POLITECNICO DI TORINO
Repository ISTITUZIONALE

A Bioinspired Neural Network-Based Approach for Cooperative Coverage Planning of UAVs

*Original*
A Bioinspired Neural Network-Based Approach for Cooperative Coverage Planning of UAVs / Godio, S., Primatesta, S., Guglieri, G., Dovis, Fabio.. - In: INFORMATION. - ISSN 2078-2489. - ELETTRONICO. - 12:2(2021), pp. 1-17. [10.3390/info12020051]

*Availability:*
This version is available at: 11583/2866578 since: 2021-01-25T10:34:30Z

*Publisher:*
MDPI

*Published*
DOI:10.3390/info12020051

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

05 September 2026

information

*Article*

# A Bioinspired Neural Network-Based Approach for Cooperative Coverage Planning of UAVs

**Simone Godio [1,*], Stefano Primatesta [1], Giorgio Guglieri [1] and Fabio Dovis [2]**

[1] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, 10129 Torino, Italy; stefano.primatesta@polito.it (S.P.); giorgio.guglieri@polito.it (G.G.)
[2] Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Torino, Italy; fabio.dovis@polito.it
* Correspondence: simone.godio@polito.it; Tel.: +39-348-911-5652

**Abstract:** This paper describes a bioinspired neural-network-based approach to solve a coverage planning problem for a fleet of Unmanned Aerial Vehicles exploring critical areas. The main goal is to fully cover the map, maintaining a uniform distribution of the fleet on the map, and avoiding collisions between vehicles and other obstacles. This specific task is suitable for surveillance applications, where the uniform distribution of the fleet in the map permits them to reach any position on the map as fast as possible in emergency scenarios. To solve this problem, a bioinspired neural network structure is adopted. Specifically, the neural network consists of a grid of neurons, where each neuron has a local cost and has a local connection only with neighbor neurons. The cost of each neuron influences the cost of its neighbors, generating an attractive contribution to unvisited neurons. We introduce several controls and precautions to minimize the risk of collisions and optimize coverage planning. Then, preliminary simulations are performed in different scenarios by testing the algorithm in four maps and with fleets consisting of 3 to 10 vehicles. Results confirm the ability of the proposed approach to manage and coordinate the fleet providing the full coverage of the map in every tested scenario, avoiding collisions between vehicles, and uniformly distributing the fleet on the map.

**Keywords:** unmanned aerial vehicle (UAV); autonomous navigation; coverage planning; fleet coordination

check for updates

**Citation:** Godio, S.; Primatesta, S.; Guglieri, G.; Dovis, F. A Bioinspired Neural Network-Based Approach for Cooperative Coverage Planning of UAVs. *Information* **2021**, *12*, 51. https://doi.org/10.3390/info12020051

Academic Editor: Hakim Ghazzai
Received: 6 December 2020
Accepted: 20 January 2021
Published: 25 January 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

Autonomous exploration with mobile robots is a widespread problem in robotics [1]. Even if this topic has been widely studied since the last decade [2], there are still some open problems, including coverage planning [3]. The coverage planning problem attracted the attention of several researchers that studied this problem both with ground [4] and aerial robots [5]. However, the coverage planning problem becomes even more complex considering a multi-vehicle scenario, such as a fleet of Unmanned Aerial Vehicles (UAVs). The use of aerial robots for surveillance and exploration presents several complexities, both technical and legal. However, it shows considerable potential in terms of the size of areas monitored and tasks performed in a limited time.

Recent critical events, such as the earthquake in L'Aquila (Italy), the explosion in Tripoli (Lybia), and the hurricanes and typhoons in Asia, have shown how UAVs could be useful to provide surveillance, monitoring, and search and rescue applications. Thanks to their responsiveness and unique top-down point of view in urban environments, they have the potential of optimizing the search and rescue operations and, thus, saved lives [6]. Therefore, the importance of a fast and autonomous exploration approach with UAVs plays an essential role for those entities involved in surveillance and safety.

One of the first contributions on control and coordination of a fleet of autonomous robots was introduced in [7], whereas the author presents a method for multi-vehicle coordination using an incremental and distributed plan-merging process. Hence, after this

work, plenty of studies have been conducted for the state-of-the-art. In [8], the authors investigate the coverage planning problem using a fleet of autonomous robots for precision farming using a distributed strategy but neglecting obstacle avoidance between vehicles and other obstacles. In a similar scenario in [9], the coverage problem is solved by optimizing the power consumption. Other similar works [10,11] focus on the coordination of a fleet of UAVs by using Particle Swarm Optimization (PSO) to perform target tracking and obstacle avoidance. Recently, in [12], a cooperative path planning optimization is proposed to minimize the traveling distance. Works presented in [13–16] show different solutions to this problem with different approaches, but the assumed scenarios are simplified and very far from the real-world scenario.

An interesting approach was proposed in [17] introducing a preliminary model for fleet coordination in urban environments considering a distribution of docking stations. Another approach proposed in [18] solves the coverage planning problem by defining a series of waypoints for UAVs to explore maps. Instead, in [19], the authors propose an optimal solution based on a genetic algorithm solving the coverage planning problem, but considering a limited and fixed number of UAVs that is a strong limitation for a flexible coverage application. Differently, the work in [20] presents a solution to cover and explore areas affected by disasters. In this case, the coverage problem is simplified because the algorithm assigns a specific portion of the area to be monitored by each vehicle. Moreover, Ref. [21] investigates cooperative coverage techniques by splitting the operative area into cells for agricultural purposes without considering the presence of obstacles. Recently, the authors in [22] proposed the same approach again for monitoring wildfires zones. The work presented in [23] describes an alternative approach for robot exploration based on gradient optimization. The search is optimized to reach specific objectives, but not to explore and cover an entire area.

Recently, cooperative coverage planning is solved using a reinforcement learning approach. In [24], the authors present a promising solution. Anyway, due to the high complexity of the proposed approach, the analyzed scenario is simplified considering simple maps and a fleet of three agents. A more complex scenario is tackled in [25] using a Deep Reinforcement Learning-based approach in complex maps, but only considering a single agent.

*Current Work*

In this work, we propose a novel approach to solve the coverage planning problem. Specifically, we solve three sub-problems simultaneously: (i) coordinate a fleet of UAVs avoiding collisions between vehicles and other obstacles in the map, (ii) displace uniformly the fleet of UAV in the map, and (iii) fully cover the area by defining a sequence of targets. Unlike other works for the state-of-the-art, our approach is not limited to cover a specific area but evaluates a further constraint maintaining a uniform distribution of the fleet on the map. This feature is mandatory for surveillance applications, where the responsiveness of the fleet to reach any position on the map is an essential element for surveillance purposes. The uniform displacement of UAVs allows them to reach any position on the map as fast as possible. Furthermore, the algorithm is flexible and adaptable to a fleet with non-fixed dimensions. In fact, in our preliminary simulations, we use fleets consisting of 3 to 10 UAVs.

The proposed algorithm differs from most of the works previously mentioned for its focus on aerial surveillance applications. Specifically, the algorithm is designed for rotary-wing UAVs with high maneuverability and reduced flight speed. Nevertheless, the algorithm is flexible for more applications, considering different operational conditions and configurations for mobile robots.

Hence, to validate the proposed method, the algorithm is tested performing some preliminary simulations using MATLAB (Natick, MA, USA) and, then, using the Robotic Operating System (ROS) framework to execute a simulation in a more realistic virtual environment performed with Software In The Loop (SITL) and Gazebo frameworks.

We organized the paper as follows. Section 2 describes the analyzed coverage planning problem. In Section 3, we present the proposed approach with the mathematical model and the pseudocode. Section 4 shows the preliminary simulations and the numerical results, as well as the realistic simulation, performed using ROS, SITL, and Gazebo. Hence, in Section 5, we draw our conclusions.

## 2. Assumptions, Notation, and Problem Description

In this section, we describe the coverage planning problem considered in this work, defining the notation used and detailing the assumption considered.

As defined in the previous section, this work aims to solve a coverage planning problem to explore and monitor a specific area with a fleet of UAVs. Specifically, the goal is to cover the entire map using a fleet of UAVs maintaining, at the same time, a uniform distribution of the fleet on the map, as well as avoiding collision between vehicles and other obstacles.

First of all, in this work, we consider the following assumptions:

- The map is known as a priori. As a consequence, the dimension of the search space (i.e., the map) and the displacement of obstacles are known;
- The dimension of the fleet is always set before the coverage planning task starts. Anyway, the proposed approach is tested in different scenarios with fleets consisting of 3 to 10 vehicles;
- The map is considered fully covered when at least 99% of the map is visited.

