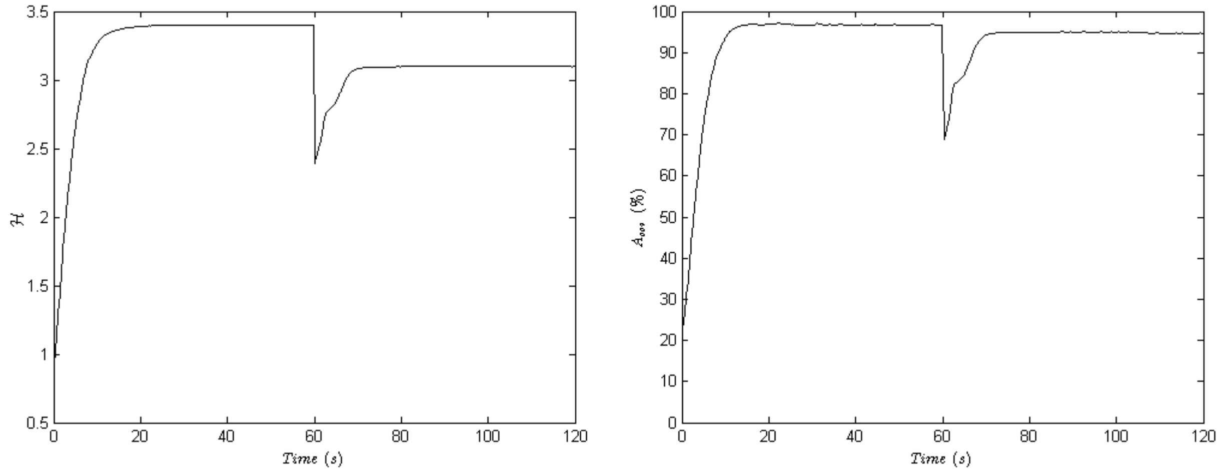
(A)

(B)

(C)

(D)

FIGURE 22. Simulation IV: Initial (a) and final configuration (d) in 2-MAAs failure scenario at t=60s. Configuration prior (b) and after the incident (c).



(A)

(B)

FIGURE 23. Simulation IV: Coverage–quality objective evolution (23a) and total area covered in 2-MAAs failure scenario at t=60s ().

[8] J. M. Luna, R. Fierro, C. Abdallah, J. Wood, An adaptive coverage control algorithm for deployment of nonholonomic mobile sensors, in: Proceedings 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 1250–1256.

[9] Y. Stergiopoulos, A. Tzes, Cooperative positioning/orientation control of mobile heterogeneous anisotropic sensor networks for area coverage, in: Proceedings IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 2014, pp. 1106–1111.

[10] O. Arslan, D. E. Koditschek, Voronoi-based coverage control of heterogeneous disk-shaped robots, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), IEEE, Stockholm, Sweden, 2016, pp. 4259–4266.

[11] Y. Stergiopoulos, M. Thanou, A. Tzes, Distributed collaborative coverage-control schemes for non-convex domains, IEEE Transactions on Automatic Control 60 (9) (2015) 2422–2427. doi:10.1109/TAC.2015.2409903.

[12] C. Nowzari, J. Cortés, Self-triggered coordination of robotic networks for optimal deployment, Automatica 48 (6) (2012) 1077–1087.

[13] V. Ramaswamy, J. R. Marden, A sensor coverage game with improved efficiency guarantees, in: Proceedings American Control Conference (ACC), Boston, MA, USA, 2016, pp. 6399–6404.

[14] A. Kwok, S. Martinez, A distributed deterministic annealing algorithm for limited-range sensor coverage, IEEE Transactions on Control Systems Technology 19 (4) (2011) 792–804.

[15] A. Mavrommati, E. Tzorakoleftherakis, I. Abraham, T. D. Murphey, Real-time area coverage and target localization using receding-horizon ergodic exploration, IEEE Transactions on Robotics 34 (1) (2018) 62–80.

[16] H. X. Pham, H. M. La, D. Feil-Seifer, A. Nefian, Cooperative and distributed reinforcement learning of drones for field coverage, arXiv preprint arXiv:1803.07250 (2018).

[17] A. Howard, M. J. Matarić, G. S. Sukhatme, Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem, in: Distributed Autonomous Robotic Systems 5, Springer, 2002, pp. 299–308.

