Decentralized Control of UAV Swarms for Bandwidth-aware Video Surveillance using NMPC

Mohammad Amin Rezaei, Gioacchino Manfredi, Vito Andrea Racanelli, Luca De Cicco, and Saverio Mascolo

***Abstract*— This paper proposes a framework for controlling a drone swarm to achieve two goals: i) covering a desired region of interest through onboard cameras that capture videos to be sent in real-time to a Ground Control Station (GCS), and ii) ensuring the highest video quality possible given the available Internet network bandwidth. Indeed, the quality of received videos depends on both the available bandwidth, which directly influences video encoding bitrate, and the altitude of drones, which influences pixel density. Thus, contrary to the conventional assumption of uniform drone altitudes, we let drones to track a reference altitude that is function of the time-varying available bandwidth to improve visual quality. To achieve the aforementioned goals, we propose a leader-follower multi-agent system formation control problem. In this setup, the leader tracks a desired path using Nonlinear Model Predictive Control (NMPC) to cover the area of interest. Follower agents track the leader using NMPC, aiming at maximizing both the total coverage area and the quality of the videos sent to the GCS, considering the constraints imposed by the network available bandwidth. At the same time, we formulate the NMPC problem to ensure that the swarm maintains a formation characterized by a given overlap percentage between the videos captured by the drones while avoiding collisions. This allows dynamical stitching of the received videos at the GCS, enabling the execution of computer vision algorithms for tasks such as object detection and surveillance.**

***Index Terms*— Unmanned Aerial Vehicle (UAV), Multi-agent Systems, Formation Control, Non-linear Model Predictive Control (NMPC), Communication-aware control**

I. Introduction

Unmanned Aerial Vehicles (UAVs), such as multi-rotor drones, are increasingly being employed as mobile flying platforms to carry out several tasks such as photogrammetry [1], remote sensing [2], search and rescue missions [3], surveillance [4], firefighting [5], etc. To perform such tasks, UAVs are equipped with sensors such as RGB or RGB-D cameras, 3D LiDARs, mmWave radars, and event-based cameras, which produce data that can be either stored on the UAVs or sent in real-time through a wireless network connection to a Ground Control Station (GCS). A *UAV swarm* is a system of coordinated UAVs designed to carry out a common task collaboratively. On the one hand, compared to operations carried out by a single UAV, swarms improve data acquisition efficiency and accuracy, increase coverage, and ensure operational resilience. Furthermore, the cooperative nature of UAV swarms enables to carry out more complex tasks, such as collaborative mapping, multi-sensor data fusion, and distributed sensing. On the other hand, swarming requires both (i) providing the UAVs with autonomous capabilities such as path planning and obstacle avoidance and (ii) the design of control strategies to coordinate the UAVs to perform a given task efficiently according to suitable performance metrics.

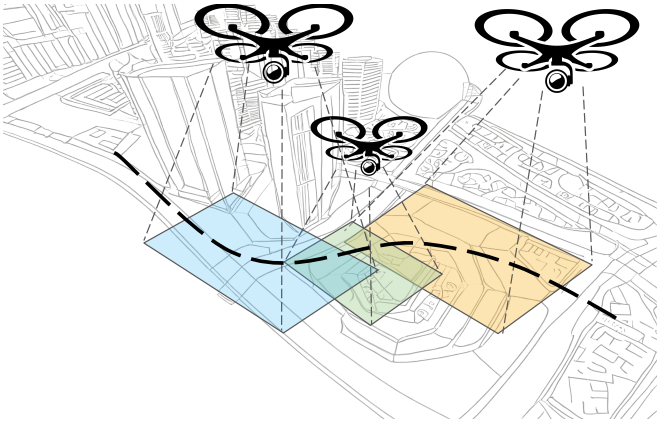


Fig. 1: A swarm of drones autonomously patrols a given area of interest while sending videos to a ground station

Coordination of UAVs in a swarm requires the development of specific control strategies that allow autonomous task execution. Such control strategies are often designed based on methodologies proposed by the *Multi-Agent Systems* (MAS) research community, comprising swarm intelligence [6], formation control [7], consensus algorithms [8], decentralized control algorithms [9].

Specific tasks such as surveillance, object detection, and patrolling, often require to stream in real-time the videos captured by the onboard cameras to a GCS. In this context, it is particularly indicated to use swarms of drones to improve mission efficiency, both in terms of mission duration and situational awareness. The scenario we aim at addressing in this paper is depicted in Figure 1. A swarm of drones is given the task of tracking a specific path while capturing videos with onboard cameras and send those videos to the GCS using a communication network which, in most cases, is the public Internet. Notice that, in order to improve situational awareness, we specifically want to achieve two main goals. Firstly, we want to *maximize the area covered* by the drones along the desired path. Notice that, as shown in Figure 1, the swarm of drones should be orchestrated in such a way

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART", CUP: D93C22000910001) and by the AGREED project (ARS01_0025) funded by the Italian Ministry of University and Research. The authors are with the Dipartimento di Ingegneria Elettrica e dell'Informazione at Politecnico di Bari, Via Orabona 4, 70125, Bari, Italy Emails: name.surname@poliba.it.