The search space used by the algorithm to search for a solution is defined by a grid map with dimension $N \times M$. Hence, we assume a fleet of UAVs consisting of $D$ UAVs defined by the set $Z$. We denoted each UAV with $z_i \in Z$ with $i$ from 0 to $D-1$. All UAVs have the same configuration and the same field of the view (FOV) with dimension CS. The field of view is assumed as a constant parameter, without considering the variation of FOV caused by flight attitude.

In particular, in this work, we analyze four different environments with increasing complexity (called Field1, Field2, Field3, and Field4) in terms of the density of obstacles and their distribution in the map. These maps do not represent a real area. However, as shown in Figure 1, they have features similar to a realistic urban environment. In real applications, maps can be reconstructed using satellite or aerial imagery [26,27].

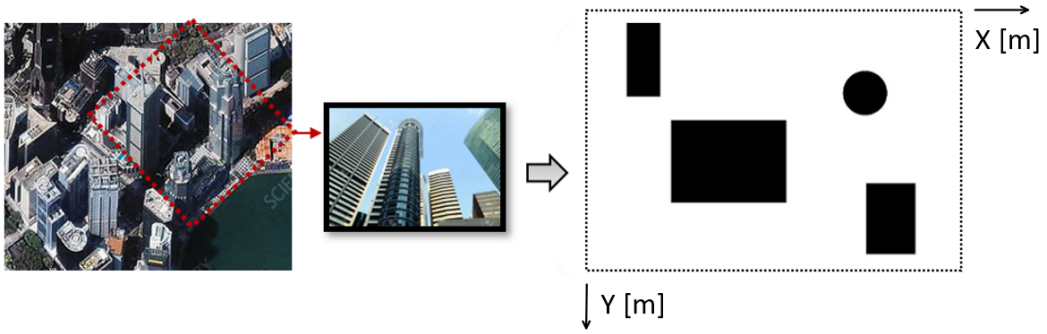


**Figure 1.** Example of a reconstruction of a simplified two-dimensional map starting from a urban image.

In most of the scenarios assumed in this work, the initial starting condition is with all the UAVs positioned in the upper left corner of the map, as shown in Figure 2.

This initial configuration is set to have the same initial condition in each simulation as well as to simulate a more realistic condition where the entire fleet is deployed from a circumscribed starting zone. An initial configuration with UAVs already uniformly distributed in the map would have some benefits on the performance of the coverage planning since it is an optimal starting condition obtaining a full coverage of the map in less time with fewer moves.

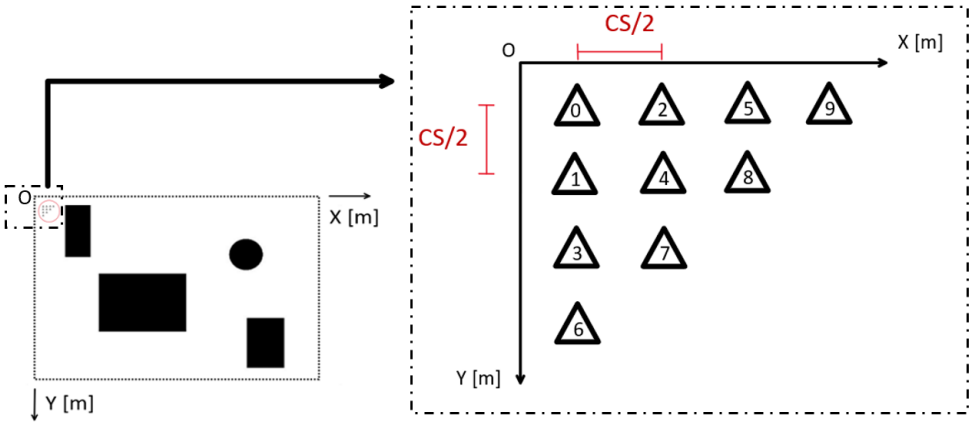


**Figure 2.** The initial configuration of the fleet of UAVs. This initial condition is assumed in most of the simulations performed in this work.

## 3. Proposed Approach

In this section, we present the proposed approach to provide cooperative coverage planning with a fleet of UAVs. The proposed method uses a bio-inspired neural network, inspired by the method proposed for the first time in [28]. The neural network is displaced as a grid of neurons, whereby the dynamics of each neuron depends on the proximity of unvisited neurons and, as a consequence, unvisited areas. Unlike traditional neural network approaches, this method does not require a training phase, since it is based on the propagation of neuron dynamics from unvisited areas in all the map, to guide vehicles toward unexplored locations.

However, the proposed approach differs from the original method proposed in [28]. First, in our work, the dynamics of each neuron are subject to unvisited areas, the presence of obstacles, and the position of each UAV of the fleet. Moreover, the dynamics of an unvisited neuron are not propagated to all neurons (i.e., to the entire map), but the propagation is guided toward UAVs avoiding evaluating useless neurons in the grid map and, as a consequence, reducing the time complexity.

Recalling the problem defined in Section 2, the method refers to a grid map that corresponds to the neural network, whereby each element of the map is a neuron. Each neuron has only local connections with neighbors, as depicted in Figure 3c.

As defined in Section 2, each UAV has a field of view with dimension CS. Therefore, the distance between neurons is defined as $CS/2+1$ to cover all the area around the selected neuron. Therefore, as shown in Figure 3c, the adjacent neuron in the grid is $CS/2+1$ away.

The main idea of the proposed approach is to define a sequence of movement on the map. Specifically, at each time step $t$, a UAV $z_i$, is located on a position defined by the neuron $x_n^{z_i}$. Then, the algorithm defines a move toward an adjacent neuron $x_{n+1}^{z_i}$ that maximises a function $f(x)$

$$x_{n+1}^{z_i} = \arg \max_{x_{nb} \in X(x_n^{z_i})} f(x_{nb}) \quad (1)$$

$$\text{subject to} \quad \sigma(x_n^{z_i}, x_{n+1}^{z_i}) \notin O \quad (2)$$

$$\sigma(x_n^{z_i}, x_{n+1}^{z_i}) \cap \sigma(x_n^{z_j}, x_{n+1}^{z_j}) = \emptyset \quad \forall \ 0 \geq i > D \ \wedge \ i \neq j \quad (3)$$

$$x_{n+1}^{z_i} \neq x_{n+1}^{z_j} \quad \forall \ 0 \geq i > D \ \wedge \ i \neq j \quad (4)$$

with $X(x_n^{z_i})$ is the set of neighbor neurons of the neuron $x_n^{z_i}$ where the drone $z_i$ is located. Practically, the neighbor neurons are the eight neurons of Figure 3c.