[18] J. Habibi, H. Mahboubi, A. G. Aghdam, Distributed coverage control of mobile sensor networks subject to measurement error, IEEE Transactions on Automatic Control 61 (11) (2016) 3330–3343.

[19] B. Davis, I. Karamouzas, S. J. Guy, C-OPT: Coverage-aware trajectory optimization under uncertainty, IEEE Robotics and Automation Letters 1 (2) (2016) 1020–1027.

[20] S. Papatheodorou, A. Tzes, K. Giannousakis, Y. Stergiopoulos, Distributed area coverage control with imprecise robot localization: Simulation and experimental studies, International Journal of Advanced Robotic Systems 15 (5) (Sep. 2018).

[21] O. Arslan, D. E. Koditschek, Voronoi-based coverage control of heterogeneous disk-shaped robots, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 4259–4266.

[22] S. Papatheodorou, A. Tzes, Y. Stergiopoulos, Collaborative visual area coverage, Robotics and Autonomous Systems 92 (2017) 126–138.

[23] M. Schwager, B. J. Julian, M. Angermann, D. Rus, Eyes in the sky: Decentralized control for the deployment of robotic camera networks, Proceedings of the IEEE 99 (9) (2011) 1541–1561.

[24] S. Papatheodorou, A. Tzes, Cooperative visual convex area coverage using a tessellation–free strategy, in: 56th IEEE Conference on Decision and Control (CDC), Melbourne, Australia, 2017, pp. 4662–4667.

[25] M. Tzes, S. Papatheodorou, A. Tzes, Visual area coverage by heterogeneous aerial agents under imprecise localization, IEEE Control Systems Letters 2 (4) (2018) 623–628.

[26] O. Arslan, H. Min, D. E. Koditschek, Voronoi-based coverage control of pan/tilt/zoom camera networks, in: Robotics and Automation (ICRA), 2018 IEEE International Conference on, IEEE, 2018.

[27] N. Bousias, S. Papatheodorou, M. Tzes, A. Tzes, Collaborative visual area coverage using aerial agents equipped with ptz-cameras under localization uncertainty, in: 2019 18th European Control Conference (ECC), 2019, pp. 1079–1084. doi:10.23919/ECC.2019.8795665.

[28] N. Bousias, Visual area coverage with collaborative autonomous aerial vehicles (2019).

[29] N. Bousias, S. Papatheodorou, M. Tzes, A. Tzes, Distributed surveillance by a swarm of uavs operating under positional uncertainty, arXiv preprint arXiv:1910.10837 (2019).

[30] E. Hernandez-Martinez, G. Fernandez-Anaya, E. Ferreira, J. Flores-Godoy, A. Lopez-Gonzalez, Trajectory tracking of a quadcopter uav with optimal translational control., IFAC 48 (19) (2015) 226 – 231, 11th IFAC SYROCO 2015.

[31] D. Mellinger, V. Kumar, Minimum snap trajectory generation and control for quadrotors, in: 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 2520–2525.

[32] J. Cortes, S. Martinez, T. Karatas, F. Bullo, Coverage control for mobile sensing networks (April 2004). doi:10.1109/TRA.2004.824698.

[33] P.-A. Absil, K. Kurdyka, On the stable equilibrium points of gradient systems, Systems & Control Letters 55 (7) (2006) 573 – 577.

[34] M. Hirsch, S. Smale, Differential equations, dynamical systems, and linear algebra, no. 60 in Pure and applied mathematics, Acad. Press, San Diego, 1974.

[35] Y. Kantaros, M. Thanou, A. Tzes, Visibility-oriented coverage control of mobile robotic networks on non-convex regions, in: IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 1126–1131.

[36] A. Breitenmoser, M. Schwager, J. C. Metzger, R. Siegwart, D. Rus, Voronoi coverage of non-convex environments with a group of networked robots, in: Proceedings IEEE International Conference on Robotics and Automation, Anchorage, Alaska, USA, 2010, pp. 4982–4989.

[37] Y. Stergiopoulos, M. Thanou, A. Tzes, Distributed collaborative coverage-control schemes for non-convex domains, IEEE Transactions on Automatic Control 60 (9) (2015) 2422–2427.