to ensure that the field of view of each drone overlaps of a given amount with that of its neighboring drones. This deliberate overlap allows executing *dynamic stitching* of received videos at the GCS, enabling comprehensive coverage and integration of the multiple perspectives shot by each drone. Secondly, even though the literature is abundant on coverage maximization, we point out that the impact of the communication network time-varying bandwidth on the *quality* of videos received by the GCS has only been tangentially studied in the literature, especially in the context of swarms of drones. In fact, the communication network bandwidth limits the rate at which each video source can be compressed and sent to the GCS. In practice, the lower the network bandwidth, the lower the encoding bitrate employed to compress the video. This results in decreased quality of received videos, consequently impacting the accuracy of computer vision tasks based on said videos. Nevertheless, consider that the available network bandwidth changes both in space and time unpredictably and, as such, it can be modeled as a disturbance. One way to improve the quality of the received videos for computer vision tasks is to increase the pixel density when the network bandwidth is scarce. The strategy that we advocate in this paper to address this challenge involves deviating from the conventional assumption of uniform drone altitudes. In particular, lowering a drone's altitude results in higher pixel density and, consequently, greater scene detail capture, thereby improving the quality of video data for computer vision analysis. Nevertheless, the naive approach of controlling the drones to be as close as possible to the ground conflicts with the first goal, i.e., the maximization of the covered area. Enabling drones to dynamically adjust their altitude in response to fluctuations in network bandwidth introduces variability in the geometry of captured images, complicating the task of ensuring consistent overlap in the field of view among drones within the swarm. Consequently, this variability poses challenges to the video stitching process, potentially impairing its effectiveness.

The control methodology adopted to tackle the aforementioned issues is a partially distributed NMPC framework. NMPC is the natural choice when the control problem involves constraints such as limitations in the actuators, energy consumption, coverage, etc. In addition, MPC controllers have been widely used for a variety of key applications such as collision avoidance, navigation, consensus, and formation control problems.

To implement a realistic scenario, each drone is subject to limited control inputs and limited changes in the control inputs. The first constraint is necessary to obtain both realistic and low energy consumption. Regarding the second constraint, limitations in acceleration guarantee a profile of the velocity control inputs that can be applied in reality.

II. Related Work

UAV surveillance and area coverage, along with video streaming is an active area of research which has attracted the attention of several research communities such as Artificial Intelligence (AI), Graph Theory, Multimedia Systems, etc. In the following, we provide a brief review of the relevant literature showing the gaps we aim to fill with this work.

An algorithm named Swarm-based UAV (SUAV) is proposed in [10]. Such an algorithm is used to detect mobile objects in natural disasters, such as floods, based on the swarm movement of UAVs. The paper proposes a swarm-based energy-aware solution for tracking mobile objects in a UAV disaster scenario. The proposed technique takes into account the visual quality of the acquired images using metrics quality such as SSIM and VQM as a function of drones' velocity. Maintaining multi-hop links between the air and ground to guarantee an improved visual quality is one of the goals of this paper. Nevertheless, the impact of the variable network available bandwidth has not been considered.

In [11], the cooperative exploration problem is addressed, employing a team of multiple UAVs to maintain line-of-sight (LOS) connections between ground stations and the UAV team, employing a relay UAV. The work considers collision avoidance, communication constraints, and feasible trajectories for each UAV, formulated as an optimization problem. Notably, differently from this work, the analysis does not include the implications of time-varying network available bandwidth.

Providing maximum coverage with minimum resource utilization is discussed in [12]. The paper takes into account the power consumption of drones to provide realistic coverage using an approach based on Particle Swarm Optimisation (PSO). Differently from our work, to perform the coverage task, the proposed solution does not use a collaborative approach based on multi-agent systems. Instead, the paper assumes that several drones simultaneously scan disjoint areas without direct communication among them.

In [13], the problem of surveillance of an area using UAVs is tackled using a camera mounted on each UAV with the goal of covering a wide area of interest. Although communication network-related aspects are considered, such as the impact of packet loss on visual quality, the effect of time-varying network available bandwidth is not considered.

In [14] authors focus on the communication between the UAVs and the ground station based on the bandwidth. Moreover, for coverage path planning, a sampling-based centroidal Voronoi diagram is employed by assigning each cell to a specific UAV for coverage within that cell without direct communication with other UAVs. Path planning is carried out for each drone in the given cell in the shortest time. Differently from [14], our research proposes a multi-agent system framework to both ensure coverage of an area of interest and a bandwidth-aware motion control with direct communication of the drones to avoid redundant movements for coverage and collisions.

Another approach related to ours was proposed in [15], which considers decentralized monitoring of an area using multiple drones. Drones are driven by distributed optimal control algorithms designed to minimize information loss. The main novelty of this paper is the ability to add or remove agents from the team while monitoring the area. Neverthe-

less, the paper does not use an active collision avoidance system for the agents. Compared to [15], a distinctive feature of our approach is that the orchestration of the swarm is performed in such a way to improve the quality of the videos captured by the cameras as a function of the network available bandwidth. In this work we assume that the path is given, but coverage path planning techniques such as [16] could be adapted for online path generation.

III. System design

This section presents the system design employed to achieve the goals described in Section I. In particular, after introducing the notation and the required definitions (Section III-A), the scenario and the proposed framework are outlined in Section III-B and Section III-C, respectively.