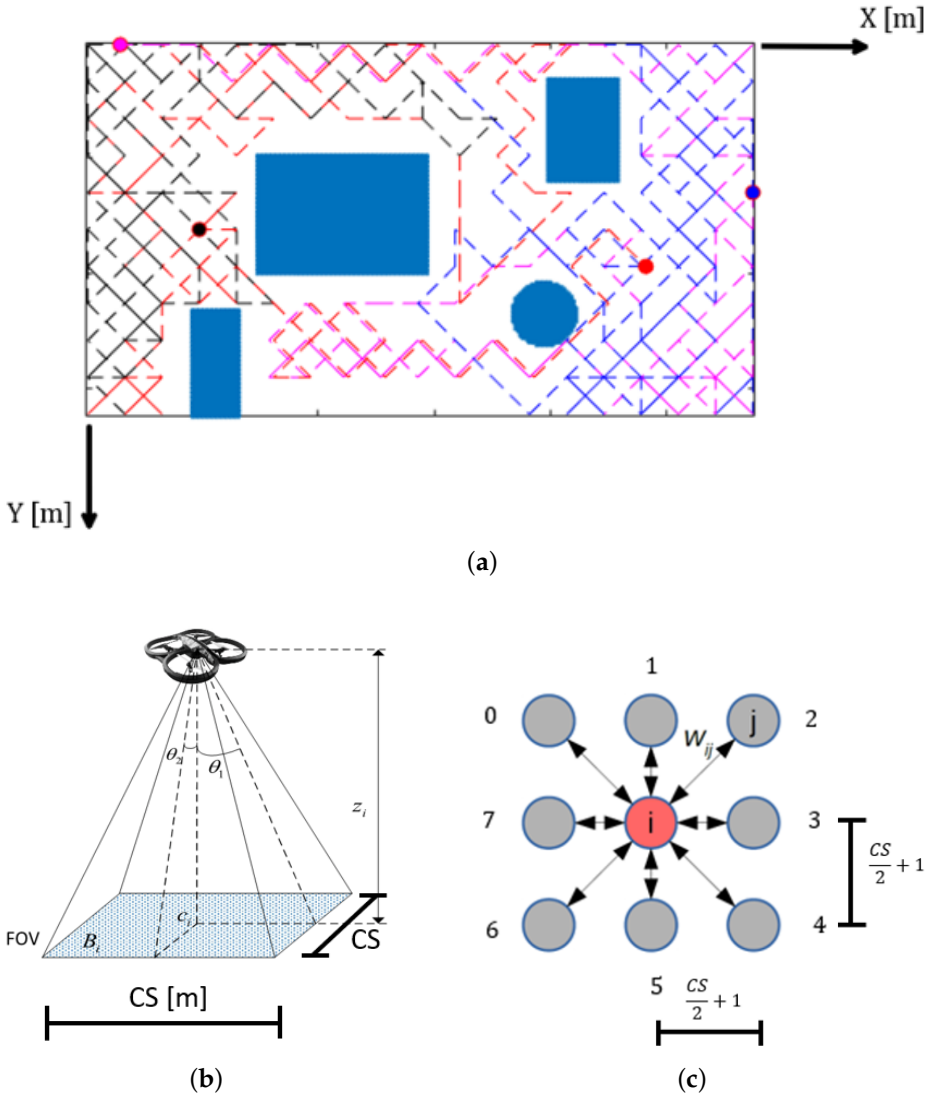


**Figure 3.** In (**a**) an example of the cooperative coverage planning performed in this work. In (**b**), the square Field of View (FOV) of the UAV assumed in this work. In (**c**), a small bio-inspired neuronal network grid as defined in our approach.

The constraints (2), (3) and (4) provide the collision avoidance. Specifically, Equation (2) provides the obstacle avoidance by checking if the motion segment $\sigma(\cdot)$ from the current neuron to the next one is outside the obstacle set $O$ containing all the elements of the map that cause a collision with obstacles. Equation (3) verifies if there is an intersection between the segment computed by the UAV $z_i$ and other segments already computed for other UAVs of the fleet. On the other hand, Equation (4) checks if the neuron $x_{n+1}$ is not already selected by other UAVs. Moreover, as shown in Figure 4, the segment $\sigma(\cdot)$ evaluates a safety corridor-wide $d_{lim}$ to take into consideration the occupation of the vehicle during the motion, as well as a safety distance.

Hence, the function $f(x)$ is defined as follows:

$$f(x) = C[x] + \frac{d_{avg}}{d_{max}} w_{md} + \frac{vr_{dist}}{vr_{max}} w_{vr} + (1 - \frac{\Delta\theta}{\pi}) \tag{5}$$

consisting of four main elements that consider an attractive cost toward uncovered areas ($C[x]$), the mean distance between UAVs ($\frac{d_{avg}}{d_{max}} w_{md}$), the standard deviation of the distri-

bution of the fleet ($\frac{vr_{dist}}{vr_{max}} w_{vr}$), and the cumulative turn angle ($1 - \frac{\Delta\theta}{\pi}$). These elements are detailed in the following paragraphs.

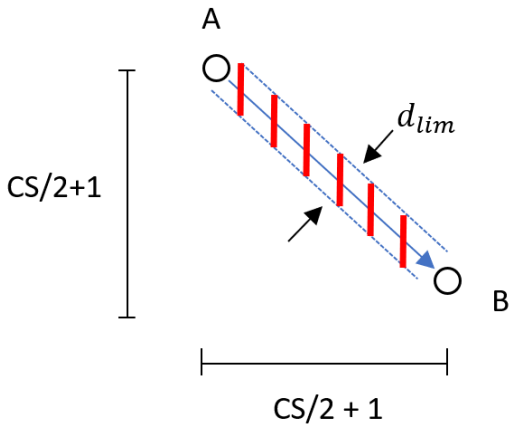


**Figure 4.** Graphical representation of the safety corridor assumed in the collision avoidance constraint. In our simulations, we use $d_{lim} = 4$ m.

The first element $C[\cdot]$ is a cost-map matrix with the same dimension of the map containing the attractive costs due to uncovered areas. Specifically, $C[x]$ represents the cost in correspondence with the neuron $x$. We compute the attractive contribution for each uncovered neurons and, then, is propagated toward UAVs. Therefore, we compute the attractive cost at the uncovered neuron as follows:

$$C[x_n] = C[x_n] + \frac{\|x_n - x_c\|_2}{d_{max}} B_k. \tag{6}$$

The norm $\|x - x_c\|_2$ is the Euclidean distance between the current neuron $x$ and the neuron $x_c$ that corresponds to the closest UAV. The term $d_{max}$ is the maximum admissible distance in the map, i.e., the maximum diagonal of the map:

$$d_{max} = \sqrt{N^2 + M^2}. \tag{7}$$

This normalization considers the maximum distance in the neural grid and allows us to manage the weights of Equation (6) more efficiently.

In this phase, it is essential to evaluate each neuron once. For this purpose, the parameter $B_k$ is used to define the cost, assigned only if a flag $g(x)$ ensures that the neuron is not already evaluated, i.e., if $g(x)$ is equal to 0. Hence:

$$\begin{cases} B_k = 1 & \text{if } \; g(x) = 0 \\ B_k = 0 & , otherwise. \end{cases} \tag{8}$$

With Equations (6) and (7), we define a normalized decreasing contribution that propagates from each unvisited neuron toward the closest UAV $x_c$, through the shortest neuron chain. Then, we propagate this cost to neighbor neurons. Specifically, the propagation is only toward selected neurons. In this propagation, we consider the distance between each neighbor neurons and the closest UAV. Hence, given a generic neuron $x_k$, the cost is propagated toward the neuron $x_{k+1}$ with the following logic:

$$x_{k+1} = \arg \min_{x_{nb} \in X(x_k)} \|x_{nb} - x_c\|_2 \tag{9}$$

with $x_{nb}$ being a neighbor neuron and $X(x_k)$ being the set of neighbors of the neuron $x_k$.

The propagation continues until it reaches the selected neuron that corresponds to an obstacle or the closest UAV. Specifically, this last condition is satisfied if a neuron is inside the field of view of the UAV.

The computation of the attractive contribution and its propagation is computed for each uncovered neuron, similarly to [29]. The propagation is continuously repeated at every time step for each uncovered neuron in the map to obtain a complete and efficient cost-map used to define the next waypoint for each UAV.

The second term of Equation (5) considers the mean distance $d_{avg}$ between the neuron $x$ and the position of other UAVs (excluded the current one), normalized with the maximum distance $d_{max}$, and multiplied by the weighting factor $w_{md}$.

The third term of Equation (5) evaluates the standard deviation of the respective distances between UAVs. This term is essential in order to avoid high concentrations of units in any area of the map. Again, a maximum value ($vr_{max} = \sqrt{d_{avg}^2/(D-1)}$) is used to normalize this term, and its respective weight is multiplied: $(vr_{dist}/vr_{max})w_{vr}$, where $vr_d$ is the standard deviation of the distances evaluated in $d_{avg}$.

The fourth term of Equation (5) is defined to penalize the excessive turns of UAVs. ($\Delta\theta = |\theta_{t+1} - \theta_t|$) and represents the variation of the yaw angle of the UAV considering the orientation estimated at the next step and the current one. This parameter influences the energy consumption and, then, the autonomy of the UAV.

*Pseudocode*

To better clarify the various steps of the algorithm, in this section, we detailed the pseudocode. Algorithm 1 defines the main part of the proposed approach. First, we initialize the map, the initial position of the fleet, and all the variables. Hence, the algorithm runs until the UAVs cover 99% of the map (line 2). It should be noted that, after the map is covered, this algorithm can be executed again by setting the map as uncovered and covering it again. For simplicity, we report the pseudocode of visiting the map only once. In line 3, the cost-map is computed by using Equations (6)–(8). The computation of the cost-map will be detailed later, when the Algorithm 2 is described. Then, for each UAV of the fleet, the function $f(x)$ is computed by considering the neuron corresponding with the UAV position (line 5), and the best adjacent neuron is selected (line 6) as a candidate for the next move. After all, UAVs are evaluated, and all UAVs are moved simultaneously (line 8). If the fleet does not cover the map again, the algorithm ends (line 10).

**Algorithm 1** The main algorithm

```
1:  Initialize()
2:  while Covered area < 99% do
3:    ComputeCostmap()
4:    for each z_uav ∈ Z do
5:      Compute f(x)
6:      Select best adjacent neuron
7:    end for
8:    Move()
9:  end while
10: return
```

Algorithm 1 omits some details to facilitate its understanding. In particular, we always check that trajectories do not intersect obstacles or other UAVs. Furthermore, a check does not allow the drone to return to the previous position to avoid an endless loop.

