Journal of Intelligent & Robotic Systems
https://doi.org/10.1007/s10846-019-01090-2

Strategies for Patrolling Missions with Multiple UAVs

Kristofer S. Kappel[1] · Tauã M. Cabreira[1] · João L. Marins[1] · Lisane B. de Brisolara[1] · Paulo R. Ferreira Jr.[1]

Received: 28 May 2019 / Accepted: 23 August 2019

Abstract

This paper proposes a set of strategies for the patrolling problem using multiple UAVs and as a result, improving our original NC-Drone algorithm. We present four strategies: Watershed Strategy, Time-based Strategies, Evaporation Strategy, and Communication-Frequency Strategy. The novel strategies consider important aspects of the patrolling movement, such as time, uncertainty, and communication. Results point out that these strategies improve the centralized version of the NC-Drone considering the uniform distribution of visits and drastically reduce in 76% the standard deviation, making the algorithm more stable. Based on the results, we found that there is a trade-off between the evaluated metrics, making it necessary to perform a large number of turns to obtain a more spatially distributed patrolling. We also present a series of strategy combinations, achieving slight improvements as more combinations are adopted. The resulting algorithm from the combination of all strategies reduces the communication frequency in 50 times and outperforms the original version of the NC-Drone in 4.5%.

Keywords Patrolling problem · Unmanned aerial vehicles · Watershed · Evaporation · Communication

1 Introduction

Unmanned Aerial Vehicles (UAVs) have been useful in the development of different fields of study, such as smart farming [1, 2], civil security [3], wildfire tracking [4, 5], and cloud monitoring [6]. These vehicles have also been used in a large set of applications, including patrolling [7], surveillance [8], search and rescue [9, 10], and photogrammetry [11]. Many of these UAV applications are related to the Coverage Path Planning (CPP) problem, which is a subtopic of robot motion planning. This problem consists of determining a path for full coverage of every location in a given scenario by an autonomous vehicle [12].

The patrolling problem is a specific domain of the CPP problem dealing with monitoring and inspecting areas of interest. In such activities, a cooperative fleet must visit and revisit areas at regular intervals in order to supervise it [13]. This surveillance task may include searching for intelligent targets [14] and monitoring climate changes [15]. Different metrics can measure the patrolling performance, such as probability of detecting a target and tracking time [14], information latency [16], and certainty and interest level [17, 18]. Other metrics prioritize the uniform visitation, such as the quadratic mean of the intervals (QMI) and standard deviation of frequencies (SDF) [19]. The QMI corresponds to frequency-regularity equilibrium, which combines the frequency of visits and evenness of intervals between these visits, while the SDF computes the distribution equality of visits over the cells of the scenario. There is also the number of complete coverage and number of turning maneuvers [20]. The former one consists of the number of times which the UAVs full cover an area, while the latter one counts the number of times a

The work of K. Kappel, L. B. de Brisolara, and P. R. Ferreira Jr. was financed in part by FAPERGS/CNPq under Grant 16/2551-0000/472-2. The work of P. R. Ferreira Jr. was also supported in part by CNPq under Grant 308487/2017-6.

✉ Kristofer S. Kappel
kskappel@inf.ufpel.edu.br

Tauã M. Cabreira
tmcabreira@inf.ufpel.edu.br

João L. Marins
joao.marins@inf.ufpel.edu.br

Lisane B. de Brisolara
lisane@inf.ufpel.edu.br

Paulo R. Ferreira Jr.
paulo@inf.ufpel.edu.br

[1] Programa de Pós-Graduação em Computação (PPGC), Centro de Desenvolvimento Tecnológico (CDTec), Universidade Federal de Pelotas (UFPel), 96010-610, Pelotas, Brazil

Published online: 12 September 2019

UAV change its sweeping direction during the patrolling. Some approaches deal with different local priorities [21] and preferential surveillance [22]. The performance metric depends on the main goal of the task and must fulfill the application requirements.