A. Preliminaries

Let us define $\|\cdot\|_p$ as the p norm of a vector and denote a *directed graph* (digraph) with $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, \dots, v_N\}$ is the set of nodes, also identified simply by its indices $i = 1, \dots, N$, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. An edge (v_i, v_j) denotes the information flow from node i to node j. Moreover, the set of neighbours of node v_i is defined as $\mathcal{N}_i = \{v_j \in \mathcal{V} : (v_j, v_i) \in \mathcal{E}\}$. A directed graph is said to be strongly connected if, for each couple of nodes i and j, there exists a directed path from i to j and vice versa. The adjacency matrix A associated with a graph represents the connections of the nodes through the edges. In other words, $A = (a_{ij})$ is such that $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. Let $D = D(G) = (d_{ij})$ be the in-degree matrix with $d_{ij} = \text{in-deg}(v_i)$ if $i = j$ and $d_{ij} = 0$ otherwise, where $\text{in-deg}(v_i)$ is the in-degree of node v_i, i.e., the number of edges pointing towards node v_i. Then, the Laplacian matrix can be defined as $L = D - A$. If G is strongly connected, $-L$ has eigenvalues such that $-\lambda_{N-1} \leq -\lambda_{N-2} \leq \cdots < -\lambda_0 \leq 0$, where $\lambda_0 = 0$ and $\text{rank}(L) = N - 1$.

Formation Control is a well-known method in multi-agent systems to keep the agents in a specific shape or distance. In this paper, such a methodology is used to keep a minimum safe distance among the agents while the shape is not fixed and can change over time. Thus, we aim at guaranteeing a *flexible* formation, i.e. a control framework that ensures safety distances among the agents without paying attention to the specific shape. On the other hand, it is required that the agents keep as close as possible to each other to allow a sufficient overlap of the areas scanned by their on-board cameras for stitching and object detection.

B. Scenario definition

In this subsection, we describe the scenario and the control goals to be achieved. Let us consider a swarm of drones equipped with cameras – such as the one depicted in Figure 1 – that are required to track a predefined path and monitor an area (e.g. a field, town, etc.) for a specific purpose such as firefighting, border patrolling, agricultural applications, surveillance, object detection, etc. The camera mounted on each drone is supposed to point vertically to the ground and scan a square area whose width depends on the drone's altitude. The video content captured by the cameras is then sent to a Ground Control Station (GCS) using a network connection such as, f.i., a 5G link. Such a station is in charge of receiving in real-time the video flows sent from all the agents and stitching together the frames to obtain an overall video scanning the area beneath the drones.

Stitching algorithms between two or more video frames usually require a certain amount of overlap, i.e. a scanned area in common, and high-quality videos [4], [17]. In fact, the first step of such algorithms is to perform feature detection in the frames, i.e. common feature points in the videos coming from different agents. For this purpose, the received videos need to be as high-quality as possible, otherwise, stitching techniques based on image features extraction might fail. In addition, since a certain overlap is required among videos, the drones should stick as close together as possible while avoiding collisions.

From a practical viewpoint, the drones are provided with an Internet connection to communicate with each other and with the base station to send the videos captured. Therefore, each drone is assumed to be able to exchange information with all the other drones according to a complete directed graph. Furthermore, in a realistic scenario, the available bandwidth depends on the drone's location in the area and on the number of users using the network bandwidth. Thus, the bandwidth is a variable not known a priori and modeled in this work as a measurable disturbance. As such, the available bandwidth can be measured indeed using both passive and active bandwidth estimation techniques [18], [19].

Due to the variable bandwidth, each drone sends the video with varying visual quality. For this reason, when the available bandwidth is low, it is desirable that the drone changes its altitude, getting closer to the ground, in order to capture more detailed frames and to allow video stitching at the ground station. On the contrary, when the bandwidth is high, the drone can increase its altitude to increase the coverage while at the same time maintaining a reasonable level of visual quality. A useful parameter that measures the size in meters of a pixel acquired by a camera is the *Ground Sample Distance* (GSD), which is a central feature in UAV-based photogrammetry [20].

At the same time, the swarm is supposed to keep a wide area coverage. To this end, it would be desirable that the drones fly as far as possible from each other but without any detachment among the corresponding scanned areas. In addition, to meet this specific goal, they would also fly at the maximum altitude in order to get the widest scanned area possible.

In the described scenario, each drone has two contrasting goals: i) provide a good percentage of overlap among the videos and the best visual quality for each video captured and ii) increase coverage of the region of interest. To achieve the first goal, each drone would keep a relatively low altitude, which consequently implies a decrease in the GSD and an increase in the visual quality. On the other hand, the

coverage is improved when the altitude is increased since the cameras mounted on the drones can capture a wider region. To meet both goals, we adopt the following trade-off: when the bandwidth is high, the drone can fly higher for coverage while still guaranteeing good resolution, and therefore a high level of visual quality. However, when the bandwidth is low, the drone goes down as previously explained. Therefore, as shown further on, a proper mapping between the altitude and the available bandwidth has to be defined in such a way that visual quality is minimally affected. Moreover, to survey the area, a leader drone is chosen to track a predefined path. The other agents are simply required to catch up with it while keeping a safe distance both from the leader and the other agents. Notice that the altitude of each drone changes as they move on according to the measured available network bandwidth.

C. Problem definition

Let us now introduce the NMPC to be executed at each drone. In particular, a two-state algorithm has been designed to avoid detachments, i.e. situations in which there is no overlap between the videos captured by the agents. Notice that avoiding this condition is crucial to allow video stitching. Then, for each of the two states, a specific cost function for the NMPC is defined. Let us also mention that a drone can communicate with other drones using the same Internet connection used to send videos to the GCS and that the bandwidth is assumed to be constant in the z-direction and time.