As previously noted, Algorithm 2 describes in detail the Compute Cost-map (CC) function as it is one of the most significant phases of the proposed algorithm. First, the cost-

map is cleared at each time step (line 2) since all the attractive contributions in the cost-map should be computed considering the updated scenario and, then, considering the areas not already covered. Then, we compute the attractive cost for each uncovered neuron (from lines 2 to 10). The attractive cost is computed using Equation (6) and evaluating the closest UAV (lines 3, 4). Hence, the cost is propagated toward the closest UAV (from lines 5 to 9) and the propagation is stopped only if the next neuron is an obstacle ($x_k \neq -1$) or the closest UAV is reached, i.e., if $x_k \notin v(z_c)$ with $v(z)$ is the field of view of the UAV $z$. After the best neuron for propagation is selected (line 6), the attractive cost is computed (line 7), and the propagation phase continues (line 8). With this logic, the cost-map continuously attracts UAVs toward uncovered areas. Figure 5 reports an example of a cost-map that dynamically changes during the exploration.

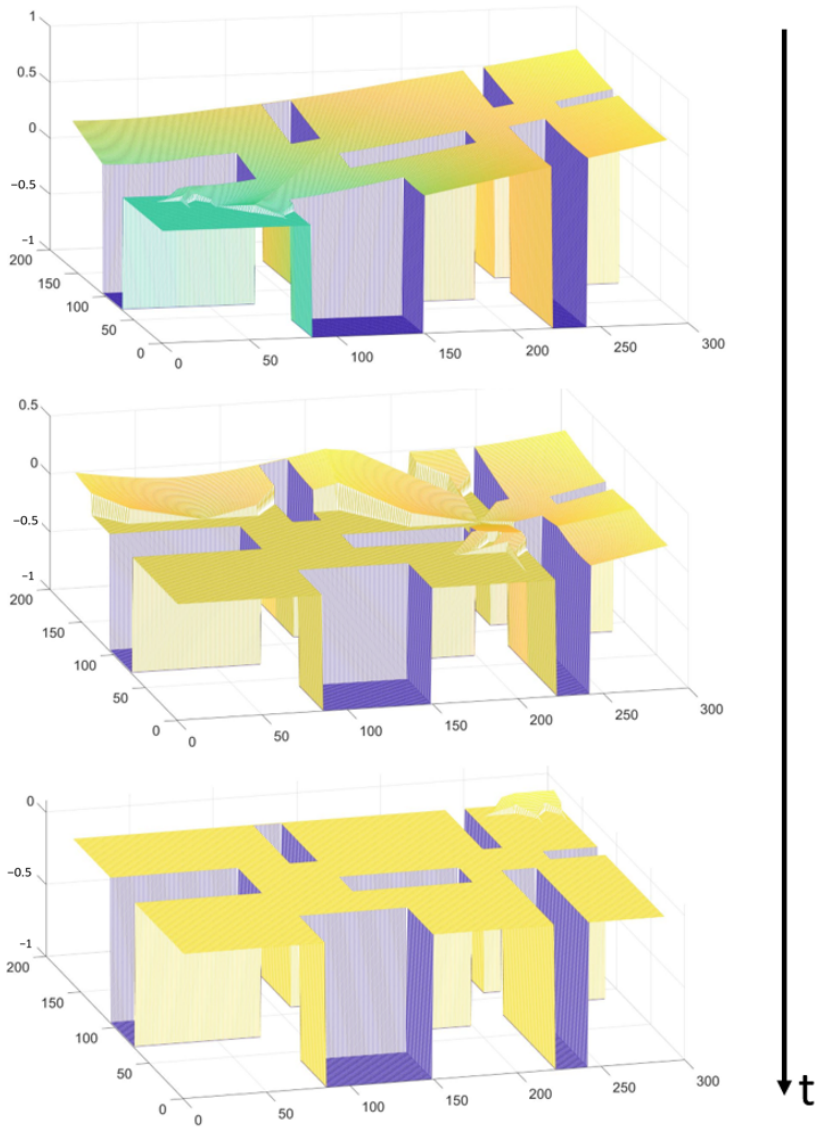


**Figure 5.** Dynamic evolution of the cost-map during the coverage of Field 4. The negative value $-1$ corresponds to neurons occupied by an obstacle. The initial condition is determined as in Figure 2.

**Algorithm 2** The ComputeCostmap() function

```
ClearCostmap()
for each uncovered neuron x_k do
  Find the closest UAV z_c ∈ Z to x_k
  Compute C(x_k)
  while x_k ≠ −1 or x_k ∉ v(z_c) do
    Select the best x_{k+1}
    Compute C[x_{k+1}]
    x_k = x_{k+1}
  end while
end for
return
```

## 4. Simulations

### *4.1. Assumptions*

Before explaining the results obtained with preliminary simulations with MATLAB and, then, using ROS and Gazebo frameworks, we define the assumptions considered during the implementation of the proposed approach:

- The positions of all UAVs are always known; therefore, the cost-map $C$, the mean distance $d_{avg}$, the standard deviation of distances between vehicles $vr_{dist}$, and $\Delta\theta$ contributions can be computed by the centralized coordination unit at each time step;
- The UAVs are flying at a fixed flight altitude, and their field of view is constant and defined as shown in Figure 3b;
- The fleet starts from the upper-left corner of the map. The UAVs are in a compacted formation, as shown in Figure 2. This assumption is valid for most of the simulations, except for the simulations of Figures 6 and 7, in which, for simplicity, the fleet is already distributed on the map;
- We always assume a known map. However, the proposed approach can be used also in unknown environments requiring obstacle detection with sensors to update the map during the exploration.

### *4.2. Preliminary Simulations*

First of all, to show and demonstrate how the proposed approach works, we test the proposed algorithm in different scenarios (from Field 1 to Field 4 maps) and with a fleet composed of a different number of UAVs. Only in these tests, to have a graphically more understandable result, we define a different initial condition, with the UAVs already distributed in the map and not in the upper left as defined in the assumptions.

Specifically, Figure 6 shows how the proposed approach works in different scenarios. Notably, increasing the environment complexity makes it more complex to coordinate the fleet for maintaining a uniform distribution. On the other hand, a higher percentage of obstacles involves a lower number of locations to be visited, requiring a lower number of moves. The layout of the obstacles has a significant influence on the coordination of the fleet.

Moreover, as anticipated, the algorithm is also tested evaluating fleets consisting of 3 up to 10 UAVs. The results are shown in Figure 7 in the Field 4 map, i.e., the most complex scenario. The fleet of three UAVs takes more moves to fully cover the map than larger fleets. Moreover, with a smaller fleet, it is easier to maintain a uniform distribution on the map, especially in complex environments.

In the following paragraphs, we analyze the performances of the proposed approach in terms of number of moves to cover at least the 99% of the area, and the distribution

of the fleet in the map, evaluated taking into account the maximum distance between obstacle-free neurons and the closest UAV.