Flight patterns based on simple geometric shapes, such as back-and-forth (BF) [23–27] and spiral (SP) [28–32], are generally adopted for coverage tasks in regular-shaped areas. However, these flight patterns are not useful for patrolling missions and military surveillance aiming at monitoring intelligent targets. Such coverage approaches follow a pre-defined pattern which repeats the same behavior during the coverage. In this way, the object can avoid being tracked by heading to a different direction and staying out of sight of the UAV.

Real-time Search Methods (RTSM) interleave between path planning and path execution during the coverage and have been addressed to solve patrolling problems with land robots. Each robot, for instance, can use an RFID tag to read and write in the cells on the floor [10]. When performing a motion, the robot must read the neighbor cells and count the number of visits on each one of them. Then, it chooses the least visited cell to move in. Finally, it updates the value of its current cell according to the adopted method. RTSM have also been addressed in patrolling missions with aerial vehicles. Such methods can prevent intelligent targets from escaping the UAV since they are unpredictable to an external observer. However, the unpredictability of the RTSM demands a large number of changes in the vehicle's direction, increasing energy consumption in aerial platforms [28]. Thus, it is necessary to minimize the number of turning maneuvers of such methods to obtain an energy-aware robust-to-failure solution.

Computational power represents a rigid constraint to any autonomous vehicle [33]. Since RTSM have low computational costs, they have been recently applied in aerial coverage using UAVs [10, 20, 34]. A Ground Control Station (GCS) is responsible for splitting the scenario into virtual cells. The GCS has a global map of the area of interest and coordinates the movement of the vehicles. The station monitors the positions of the UAVs and reads the neighbors cells, sending the corresponding coordinates to guide their motions. The main goal is to lead the UAVs to the less frequently explored areas. The GCS is also in charge of avoiding crashes between the aircrafts [35, 36]. However, this centralized decision-making process requires massive communication among the UAVs and the station. Each action of a vehicle depends on the GCS. It is necessary to periodically update the station about the current status of the fleet. Interference or failure in the communication may compromise the mission, resulting in inefficient coverage. Loss of a UAV, a collision between vehicles or a crash in an inhabited place are some of the problems which may also occur. This fashion of decision-making process imposes several limitations to its application in real-world scenarios.

Our previous work presented an energy-aware RTSM for cooperative patrolling missions, called NC-Drone [20]. This proposed method is an extension of the Node Counting (NC) algorithm aiming at minimizing the number of turning maneuvers to save energy while keeping an unpredictable behavior. We compared the NC-Drone with four original RTSM: NC, Learning Real-Time A* (LRTA*), Thrun's Value-Update Rule (TVUR), and Wagner's Value-Update Rule (WVUR). The performance metrics used to compare the experiments were: Quadratic Mean of Intervals (QMI), Standard Deviation of Frequencies (SDF), Number of Complete Coverages (NCC), and Number of Turning Maneuvers (NTM). Decentralized variations of the NC-Drone were also explored using internal matrices as the representation of the global map, instead of adopting a centralized scheme with a GCS. We also presented and compared some variations of synchronization schemes to merge and combine data stored in the matrices. These schemes require minimal communication among vehicles and are capable of replacing the centralized decision-making process, supporting real-world applications.

This paper extends our previous work [20] and we aim to propose cooperative strategies for the patrolling problem using multiple UAVs. These strategies improve the previously proposed NC-Drone and further explore the problem considering relevant aspects, such as time, uncertainty, and communication. First, we introduce the Watershed Strategy (WS), a technique used to represent matrices as topographic relief. In this relief, the lower elevations correspond to the minor values in the matrix, i.e., less frequently explored cells. The UAVs are attracted to clusters of cells formed in these needy areas. Second, we present the Time-based Strategy (TS), an approach exploring not only the number of visits but also the time when the last visit occurs to guide the UAVs during coverage. Both strategies can be combined. Then, we propose the Evaporation Strategy (ES) to model uncertainty due to the absence of visits in certain places during patrolling.