1) Drone Dynamics and Cost Functions: The discrete-time dynamics describing the evolution of the state for each drone is assumed to be a simple integrator for simplicity.

$$s_i(k+1) = s_i(k) + T_s u_i(k), \qquad i = 1, \dots, N \tag{1}$$

where N is the number of agents, $s_i = (x_i, y_i, z_i)^T$ is the vector of the spatial coordinates of agent i, T_s is sampling time, and $u_i = (u_{x,i}, u_{y,i}, u_{z,i})$ is the control input for agent i. In addition, the 3D orientation of the drone is not modelled in this paper since we assume video cameras are mounted on gimbals set to always point to the ground with a common yaw angle.

Equipped with this notation, we are now ready to introduce the proposed NMPC designed for the problem. The general structure is the following:

$$\begin{aligned}
\underset{u_i}{\text{Minimize}} \quad & \mathcal{J}_i && (2)\\
\text{subject to} \quad & s_i(k+1) = s_i(k) + T_s u_i(k) && (3)\\
& h_{\min} \le h_i \le h_{\max} && (4)\\
& \|X_i(t) - X_j(t)\|_2 \ge 2R_{\text{safe}} && (5)\\
& |u_i| \le u_{\max} && (6)\\
& |\Delta u_i| \le \Delta u_{\max} && (7)
\end{aligned}$$

where $X_i = (x_i, y_i)^T$ is the 2-D position of the i-th agent, h_i is the drone's altitude, and Δu_i denotes the variation in the speed between the current and the previous time-step. Eq. (4) is the constraint related to the altitude, which must be kept within a certain interval for safety constraints and can be considered as a *mission parameter*. Eq. (5) constrains the i-th agent to maintain a minimum distance ($2R_{\text{safe}}$) from any other agent. Finally, equations (6) and (7) are introduced to limit the control input and its variation over time to achieve realistic control efforts.

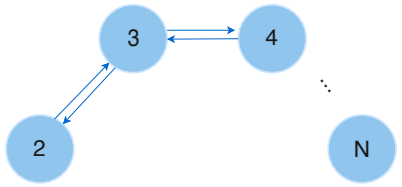


Fig. 2: Graph used for the computation of J_f

It is important to notice that the NMPC of the leader agent, denoted by $i = 1$, does not include constraint (5). This is because the primary objective of the leader is to track an assigned path. Consequently, ensuring a safe distance from the leader is up to the other drones.

Let us now describe the two-state Track-Recovery algorithm, in which the cost function $\mathcal{J}_i$ is defined according to the specific state and agent.

2) Track-Recovery Algorithm:
The swarm can assume one of two possible states: the *Tracking State*, and the *Recovery State*. In the Tracking state (T), there is minimal risk of detachment, characterized by sufficient overlap among the areas scanned by the cameras. Conversely, when in the Recovery state (S), there is a high risk of detachment, requiring a change in the cost function within the NMPC framework for each agent. During this state, we impose the leader to remain stationary. Only when adequate overlap is reestablished, the system transitions back to the Tracking State.

3) State T - Tracking:
In this state, the leader tracks a desired path while the other agents follow the leader, ensuring optimal coverage and video overlap within the swarm. Additionally, each agent is required to track a reference altitude dynamically adjusted based on the measured network available bandwidth.

For this state, we can define the cost function of the leader, i.e. $\mathcal{J}_{1,T}$, for problem (2)-(7) without the constraint (5) as already explained. Notice that, to avoid duplication, the equations below are written using the generic index i. Nevertheless, when we refer to the leader's cost function, it is intended $i = 1$.

$$\mathcal{J}_{1,T} = w_p J_p + w_h J_h + w_u J_u + w_{\Delta u} J_{\Delta u}, \tag{8}$$

$$J_p = \sum_{k=1}^{N_p} \left\| X_i(t+k|t) - \bar{X}(t+k) \right\|_\infty \tag{9}$$

$$J_h = \sum_{k=1}^{N_p} \left\| h_i(t+k|t) - \bar{h}_i(t+k) \right\|_2 \tag{10}$$

$$J_u = \sum_{k=1}^{N_c} \|u_i(t+k-1)\|_2 \tag{11}$$

$$J_{\Delta u} = \sum_{k=1}^{N_c} \|\Delta u_i(t+k-1)\|_2 \tag{12}$$

J_p is the component of the cost function concerning the path tracking and in particular $X_1(t)$ is the 2-D position of the leader at time t, N_p is the prediction horizon for the NMPC controller and $\bar{X}(t)$ is the x and y path reference at time t. J_h is responsible for altitude tracking, with $h_1(t)$ the leader's altitude and $\bar{h}_1(t)$ the leader's reference altitude at time t. J_u and $J_{\Delta u}$ limit the control effort and the variation in the control input. N_c is the control horizon for the NMPC controller.

The cost function $\mathcal{J}_{i,T}$, for $i = 2, \ldots, N$ used in the NMPC of the other agents for the state T is defined below.

$$\mathcal{J}_{i,T} = w_l J_l + w_h J_h + w_u J_u + w_{\Delta u} J_{\Delta u}, \tag{13}$$

$$J_l = \sum_{k=1}^{N_p} \left(\|X_i(t+k|t) - X_1(t+k)\|_\infty - r(R_i+R_1)\right)^2 \tag{14}$$

where J_l encourages agent i to follow the leader.

4) State S - Recovery:

There are cases where the swarm might be close to losing formation, critical to ensure video overlap among agents. Such situations can lead to possible *detachments*, which define the cases in which at least one video does not overlap with any other video of the swarm. Since each agent knows the location of all the other agents, they can detect such a risk, thus switching to the *Recovery* state (S). In this state, the leader halts and waits for the other agents to properly reassemble. Once the agents have reestablished proper alignment and adjusted their altitudes, the system can transition back to the Tracking state (T) to resume normal operation.