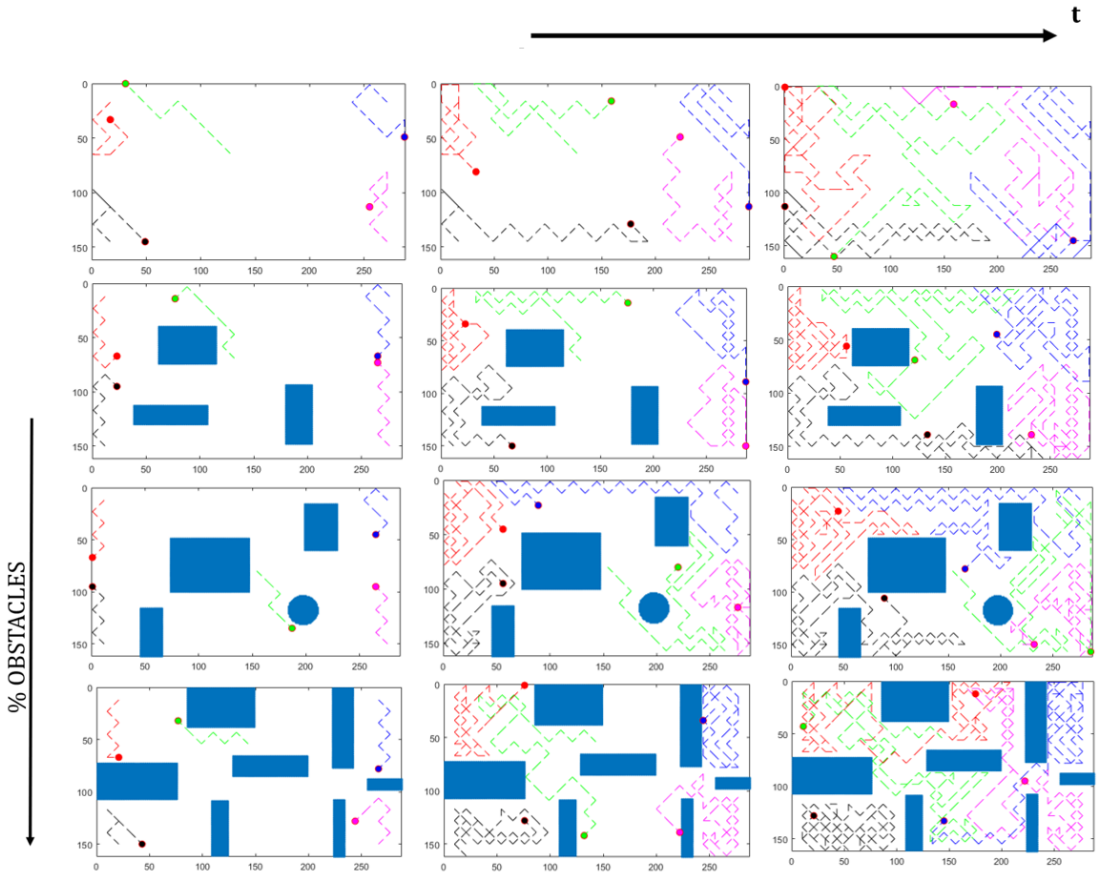


**Figure 6.** Temporal evolution of the fleet performing the coverage task with five UAVs. From the top, the complexity of maps is increased incrementally.

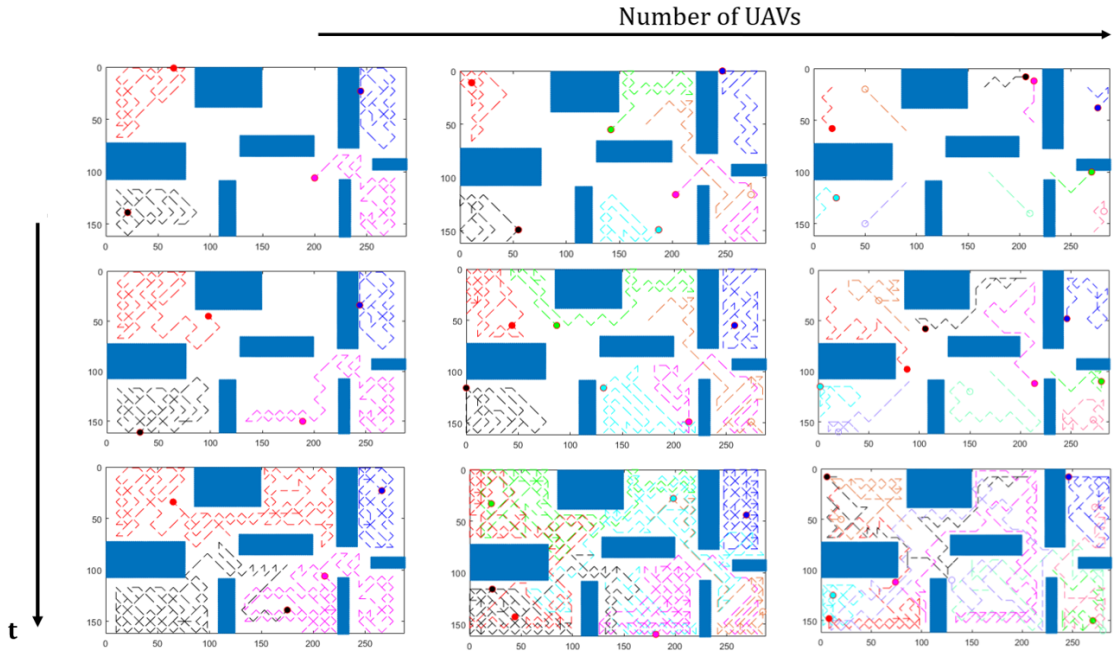


**Figure 7.** Temporal evolution of the fleet performing the coverage task in the Field4 map. Simulations are performed with a fleet composed of 4, 7, and 10 UAVs.

### *4.3. Tuning Parameters*

In the proposed coverage planning problem strategy, there are two main features of the coverage that can be optimized. The first one is the coverage capacity, i.e., the number of moves required by the fleet to cover the 99% of the map. The second one is the ability of the UAVs to maintain a uniform distribution in the map, evaluated using the maximum distance between obstacle-free locations in the map and the closest UAV. This distance is denoted as *"Max UAV - free point dist"* in Figures 8 and 9.

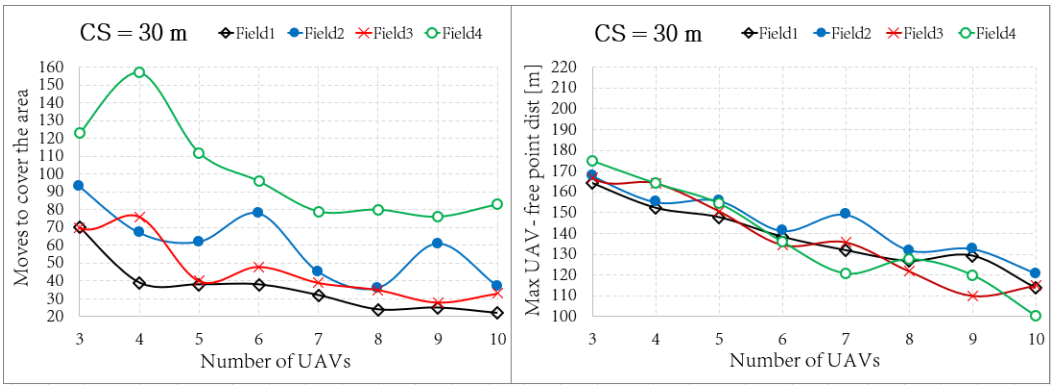


**Figure 8.** Results obtained after the fleet's uniform distribution optimization shown in Figure 10.

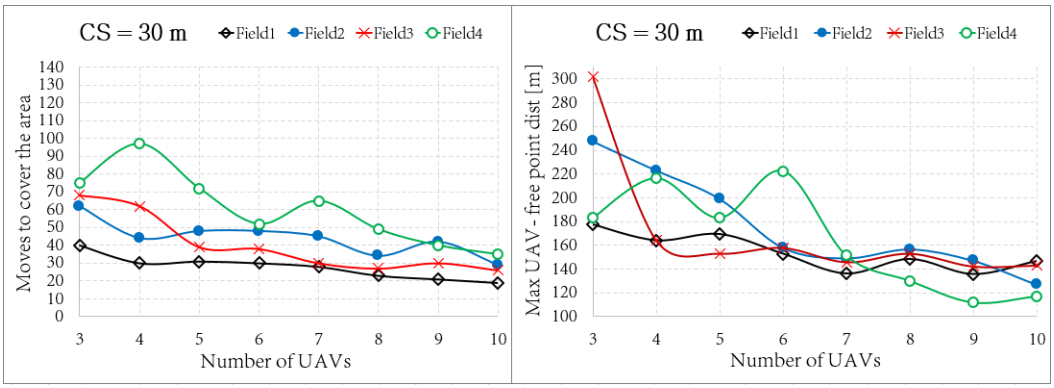


**Figure 9.** Results obtained after the fleet's coverage optimization shown in Figure 11.

The behavior of the proposed approach can be balanced by tuning the coverage capacity of the uniform distribution of the fleet by determining the weighting factors $w_{md}$ and $w_{vr}$ defined in Equation (5). To identify how to tune these factors, an extensive set of simulations is performed. All combinations of parameters are evaluated (with a step of 0.1), and the best values are reported in the plots of Figures 10 and 11, by optimizing the distribution of the fleet and the coverage capacity, respectively. Each test is performed until the coverage of the 99% of the map is provided.