We also explore the concept of full-range communication with Communication-Frequency Strategy (CFS), where UAVs reduce communication, exchanging matrix-information from time to time. We are interested in studying the impact of the communication in the patrolling performance in terms of spatial and temporal distribution of visits and the trade-off between communication and performance. Finally, we combine all strategies into a single solution for the patrolling problem.

This paper is organized as follows: Section 2 presents the theoretical foundations regarding real-time search methods. Section 3 addresses the related work considering pheromone-based methods. Section 4 recaps the previous work and explains the proposed cooperative strategies. Section 5 presents the simulation experiments, discussing the obtained results. Section 6 concludes the work and expands ideas for future work in the field of patrolling using decentralized pheromone-based approaches.

2 Theoretical Foundations

A coverage trajectory for a given area can be planned offline, i.e., before performing the path or online, alternating between planning and execution. In the former case, the UAV path planner considers a static scenario and possesses full knowledge about it, including area coordinates, obstacles positions, and no-fly zones. In the latter case, the UAV path planner deals with a dynamic scenario which may contain mobile obstacles and targets. Other aerial vehicles may also be present in the area. In this case, the UAV has to gather data using onboard sensors to create an internal map while covering the area. This map must be constantly updated to avoid collisions with other vehicles and to maintain quality of coverage.

Real-time search methods (RTSM) are suitable for this type of mission where UAVs contain incomplete information about a partially known environment. These biologically-inspired methods comprise algorithms based on social colonies of insects, specifically ants, as their natural behavior consists of leaving traces of pheromones while exploring areas outside the anthill [37]. The pheromones are useful to guide the swarm during the forage for food. When an ant finds food, it returns to the anthill using the same path and reinforces the pheromone traces previously left along the way. Stronger traces of pheromones attract other ants to the prosperous area, which explains the success of the social colonies.

RTSM mimic the ants' pheromones by applying virtual marks on a grid-discretized scenario. The area of interest is decomposed into a set of regular cells using the approximate cellular decomposition [12]. The size of the cells depends on the resolution of the onboard camera of the UAV and the flight altitude adopted during the patrolling task. The higher the altitude, the larger the footprint of the UAV and the size of each cell, and the lower the quality of the images. The balance between these elements depends on the application requirements. The UAV's field of view is restricted to the eight neighbor cells and the current one, as illustrated by Fig. 1a. However, the UAV's actions of reading and writing in the cells for search and displacement are based on the von Neumann neighborhood, as presented by the connected graph in Fig. 1b. Coverage is performed by adopting local search and exploring only the nearest neighborhood area.

While moving around the area of interest, UAVs leave virtual marks in the visited cells. Each cell works with an associated value, called *u-value*, to count the total number of visits performed by UAVs. Thus, the area of interest acts as an indirect communication channel. The *u-value* information is useful to help other vehicles in the next destiny decision-making. To obtain full coverage, UAVs should move to unvisited or less frequently visited locations, which means exploring cells with the least amount of pheromone. At each motion, UAVs read the *u-value* of neighbor cells, choose the one with the minor value, and update the current cell according to a certain rule. As time passes by, uncertainty about a given location increases since we are dealing with a dynamic scenario [17]. The degradation of the virtual marks over time emulates this behavior and represents the natural process of evaporation of pheromones [38]. This feature allows the locations to be covered several times during patrolling. Furthermore, different types of pheromones can be used to guide the vehicles, such as attraction and repulsion pheromones [39].

There is a variety of RTSM with different update rules, as presented in Table 1. The four heuristics Node Counting (NC), Learning Real-Time A* (LRTA*), Thrun's Value-Update Rule (TVUR), and Wagner's Value-Update Rule