The cost function for the leader in this case prevents the leader from following the path and makes it wait for the others to catch up. In the meantime, the leader adjusts the altitude based on the reference altitude and also penalizes the control input and its changes. Such a cost function is defined as follows.

$$\mathcal{J}_{1,S} = w_h J_h + w_u J_u + w_{\Delta u} J_{\Delta u} \tag{15}$$

In this case, the cost function for the other agents does not penalize the control input and its changing rate to use maximum control effort to recover as soon as possible, while ensuring the constraints on control inputs and the variation of control inputs are met.

Moreover, it is also required that agent i keeps close to the other agents, too. To this end, considering a complete directed graph, as we have done so far, would cause issues with the altitude tracking or the leader following; therefore, a graph with fewer edges is employed. In particular, we consider a topology without the leader like the one shown in Figure 2, in which each agent communicates only with one or two other agents. For this new graph, let $\mathcal{N}_i = \{j \in \mathcal{V} : (j,i) \in \mathcal{E}\}$. At this point, we can identify J_f as the component of the cost function that keeps the agents together and at the same time encourages a certain overlap of the scanning areas through the term $r(R_i + R_j)$. R_i and R_j are the coverage area radii for drones i and j, respectively. J_f is defined as follows:

$$J_f = \sum_{j \in \mathcal{N}_i} \Big(\sum_{k=1}^{N_p} \big(\|X_i(t+k|t) - X_j(t+k)\|_\infty - r(R_i+R_j)^2\big)\Big) \tag{16}$$

Using simple geometrical arguments, the relationship between R_i and h_i is given by $R_i = h_i \tan(\phi/2)$, where ϕ is the angular aperture of the video camera [21]. Furthermore, r determines the desired amount of overlap. Setting a value of r close to 1 implies a relatively small percentage of overlap. On the contrary, decreasing r pushes the agents closer to one another. It is important to highlight that the actual communication graph is a complete directed graph since every agent can communicate with any other agent through the Internet connection. Only for J_f, a subgraph with fewer edges is employed to solve issues related to the agents stuck at the lowest altitude and to avoid redundant communication among the agents. Notice that considering different graphs according to the particular task to be completed is an approach known in the literature, as proposed in [22], where three topologies are introduced: Sensing Topology, Communication Topology, and Actuation Topology. Each of them is employed to accomplish different goals.

The cost function for the other agents, i.e. for $i = 2, \ldots, N$, is thus defined as follows:

$$\mathcal{J}_{i,S} = w_l J_l + w_h J_h + w_f J_f \tag{17}$$

Finally, $w_p, w_l, w_f, w_h, w_u, w_{\Delta u}$ are weights to be properly tuned to obtain the best performance possible. Such a tuning can be performed empirically or using parameter optimization tools.

Therefore, according to the type of the agent and state, the cost function in (2) can be $\mathcal{J}_{1,T}, \mathcal{J}_{i,T}, \mathcal{J}_{1,R}$ or $\mathcal{J}_{i,R}$, $i = 2, \ldots, N$.

5) Correlation Between Bandwidth and altitude: To determine $\bar{h}_i$ in (10) for each agent, we use the following monotonically increasing function to build a relationship between the estimated available bandwidth b and $\bar{h}_i$, based on the arguments given in Section III-B.

$$\bar{h}_i(b) = \begin{cases} h_{\min} & b < b_{\min} \\ S \cdot b + h_{\max} - S \cdot b_{\max} & b_{\min} \le b \le b_{\max} \\ h_{\max} & b > b_{\max} \end{cases} \tag{18}$$

where S is defined as $S = (h_{\max} - h_{\min})/(b_{\max} - b_{\min})$.

Notice that this function could be replaced by any other mapping able to describe the relationship between altitude, visual quality, and available bandwidth more effectively. The

TABLE I: Values of the weights in the cost functions of the NMPCs

	w_l	w_p	w_f	w_h	w_u	$w_{\Delta u}$
$\mathcal{J}_{1,T}$	1.5	-	-	5	0.001	0.001
$\mathcal{J}_{i,T}$	-	0.5	-	1.5	0.001	0.001
$\mathcal{J}_{1,S}$	-	-	-	0.5	0.001	0.001
$\mathcal{J}_{i,S}$	1.5	-	0.3	1.5	-	-

identification of such a mapping is beyond the scope of this paper and is left as a future work.

IV. Simulations

In this section, simulation results are provided to validate the effectiveness of the proposed approach. Simulations were run for different values of some parameters involved. In particular, we discuss the results obtained for $N \in \{3, 5, 6\}$, where N is the number of agents, and for three possible predefined paths: a sine-like, a semicircle and a generic shape, which aims at simulating a road. Moreover, we have set the values for the weights in the cost functions previously introduced as in Table I.

Then, to run the NMPC, we set $r = 0.4$, $N_p = 5$, $N_c = 2$, $R_{\text{safe}} = 2$, $h_{\min} = 4$, and $h_{\max} = 10$. Finally, while the upper and lower bounds for $\Delta u_x, \Delta u_y, \Delta u_z$ have been set to an absolute value of 0.1 for all agents. Notably, the velocity of the leader is intentionally set lower than that of the follower agents. This allows the swarm to maintain formation, ensuring that the followers can keep pace with the leader. For this reason, $u_{x,\max} = 0.7, u_{y,\max} = 0.7$ in (6) for the leader whereas $u_{x,\max} = 1, u_{y,\max} = 1$ for the other agents. Notice that the effect of the maximum velocity of the leader is further studied in the next subsection. Regarding $u_{z,\max}$, it has been set to 0.1 for all the agents to avoid abrupt changes in altitude and, therefore, in terms of covered area.