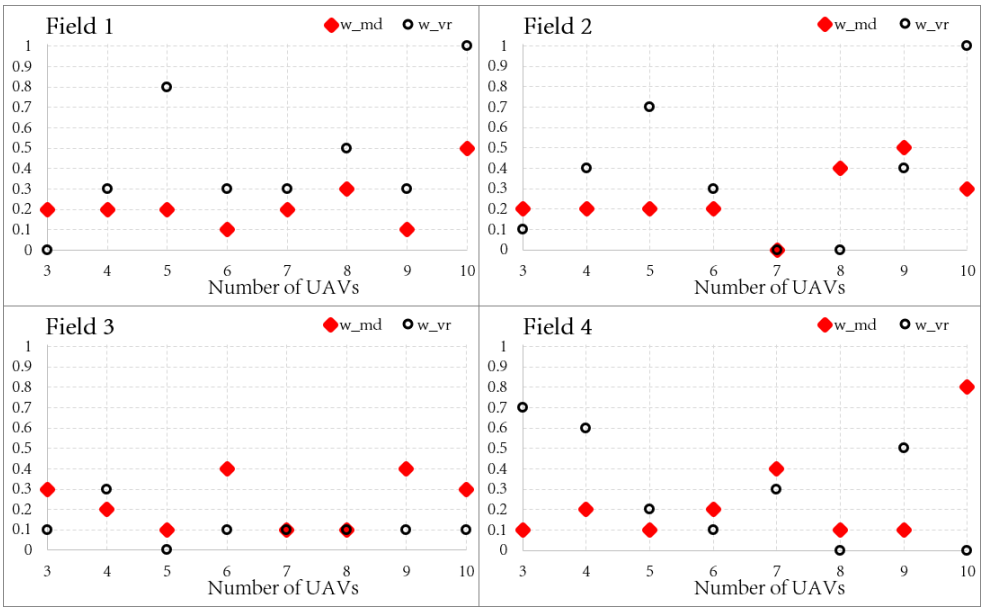


**Figure 10.** Weight parameters ($w_{md}$ and $w_{vr}$) tuning to optimize the uniform distribution of the UAVs in the map (with $CS = 30$ m). The tuning is performed for all of the four maps shown in Figure 6.

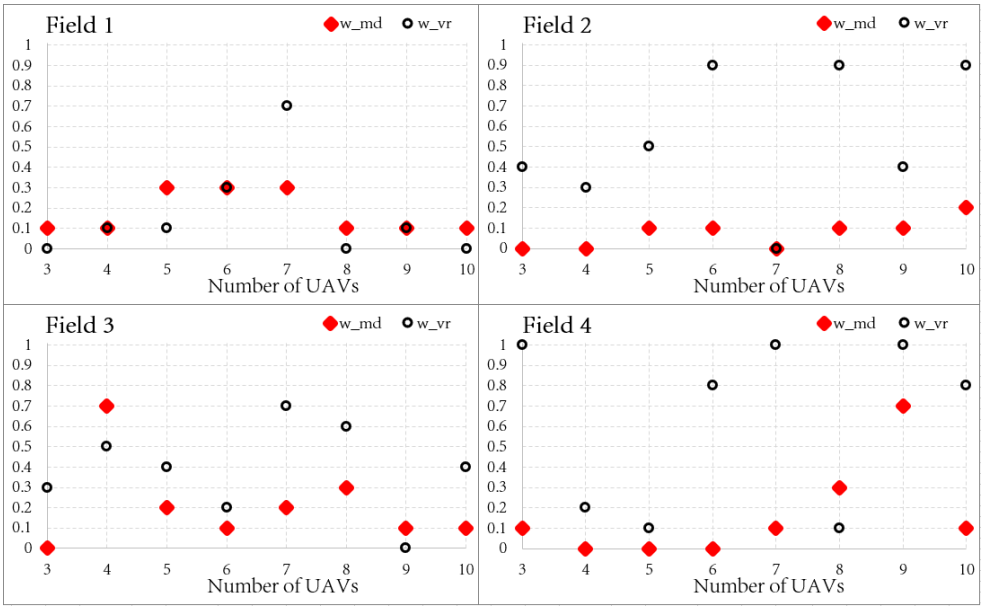


**Figure 11.** Weight parameters ($w_{md}$ and $w_{vr}$) tuning to optimize the fleet moves needed to cover the 99% of the map (with $CS = 30$ m). The tuning is performed for all four maps shown in Figure 6.

### *4.4. Collected Results*

The best set of parameters defined in the previous paragraph are used to provide the final results. Figures 8 and 9 show the results optimizing both the distribution of the fleet and the coverage capacity. The results are evaluated in terms of moves needed to fully cover the area and the so-called *Max UAV-free point dist* which represents the maximum distance between an obstacle-free location in the map and the nearest UAV. Specifically, this feature is adopted to evaluate the uniform distribution of the fleet on the map.

Analyzing the obtained results, we can affirm that, with larger fleets (around 10 units), there is no substantial difference in both scenarios, i.e., optimizing the coverage capacity and the uniform distribution in the map. The results converge towards similar values, except for the most complex environment (Field 4), where the optimization of the coverage capacity has a strong effect.

On the contrary, with smaller fleets, the optimization changes the fleet behavior a lot, especially considering the Field 4 scenario. Figure 9 shows how the moves required to cover the area in the most complex map have peak values lower than those in Figure 8, where, instead, the same beneficial effect is notable for the parameter related to the uniform distribution.

Moreover, we want to highlight the results of the simulation performed in the Field 3 map and reported in Figure 9, in which there is a non-optimal behavior of the fleet composed of three UAVs, with a bad performance in terms of uniform distribution without having a particular benefit on the coverage parameter that remains quite similar to the result of Figure 8. However, this bad behavior disappears for all of the other simulations with different maps and different fleet dimensions.

Overall, it is notable that the effect of tuning generates an improvement in the respective performances. However, the best choice of those parameters is related to the type of application that should be conducted.

### *4.5. Computational Time*

One of the main advantages of the proposed approach is the short computational time required to compute the target for each UAV of the fleet. As shown in Figure 12, the computational time varies from 2 s to 0.1 s, without considering the initialization that requires an additional computational time. However, the computational time is a short value considering the hardware used to perform the simulation (4-core 2.80 GHz). This time can be drastically reduced using high-performance computers.

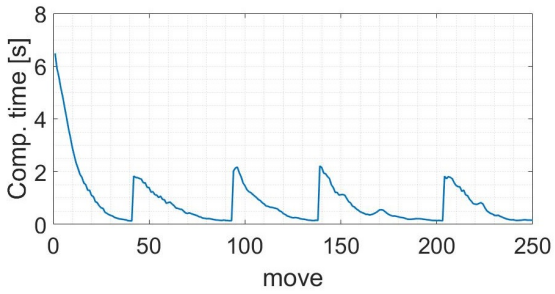


**Figure 12.** The computational time required to perform the most complex simulation, i.e., with a fleet of 10 UAVs in the Field 4. Excluding the initialization phase, the computational time varies between 2 s and 0.1 s.

As depicted in Figure 12, the computational time decreases during the exploration due to the decreasing number of attractive contributions to be assigned, as described in Algorithm 2. Once all the map is visited, the coverage planning task is repeated cyclically.

The target position computed for each UAV is always at a distance that depends on CS, as shown in Figure 3. As a consequence, the shortest move is $d_{min} = \frac{CS}{2} + 1$ and, considering a constant cruise speed, we can define the maximum cruise speed $v_{max}$ admissible to be sure that the algorithm is always able to compute the next targets. Then,

$$v \leq v_{max} = \frac{d_{min}}{\max(T_{comp})}, \tag{10}$$

with $v$ the cruise speed of the UAV.

Considering the simulated scenario, we have $CS = 30$ m and, then, $d_{min} = 16$ m. Considering the maximum computational time of 2 s and applying Equation (10), we should maintain a constant cruise speed lower than about 8 m/s. Hence, even with the current performances, the developed approach is suitable for real applications. In fact, by analyzing the scenario used for the simulation, the developed algorithm can continuously compute the next target for each UAV. However, this speed constraint can be changed

by increasing the field of view (e.g., changing the camera or increasing the flight altitude) or using hardware with higher performances.

### *4.6. Realistic Simulations*

As anticipated, the proposed approach is tested and validated with a more realistic simulation using Robot Operating System (ROS), Gazebo simulator, and Software In The Loop (SITL).

ROS is an open-source meta-operating system for robotic systems [30]. Practically, ROS is a standard for robot programming and offers a general-purpose robotics library for robotic applications.

Gazebo is an open-source multi-robot simulator fully compatible with ROS [31] able to simulate robots, sensors, and rigid body dynamics.

In particular, we based the simulation on the framework proposed in the PX4 flight stack [32]. Specifically, PX4 is an open-source flight control software for drones and other autonomous vehicles. The PX4 autopilot is used to control UAVs simulated with Gazebo using the SITL framework [33], where SITL allows PX4 to be executed without using any hardware.

Practically, instead of exchange data with a real drone platform, the PX4 Autopilot controls the UAV simulated in Gazebo, which executes control commands and provides sensor data from simulated sensors. Hence, we control the UAV offboard using ROS and the mavros package, enabling the communication between ROS and the PX4 autopilot via MAVLink.