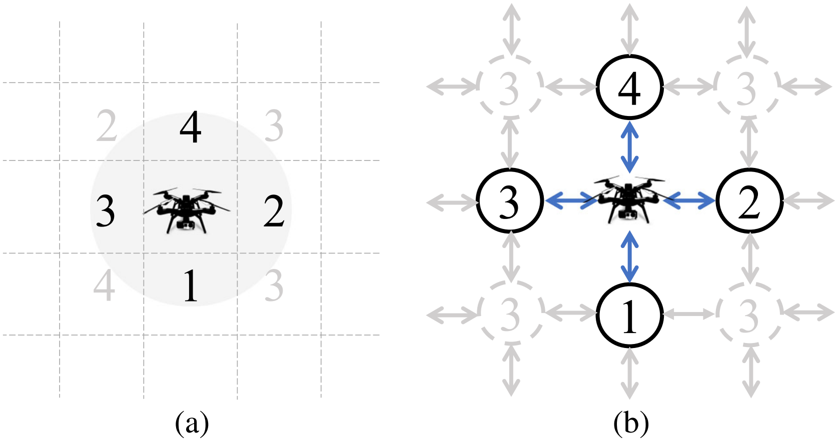


Fig. 1 The scenario is discretized into a regular grid using the approximate cellular decomposition. **a** Field of view, **b** Connected graph

Table 1 Update rules of the main real-time search methods

Method	Update rule
Node Counting	$u(s) := u(s) + 1$
Learning Real-time A*	$u(s) := u(next(s)) + 1$
Thrun's Value-Update Rule	$u(s) := max(u(s), u(next(s)) + 1$
Wagner's Value-Update Rule	*if* $u(s) \leq u(next(s))$ *then* $u(s) := u(s) + 1$

(WVUR) were originally compared by [33] in simulations with ant-robots for land coverage. Recent studies also compared, extended, and applied those heuristics in simulations and real experiments using UAVs [10, 20, 34].

The NC consists of a heuristic using a basic rule to increase the *u-value* of the visited cells. While performing the coverage, the algorithm verifies the *u-value* of the neighbor cells and selects the one with the lowest value - at the beginning, all cells contain zero. Thus, it updates the current cell by adding one unit to the original value *u(s)*. Finally, it moves the vehicle to the selected cell. All vehicles repeat this process at each iteration while moving around the scenario. The algorithm keeps running until a stop criterion, such as the minimum number of complete coverage or the predefined number of cycles. Algorithm 1 presents the Node Counting.

Algorithm 1 Node counting.

```
Input: A set of cells $\{c_1, ...c_p\}$
Output: A set of metric values $\{m_1, ...m_p\}$
repeat
    Read the u-value of the neighbor cells
    Select the cell with lowest u-value
    Update with +1 the u-value of the current cell
    Move to the selected neighbor
until Number of cycles is reached
```

LRTA* heuristic adopts a different rule to update the *u-value* of the current cell. After reading the neighbor cells and choosing the next location to move, the algorithm changes the value of the current cell *u(s)* by the value of the selected cell *u(next(s))* added by one unit. TVUR heuristic compares the value of the current cell *u(s)* with the value of the next cell to be visited *u(next(s))*. The highest value is increased by one unit and replaces the original value of the current cell *u(s)*. WVUR heuristic presents a conditional update rule where the value of the current cell *u(s)* is increased by one unit if it is not larger than the value of the next cell *u(next(s))*. Otherwise, the value of the current cell is not updated.

3 Related Work

Biological organisms are capable of adapting, evolving and surviving in wild nature, a complex environment characterized by rapid changes, high uncertainty, and limited information [40]. Such sophisticated organisms have been successful in developing intelligent strategies to perform complicated tasks. Thus, these organisms have been the inspiration for many computational systems over the years to find solutions for optimization problems [37]. Biologically-inspired algorithms based on the natural intellectual capacity have been explored by several authors to deal with the patrolling problem using UAVs.