Concerning the network available bandwidth, we set it by drawing from a uniform random distribution of bandwidth values over the area to be scanned. Such values are then variable in the space but assumed to be constant in time and altitude. This distribution has been employed to run all the following simulations for fair comparisons and considerations. Notice that the choice of generating random bandwidth between a minimum value of 500 kbit/s and a maximum value of 5000 kbit/s defines a realistic bandwidth profile due to its unpredictability. Moreover, to the best of our knowledge, no dataset providing measurements of wireless channels as a function of the geographical position of a drone is available.

A. Overall results

Let us start by considering the case $N = 3$, where the leader is required to track a sine-like path. The drone's trajectories and the area covered by their cameras is shown in Figure 3. From the figure, it is possible to state that the drones, whose initial position is denoted with round markers, strive to follow the leader in the space while sticking together. On the x, y-plane, the reference path for the leader (black dashed line) and the leader's trajectory projection on this plane (light blue line) are depicted, thus showing good performance in terms of tracking. Moreover, this plane

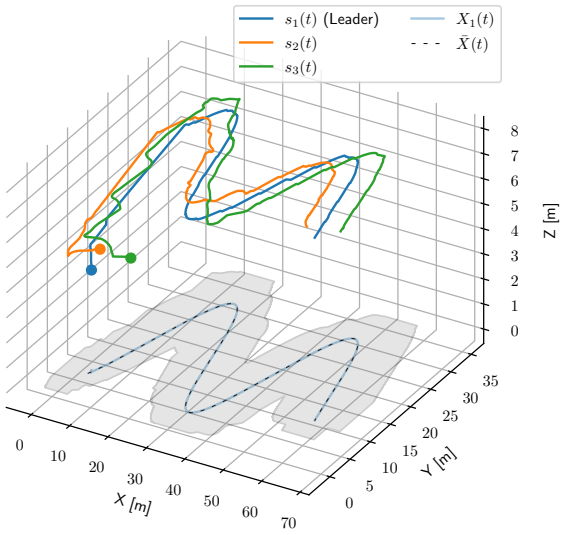


Fig. 3: Drones' trajectories and area covered for $N = 3$

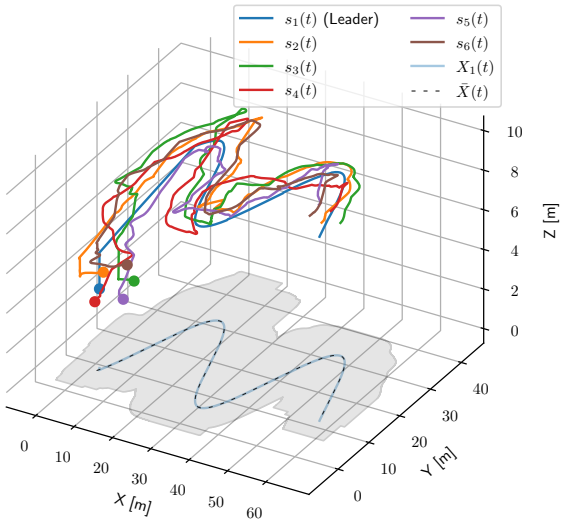


Fig. 4: Drones' trajectories and area covered for $N = 6$

also displays in gray the total area scanned by the swarm. The z-axis represents the altitude of the drones, which is shown to change along the path due to its dependence on the bandwidth. To have a visual comparison, consider Figure 4, which is obtained in the case of $N = 6$ and the reference path is the sine-like signal. The trajectories are close to each other, guaranteeing overlap while always avoiding collisions. Moreover, it is possible to see that the swarm is able to cover a considerably wider area around the reference path. Similar results have been observed for the case of $N = 5$ agents and for the other two considered paths.

Figure 5 shows the dynamics of the tracking error of the altitude for the case of $N = 6$ agents. In other words, it can be seen that, starting from the same initial altitude, each agent is able to track its reference altitude almost perfectly after a transient. Notice that, as already explained, each agent has a different reference altitude in that it depends on the particular position of the agent at each time-step and on the bandwidth available in that point. Similar considerations can be made for the other cases.

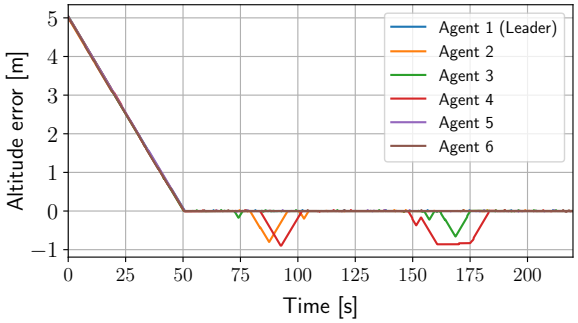


Fig. 5: Tracking error of the altitude for $N = 6$

As expected, increasing the number of agents leads to an increased total scanned area. In Table II, the total coverage areas produced by the agents for $N \in \{3, 5, 6\}$ are listed. Notice that, for each reference path, such areas are normalized *wrt* the area obtained when $N = 3$. As expected, the data show a growing trend and the scanned areas experience an increase of about 30 % when passing from $N = 3$ to $N = 6$ for all the three paths.