In particular, in each simulation, we use the centralized architecture illustrated in Figure 13. We defined a ROS environment for each UAV of the fleet by allocating all the ROS nodes to simulate and control the UAV. Hence, we present a centralized ROS environment able to manage and coordinate the fleet of drones using the proposed coverage planning methodology. The proposed algorithm receives the poses from all the UAVs in the fleet and, using the proposed method, determines the sequence of goals for each UAV providing the coverage planning.

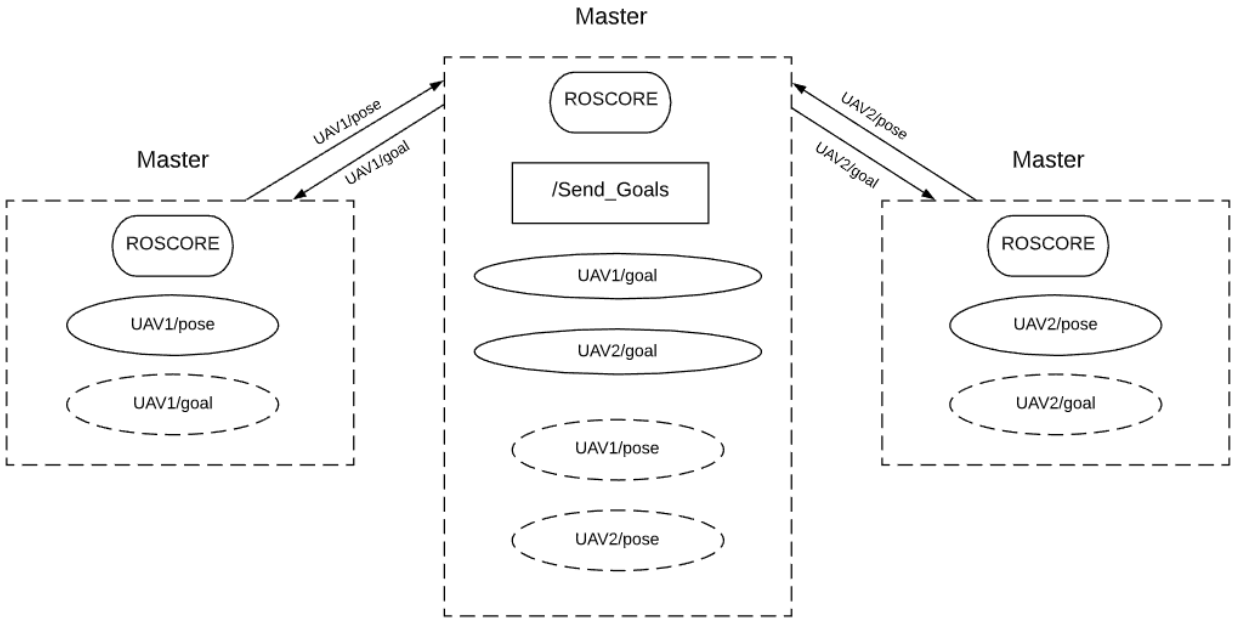


**Figure 13.** ROS general architecture of a multi-robot system, similar to the one presented in [34]. In this case, we show a fleet configuration with two UAVs; for larger fleets, the logic is the same.

The realistic simulations using ROS/Gazebo/SIT are shown in Figure 14.

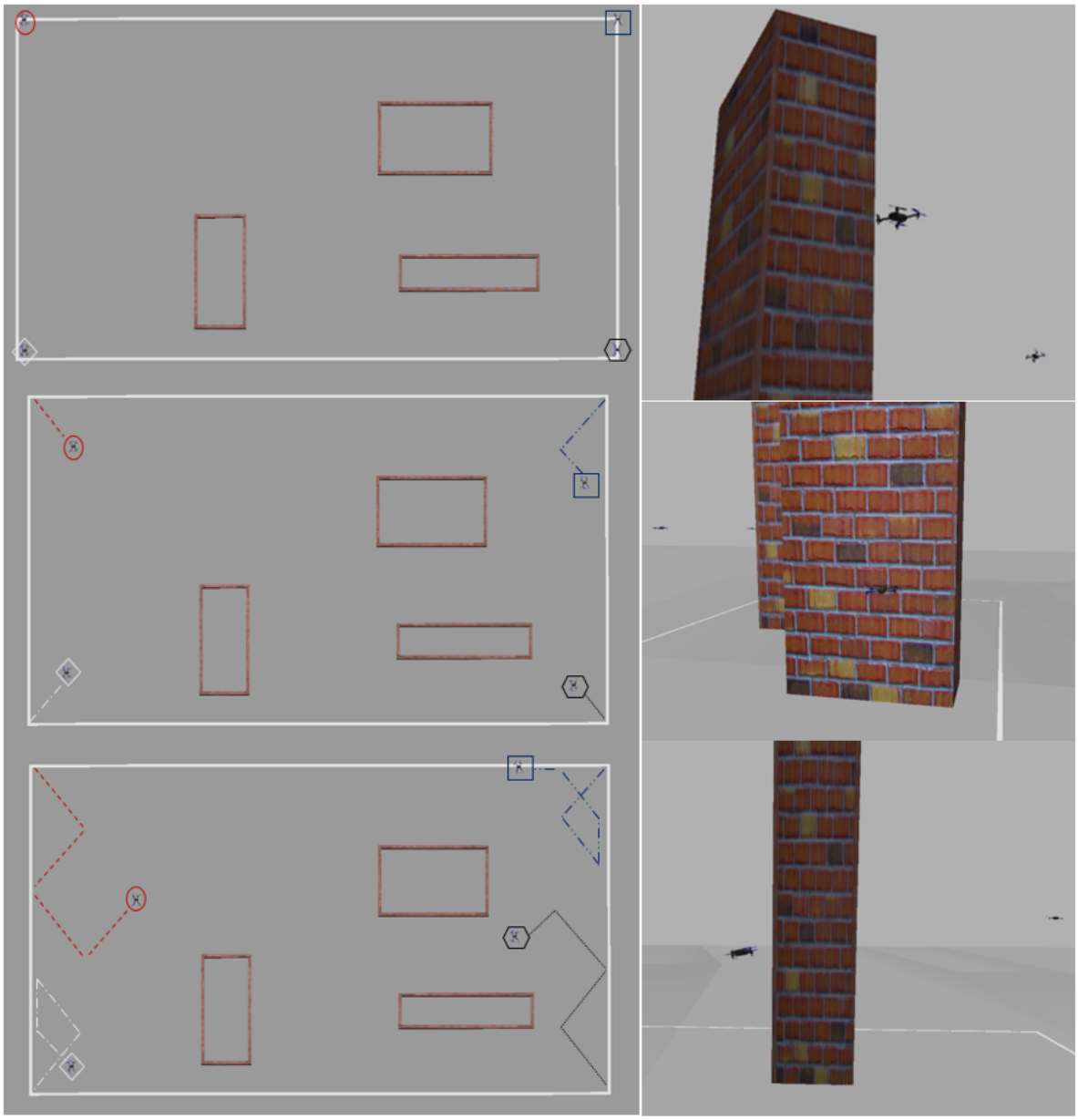

**Figure 14.** Top and side view of the simulation performed using ROS/Gazebo/SITL in the Field 2 map and using a fleet of four UAVs. From the top, three steps at different simulation time are reported. The fleet is coordinate using the ROS architecture of Figure 13.

## 5. Conclusions and Further Developments

In this paper, we present an innovative approach to solve the coverage planning problem with a fleet of UAVs. The proposed solution is based on a bioinspired neural network approach in which the dynamics of neurons depend on unvisited areas, the presence of obstacles in the map, and the position of UAVs in the map. As result, the neural network guides UAVs to fully cover the map and, at the same time, maintaining a uniform distribution of the fleet in the map. The results obtained in simulations show that the proposed strategy can manage a fleet composed of 3 to up to 10 units of good performances, as well as in complex maps.

The proposed solution wants to define a promising approach to coordinate a fleet of UAVs for surveillance or search and rescue purposes in urban areas. The uniform distribution of the UAVs in the map improves the responsiveness of the fleet to reach any position on the map as fast as possible. This feature is essential for surveillance

and monitoring applications, in which, in case of emergency, a vehicle needs to reach a specific location.

Moreover, by tuning ad-hoc the algorithm and disabling the possibility of hovering, it is possible to exploit the proposed approach for fixed-wing aircraft. Anyway, it requires an adaptation of parameters and assumptions, since, generally, fixed-wing aircraft involve a large field of view and a greater cruise velocity.