Koenig and Liu [33] present a performance analysis of real-time search methods applied to the coordination of a team of ant-robots performing a vacuum-cleaning task. The authors consider four classical pheromone-based heuristics, including Node Counting (NC) [41], Leaning Real-Time A* (LRTA*) [42], Thrun's Value-Update Rule (TVUR) [43], and Wagner's Value-Update Rule (WVUR) [44]. The coverage task consists of cleaning an office building containing three rooms and a small waiting area. The performance metrics adopted to evaluate the studied approaches were the coverage time for single coverage and the frequency of visits and the interval between those visits for continuous coverage. The authors also performed experiments under various failure conditions, such as ant-robot malfunction and displacement from its current position, and virtual markings removal. Despite being one of the most significant studies on real-time search methods, the simulations presented in the work of [33] are restricted to ant-robots.

Performances of real-time heuristics are evaluated by [10] using simulations with land robots and real experiments with UAVs. The authors analyze four different methods, including Edge Counting [45], PatrolGRAPH* [46], NC and LRTA*. The simulations consist of a coverage time analysis, which is directly related by the authors to the energy necessary to execute coverage. Different experiment designs were conducted changing parameters such as grid size and topology, number of vehicles, and required visits. Due to the good results of the NC over other approaches in simulated experiments, the authors implemented it in a real environment containing two UAVs, a quadcopter and a hexacopter.

The decision-making process is fully centralized in an off-board station using the Robot Operating System (ROS) [47], while the dynamics and the localization of the vehicles are controlled and monitored onboard with the assistance of the ETHNOS framework [48]. The station is responsible for sending target positions to the vehicles while guiding their motions during the patrolling mission. It also receives data about the current position of the vehicles. The communication between the base and the aerial vehicles is done using a ROS/ETHNOS interface. The system is not robust to failures since the UAVs depend on the communication with the off-board station to perform their motions. Signal interference or communication loss can compromise the system and put the mission in risk. We are interested in a decentralized system to employ virtual marks of pheromone in an aerial context. The UAVs should not depend exclusively on a central station to perform their actions. Communications between vehicles should only occur sporadically to exchange map information for performance enhancement.

A reinforced random walk approach is employed as an exploration strategy for a weed mapping application by [39]. A UAV swarm has the task of detecting the presence and the number of weeds on a field. It uses attraction and repulsion vectors to guide the coverage along the area. Each swarm member divides its search area into two semi-planes, tending to choose the semi-plane ahead. The more aligned a vehicle is in the sweeping direction, the more likely it is to move in this direction. The presence of other vehicles also influences the motion of a vehicle, pushing it to less frequently covered areas. The swarm trades information among its members to avoid cover already explored locations. It is also able to call upon its members using an attraction vector, leading them to potential areas to be explored. In our previous work [20], we adopted a matrix-form internal map to mark the visits of each vehicle. We also exchange this information between the vehicles to lead them to poorly explored locations.

A cellular automata approach based on a pheromone strategy is proposed for coordinating land robots in coverage missions by [35] and further explored for controlling two quadcopters in monitoring applications [36]. The approach consists of a set of rules which determines the actions of the robots. The area of interest acts as a shared memory among robots, containing virtual marks for coordinating them. These virtual marks emulate pheromone traces left during motion and symbolize the number of visits on each location. A GCS splits the area into digital cells, tracks the location of the UAVs, and avoids crashes among them.

The pheromone evaporation is emulated by [38] to analyze the system reaction in case of loss of communication. The authors faced problems in implementing the pheromone degradation because it drastically drops the algorithm's efficiency, causing several collisions among UAVs. Another issue detected during the experiments was the elevated concentration of pheromones. When UAVs cover narrow places in the scenario, they may pass by the same cells multiple times in brief intervals. This situation increasingly raises the amount of pheromone in these places, blocking the passage of the remaining vehicles. The authors consider their approach as a decentralized system capable of adapting to different circumstances. However, UAVs do not decide which places to visit. They depend on a GCS containing a global vision of the map to process the information and indicate which locations they need to go in order to perform coverage. With our previous and current work, we are aiming at developing a solution where the UAVs use only their internal maps. The communication for information exchange among fleet members is executed only to improve the coverage performance.