B. In-depth analysis

Let us now carry out a sensitivity analysis for the case of $N = 5$ agents tracking a sine-like path[1]. To evaluate the efficiency in terms of detachment avoidance, we test the framework by varying the values of $u_{x,\max}$ and $u_{y,\max}$ of the leader from the set $\{0.7, 0.8, 0.95\}$. Subsequently, we are interested in evaluating the time spent by the swarm in the *Recovery* state S, i.e. when the leader stops tracking the desired path to wait for the other agents to catch up and regroup. The higher this amount of time, the more the efficiency of the framework deteriorates since more time is required to the swarm to complete the task. To the purpose, we define the following *efficiency index*:

$$\eta = 1 - \frac{T_S}{T_{\text{tot}}} \tag{19}$$

where T_S is the amount of time spent in *Recovery* mode while T_{tot} is the time spent to accomplish the whole mission. Therefore, a higher η implies a better performance and a low amount of fraction of time spent in state S.

Results show that, when the leader's maximum velocity is set to 0.7, the measured efficiency η is approximately 0.9. Increasing the velocity to 0.8 results in lowering the efficiency to 0.86, and further increasing it to 0.95 reduces η to 0.82.

This observation suggests, as expected, that allowing the leader a higher velocity to track the reference path increases the likelihood of switching to the *Recovery* state.

In other words, the leader will stop more times to wait for the rest of the swarm. Notice that, on the other hand, making the velocity of the leader too low would lead to tracking issues in that the number of samples of the reference path should be increased and the total time to complete the task-even without switching to the *Recovery* state-would increase, too. Therefore, finding an optimal balance in the leader's velocity is crucial for maintaining efficient swarm operation and minimizing the need for recovery interventions.

[1]Similar conclusions can be drawn considering different desired paths or number of drones.

TABLE II: Normalized total scanned area as a function of N for each considered scenario

	$N = 3$	$N = 5$	$N = 6$
Sine	1.0	1.19	1.37
Road	1.0	1.17	1.28
SemiCircle	1.0	1.26	1.36

TABLE III: Efficiency η as a function of N for each considered scenario

	$N = 3$	$N = 5$	$N = 6$
Sine	0.94	0.87	0.91
Road	0.90	0.85	0.83
SemiCircle	0.91	0.87	0.88

With this in mind, let us further investigate what happens when the swarm switches from the *Tracking* to the *Recovery* state and vice versa. During the simulation with $N = 5$ agents tracking a sine-like path, one of the cases when the swarm enters the *Recovery* state is in the interval 86-97 seconds. This period is depicted by the shaded gray area in Figure 6. This figure shows the dynamics of the x and y coordinates of the agents. In the shaded area, that is, when the swarm is in state S, it can be observed that the coordinates of the leader do not change over time, i.e. the leader has stopped to allow the swarm to regroup.

By looking at the coordinate dynamics, it can be seen that, when in *Recovery* state, they are rather far from each other, whereas before or after this switch the coordinates present a converging trend. To better understand this concept, Figure 7 provides a visual representation of the areas scanned by the drones when in *Recovery* state (dashed squares) and in *Tracking* state (solid-lined squares). In the *Recovery* state, the agents are not close enough to each other, increasing the risk of detachment. Therefore, the leader, represented in the figure as a blue square tracking the dashed reference path, does not move until the scanned area of each agent has

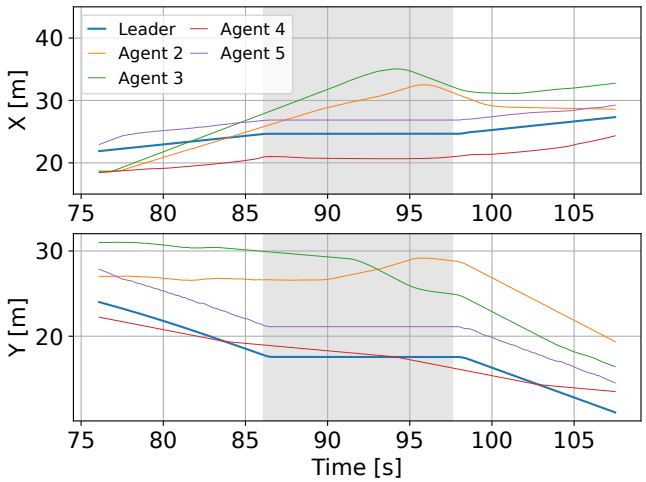


Fig. 6: Agents' coordinates dynamics in the case of a switch between *Tracking* and *Recovery* state

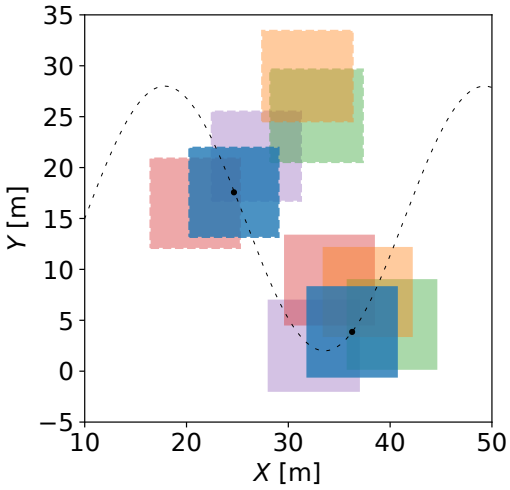


Fig. 7: Areas scanned by the agents at two different time-steps

enough overlap. The solid-lined squares show an example of swarm in the *Tracking* state, thus without risk of detachment, which presents a considerably higher total overlap.