Future works will include the development of the proposed approach to be used for real-world scenarios. One of the essential aspects that must be taken into account is the bidirectional communication between the UAVs and the server. In fact, in the current work, we have assumed an ideal communication channel, not affected by delays or disconnections. Moreover, the proposed approach should be extended to be used in an unknown area, requiring the obstacle detection using on-board sensors and, then, updating the map and, as a consequence, the grid neural network.

**Author Contributions:** Conceptualization, S.G.; methodology, S.G. and S.P.; software, S.G.; validation, S.P., G.G. and F.D.; formal analysis, S.P., G.G.; investigation, S.G. and S.P.; resources, S.P., G.G. and F.D.; data curation, S.G. and S.P.; writing—original draft preparation, S.G., S.P. and G.G.; writing—review and editing, S.G., S.P. and G.G.; visualization, S.G. and S.P.; supervision, G.G. and F.D.; project administration, G.G. and F.D.; funding acquisition, S.P. and G.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data is contained within the article.

**Acknowledgments:** The Ph.D. fellowship of Simone Godio is funded by the Leonardo Company, Italy. The research program is shared with PIC4SeR—Politecnico di Torino Interdepartmental Centre for Service Robotics.

**Conflicts of Interest:** The authors declare no conflict of interests.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| CS | Covered square side (in meters) of the field of view |
| max point dist | Maximum distance between an obstacle free point of the map and the closest UAV |
| SITL | Software In The Loop |
| ROS | Robot Operating System |
| UAV | Unmanned Aircraft System |
| D | Number of Units |
| FOV | Field Of View |

## References

1. Otto, A.A.; Agatz, N.; Campbell, J.J.; Golden, B.B.; Pesch, E.E. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones. *Networks* **2018**. [CrossRef]
2. Juliá, M.; Gil, A.; Reinoso, O. A comparison of path planning strategies for autonomous exploration and mapping of unknown environments. *Auton Robot* **2012**, *33*, 427–444. [CrossRef]
3. Chen, Y.; Zhang, H.; Xu, M. The coverage problem in UAV network: A survey. In Proceedings of the Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Hefei, China, 11–13 July 2014; pp. 1–5. [CrossRef]
4. Khan, A.; Noreen, I.; Habib, Z. On Complete Coverage Path Planning Algorithms for Non-holonomic Mobile Robots: Survey and Challenges. *J. Inf. Sci. Eng.* **2017**, *33*, 101–121.
5. Cabreira, T.M.; Brisolara, L.B.; Ferreira, P.R., Jr. Survey on coverage path planning with unmanned aerial vehicles. *Drones* **2019**, *3*, 4. [CrossRef]
6. Waharte, S.; Trigoni, N. Supporting search and rescue operations with UAVs. In Proceedings of the 2010 International Conference on Emerging Security Technologies, Canterbury, UK, 6–7 September 2010; pp. 142–147.

7. Alami, R. A fleet of autonomous and cooperative mobile robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Osaka, Japan, 4–8 November 1996.
8. Albani, D.; Nardi, D.; Trianni, V. Field coverage and weed mapping by UAV swarms. In Proceedings of the IEEE International Workshop on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017.
9. Shakhatreh, H.; Khreishah, A.; Chakareski, J.; Salameh, H.B.; Khalil, I. On the continuous coverage problem for a swarm of UAVs. In Proceedings of the IEEE Princeton Section Sarnoff Symposium, Newark, NJ, USA, 19–21 September 2016.
10. Belkadi, A.; Abaunza, H.; Ciarletta, L.; Castillo, P.; Theilliol, D. Design and Implementation of Distributed Path Planning Algorithm for a Fleet of UAVs. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 2647–2657. [CrossRef]
11. Shao, Z.; Yan, F.; Zhou, Z.; Xiaoping, Z. Path Planning for Multi-UAV Formation Rendezvous Based on Distributed Cooperative Particle Swarm Optimization. *Appl. Sci.* **2019**, *9*, 2621. [CrossRef]
12. Ropero, F.; Munoz, P.; R-Moreno, M. TERRA: A Path Planning Algorithm for Cooperative UGV-UAV Exploration. *J. Logo* **2018**, *78*, 260–272. [CrossRef]
13. Ashraf, A.; Majid, A.; Troubitsyna, E. Online Path Generation and Navigation for Swarms of UAVs. *Sci. Program.* **2020**, *2020*, 8530763. [CrossRef]
14. Gorecki, T.; Piet-Lahanier, H.; Marzat, J. Cooperative guidance of UAVs for areaexploration with final target allocation. In Proceedings of the IFAC Symposium on Automatic Control for Aerospace, Wurzburg, Germany, 2–6 September 2013.
15. Messous, M.; Senouci, S.; Sedjelmaci, H. Network Connectivity and Area Coverage for UAV Fleet Mobility Model with Energy Constraint. In Proceedings of the IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016.
16. Maza, I.; Caballero, F.; Capitan, J.; Martinez-de Dios, J.; Ollero, A. Experimental Results in Multi-UAV Coordination for Disaster Management and Civil Security Applications. *J. Intell. Robot. Syst.* **2011**, *61*, 563–585. [CrossRef]
17. Perez-Montenegro, C.; Scanavino, M.; Bloise, N.; Capello, E.; Guglieri, G.; Rizzo, A. *A Mission Coordinator Approach for a Fleet of UAVs in Urban Scenarios*; Elsevier: Amsterdam, The Netherlands, 2018.
18. Elias, X.; Zacharia, P.; Nearchou, A. Path Planning and scheduling for a fleet of autonomous vehicles. *Robotica* **2015**, *34*, 2257–2273.
19. Shorakaei, H.; Mojtaba, V.; Imani, B.; Gholami, A. Optimal cooperative path planning of unmanned aerial vehicles by a parallel genetic algorithm. *Robotica* **2016**, *34*, 823–836. [CrossRef]
20. Mezghani, F.; Mitton, N. Opportunistic multi-technology cooperative scheme and UAV relaying for network disaster recovery. *Information* **2020**, *11*, 37. [CrossRef]
21. Liang, M.; Delahaye, D. Drone Fleet Deployment Strategy for Large Scale Agriculture and Forestry Surveying. In Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019.
22. Bailon-Ruiz, R.; Bit-Monnot, A.; Lacroix, S. Planning to Monitor Wildfires with a Fleet of UAVs. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
23. Zhang, M.; Song, J.; Huang, L.; Zhang, C. Distributed Cooperative Search with Collision Avoidance for a Team of Unmanned Aerial Vehicles Using Gradient Optimization. *J. Aerosp. Eng.* **2017**, *30*, 1–11. [CrossRef]
24. Pham, H.X.; La, H.; Feil-Seifer, D.; Nguyen, L. Cooperative and Distributed Reinforcement Learning of Drones for Field Coverage. *arXiv* **2018**, arXiv:1803.07250.
25. Theile, M.; Bayerlein, H.; Nai, R.; Gesbert, D.; Caccamo, M. UAV Coverage Path Planning under Varying Power Constraints using Deep Reinforcement Learning. *arXiv* **2020**, arXiv:2003.02609.
26. Zhu, L.; Zheng, X.; Li, P. Reconstruction of 3D Maps for 2D Satellite Images. In Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China, 20–23 August 2013.
27. Lee, D.; Yom, J.; Shin, S.W.; Oh, J.; Park, K. Automatic Building Reconstruction with Satellite Images and Digital Maps. *ETRI J. Comput. Sci.* **2011**, *33*, 537–546. [CrossRef]
28. Glasius, R.; Komoda, A.; Gielen, S.C. Neural network dynamics for path planning and obstacle avoidance. *Neural Netw.* **1995**, *8*, 125–133. [CrossRef]
29. Yang, S.X.; Luo, C. A neural network approach to complete coverage path planning. *IEEE Trans. Syst. Man Cybern. Part B* **2004**, *34*, 718–724. [CrossRef] [PubMed]
30. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–13 May 2009; Volume 3.2, p. 5.
31. Koenig, N.P.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *IROS Citeseer* **2004**, *4*, 2149–2154.
32. Meier, L.; Honegger, D.; Pollefeys, M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In Proceedings of the 2015 IEEE international conference on robotics and automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6235–6240.
33. SITL Contributors. SITL Guide. 2020. Available online: http://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html (accessed on 22 January 2021).
34. Singhal, A.; Pallav, P.; Kejriwal, N.; Choudhury, S.; Kumar, S.; Sinha, R. Managing a fleet of autonomous mobile robots (AMR) using cloud robotics platform. In Proceedings of the 2017 European Conference on Mobile Robots (ECMR), Paris, France, 6–8 September 2017; pp. 1–6. [CrossRef]