Lim and Bang [18] present a waypoint coverage solution for a surveillance mission. The scenario is decomposed into a hexagonal grid containing Information Points (IP). The IPs hold a value representing certainty about the scenario locations. Each region has to be continuously monitored to keep the confidence level. When this value is sufficiently elevated, the point possesses reliable information and monitoring around those places is not necessary. However, this value decreases when a given location remains without any visitation for a while. Lower values mean not trustworthy information and demand attention for more observations to update the confidence level in these locations. A set of surrounding points determines the confidence level of each region. The distance and the interest level also influence the vehicle's decision to explore specific zones.

A cooperative coverage strategy for a fleet of small-scale UAVs looking for target locations is explored by [49]. The approach consists of merging distributed information from multiple vehicles with restricted sensing capabilities over a regular grid area. Vehicles can move in four directions or remain in the current cell while performing coverage. The mission displays false alarms triggered according to a certain probability, which can be detected by the vehicle's onboard sensors. UAVs improve sensing skills and accelerate coverage by combining local maps. We are also combining internal maps in matrix-form in our proposed approach. The vehicles share their visits among each other to enhance coverage performance.

Another surveillance algorithm is proposed by [50]. The UAVs should periodically monitor areas and minimize the interval between the visits on each location. The frequency of visits depends on the area profile and should be increased in interesting areas and reduced in risky locations. In some cases, the coverage should be avoided depending on the risk. An extension of the approach is proposed by [22], where the authors further explore preferential surveillance considering a variety of priority specifications. Since we are looking for a homogeneous distribution of coverage, we are not interested in a different frequency of visits.

Kuiper and Nadjm-Tehrani [51] present a multi-UAV coverage approach based on an adaptation of the classical Ant Colony Optimization (ACO) algorithm. As the previous pheromone-based solutions, this one indicates the places visited by the aerial vehicles through rates of pheromones. Repulsive pheromones are responsible for guiding UAVs to the unexplored zones. A variation of ACO called Chaotic Ant Colony Optimization to Coverage (CACOC) is explored by [52]. This approach integrates ACO with the chaotic dynamical system and can be employed in military surveillance. Thanks to the pheromone map with deterministic behavior, the GCS human operator can predict the coverage paths, while keeping them untraceable for external observers or possible enemies. By adopting this map, it is possible to monitor UAVs localization, eliminating the need for communication between the base and the UAV fleet for this purpose. However, the fleet still shares a virtual map to guide the motions of the vehicles. [53] employ the V-Rep simulation environment [54] for additional experiments to evaluate the performance of CACOC.

An approach exploring B-spline curves to represent coverage paths is presented by [55]. Vehicles move from left to right corner. In between, middle points form curves which should be optimized to maximize the path desirability. The algorithm launches multiple ants, which leave traces of pheromone in the control points. A Gaussian distribution function represents the pheromone concentration. The function is superposed to generate a joint distribution function, which is rescaled to build a probability density function. The main goal is to define the y-positions of the points between the two extremities to determine the coverage path.

A pheromone-based strategy combined with Genetic Algorithm (GA) is explored by [56] for coordinating a fleet of UAVs. The GA splits the area of interest and allocates the resulting rectangular sub-regions to the UAVs. Two types of pheromones are used by the authors: search and path. The former one is related to uncertainty and represents the needing coverage of a zone. The latter one marks the places covered by other UAVs, preventing crashes among them, while decreasing revisitation. The pheromone-based GA generates the sub-regions by maximizing the search pheromone and minimizing the path pheromone. The coverage trajectories performed by UAVs inside each sub-region consist of simple back-and-forth motions. Furthermore, the trajectory to move the UAV from its current position to the closest vertex of the designated sub-region is planned using A* algorithm. Our proposed approach does not intend to divide the area and restrict the motions of the vehicles to a back-and-forth pattern inside sub-regions.