Similar conclusions could be drawn for the case of a different number of agents or a different reference path. Finally, Table III reports the measured efficiency index for all the considered cases. In general, when the number of agents increases, the efficiency tends to decrease. This is to be expected since, when the number of agents becomes large, it is harder to move together, thus increasing the possibility of detachment as well as the value of T_S.

V. Conclusions

A decentralized NMPC-based control framework for a swarm of drones has been proposed to monitor an area through onboard cameras that capture videos to be sent in real-time to a ground station. To send such videos at the highest possible quality, the available bandwidth and the drone's altitude are taken into account. The proposed method leverages leader-follower multi-agent system formation control to keep the drones close to each other without collisions to provide overlapped videos suitable for stitching at the Ground Control Station. Meanwhile, the leader tracks a given desired path. To this end, the agents employ NMPC to optimize a suitable cost function to achieve all the aforementioned goals. In particular, the algorithm consists of two states named *Tracking* and *Recovery*. These states are developed to avoid any potential detachments among the areas scanned by the agents. In each state, different cost functions are fed to the agents. Results show the effectiveness of the proposed approach, which ensures video overlap for stitching, path and altitude tracking for video quality and area coverage.

References

[1] T. M. Cabreira, C. D. Franco, P. R. Ferreira, and G. C. Buttazzo, "Energy-Aware Spiral Coverage Path Planning for UAV Photogrammetric Applications," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3662–3668, 2018.

[2] S. Asadzadeh, W. J. de Oliveira, and C. R. de Souza Filho, "Uav-based remote sensing for the petroleum industry and environmental monitoring: State-of-the-art and perspectives," *Journal of Petroleum Science and Engineering*, vol. 208, p. 109633, 2022.

[3] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Computer Communications*, vol. 156, pp. 1–10, 2020.

[4] X. Meng, W. Wang, and B. Leong, "Skystitch: A cooperative multi-uav-based real-time video surveillance system with stitching," in *Proceedings of the 23rd ACM international conference on Multimedia*, 2015, pp. 261–270.

[5] S. P. H. Boroujeni and A. Razi, "Ic-gan: An improved conditional generative adversarial network for rgb-to-ir image translation with applications to forest fire monitoring," *Expert Systems with Applications*, p. 121962, 2023.

[6] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 10, pp. 1627–1643, 2021.

[7] F. Mehdifar, C. P. Bechlioulis, F. Hashemzadeh, and M. Baradarannia, "Prescribed performance distance-based formation control of multi-agent systems," *Automatica*, vol. 119, p. 109086, 2020.

[8] M. A. Rezaei, P. Bagheri, and F. Hashemzadeh, "Consensus tracking of multi-agent systems in the presence of byzantine agents using model predictive control," in *2022 8th International Conference on Control, Instrumentation and Automation (ICCIA)*, 2022, pp. 1–5.

[9] ——, "Predictive consensus tracking of multi-agent systems in the presence of byzantine agents and connection loss of reference signals," *Optimal Control Applications and Methods*, vol. 45, no. 2, pp. 842–854, 2024.

[10] I. Medeiros, A. Boukerche, and E. Cerqueira, "Swarm-based and energy-aware unmanned aerial vehicle system for video delivery of mobile objects," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 766–779, 2021.

[11] H. Jiang, Y. Chang, L. Yang, X. Liu, and Y. He, "Cooperative Exploration of Heterogeneous UAVs in Mountainous Environments by Constructing Steady Communication," *IEEE Robotics and Automation Letters*, 2023.

[12] H. S. Munawar, A. W. Hammad, and S. T. Waller, "Disaster Region Coverage Using Drones: Maximum Area Coverage and Minimum Resource Utilisation," *Drones*, vol. 6, no. 4, p. 96, 2022.

[13] A. Bist and C. Singhal, "Efficient immersive surveillance of inaccessible regions using uav network," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–6.

[14] U. Choi and S. Lee, "Bandwidth-aware coverage path planning for swarm of uavs with aerial base station," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 360–365.

[15] M. Schwager, B. J. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of the IEEE*, vol. 99, no. 9, pp. 1541–1561, 2011.

[16] A. Tahirovic and A. Astolfi, "A convergent solution to the multi-vehicle coverage problem," in *2013 American Control Conference*. IEEE, 2013, pp. 4635–4641.

[17] L. Wei, Z. Zhong, C. Lang, and Z. Yi, "A survey on image and video stitching," *Virtual Reality & Intelligent Hardware*, vol. 1, no. 1, pp. 55–83, 2019.

[18] R. Prasad, C. Dovrolis, M. Murray, and K. Claffy, "Bandwidth estimation: metrics, measurement techniques, and tools," *IEEE network*, vol. 17, no. 6, pp. 27–35, 2003.

[19] G. Carlucci, L. De Cicco, S. Holmer, and S. Mascolo, "Congestion control for web real-time communication," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2629–2642, 2017.

[20] J. C. Leachtenauer and R. G. Driggers, *Surveillance and reconnaissance imaging systems: modeling and performance prediction*. Artech House, 2001.

[21] H. Huang, A. V. Savkin, and C. Huang, "Decentralized autonomous navigation of a uav network for road traffic monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 4, pp. 2558–2564, 2021.

[22] H.-S. Ahn, *Formation control*. Springer, 2020.