Vehicles using real-time heuristics usually have a perception restricted to the surrounding areas. They can read information from neighbor cells and write in the current cell. These heuristics are known as 1-range strategies. Sampaio et al. [19] present a method to transform 1-range strategies into 0-range strategies. In these strategies, each cell needs to store information from all its neighbors. Instead of reading the neighbor cells, UAVs sense only the current cell to choose the next destination. They have a cell-like structure in their memories to store information from visited cells. They use this structure to update other cells in future visits. In our work, we adopted three performance metrics. Two of them were presented by [19]: Quadratic Mean of Intervals (QMI) and Standard Deviation of Frequencies (SDF). These metrics will be explained in detail in Section 5.

4 Proposed Approach

In our previous work [20], which is expanded here, we presented an extension of the Node Counting algorithm for patrolling missions, called NC-Drone. This extension reduces the massive number of turning maneuvers performed by the UAVs running the NC algorithm to save energy. At the same time, the NC-Drone keeps the unpredictable behavior for an external observer, which is a fundamental aspect of patrolling missions. To achieve this goal, we developed two types of NC-Drone, a centralized algorithm and a decentralized one with a few variations. In Section 4.1 we shortly walk through the previously presented approaches for the sake of comprehension.

In this paper we propose a set of novel strategies that deals with different aspects of patrolling problem trying to improve its performance. Section 4.2 describes the Watershed Strategy (WS), by which the algorithm creates clusters of cells using a topographic relief representation of the matrix. Section 4.3 explains the Time-based Strategy (TS) used to decide among ties in cells with the same number of visits by choosing the less recently visited. Section 4.4 discusses the Evaporation Strategy (ES), which reduces the amount of pheromone in places not visited for

Algorithm 2 Centralized NC-Drone.

```
Input: A set of cells $\{c_1, ...c_p\}$
Output: A set of metric values $\{m_1, ...m_p\}$
repeat
    Read the u-value of the neighbor cells
    Select the neighbor cell with lowest u-value
    if there is a tie and one of the cells is aligned with the sweep direction then
        Choose the aligned cell as the next place to be visited
    else
        Randomly choose one of the cells
    end if
    Update with +1 the u-value of the current cell
    Move to the selected neighbor cell
until Number of cycles is reached
```

a while to attract UAVs to cover these zones. Section 4.5 presents the Communication-Frequency Strategy (CFS) where UAVs reduce communication, exchanging matrix-information from time to time.

4.1 Previous Presented NC-Drone Variations

RTSM are usually 1-range strategies capable of sensing the closest surrounding cells. During coverage, these methods always choose the locations with the smallest amount of pheromone (number of visits) as the next place to be visited by the UAV. When there is more than one cell with the same minimum value, the algorithm randomly chooses among one of the cells. All cells have zero at the beginning of the mission, which causes several ties and constantly changes in the orientation of the UAVs. As stated by [27], the higher the number of turning maneuvers, the higher the energy consumption.

Centralized NC-Drone works exactly like the original Node Counting in terms of sensing neighbor cells and choosing the least visited one. But this extension is also able to identify when there is a tie between two or more cells with the minimum value. The algorithm verifies if one of these cells is aligned with the sweep direction of the UAV. In this case, the correspondent cell is selected as the next place to be visited, keeping the UAV in the same direction and avoiding unnecessary turns. If there is no aligned cell, the algorithm randomly takes the decision and sustains the unpredictable behavior. Algorithm 2 describes Centralized NC-Drone.

Decentralized NC-Drone explores a model where the UAVs have a matrix-form internal map of the area of interest to be covered. The matrix representation is straightforward since the area is decomposed into a regular grid using approximate cellular decomposition. A pair of coordinates (line and column) denotes a cell and keeps the number of visits performed in the corresponding location. Each UAV stores the visited places in its internal matrix and selects the next destinations based on the number of visits stored in the neighbor positions of its current position in the matrix. The vehicle always chooses the minor value and respects the tie rule implemented in the centralized version to maintain the sweep direction.

A matrix-based communication model can be employed to share and synchronize individual information stored in the internal matrices of the UAVs. In the simulation, the UAVs can verify the existence of other vehicles in the surrounding area, establishing a monitoring perimeter. In real flights, the vehicles can communicate with each other through wireless messages using XBee devices. These long-range devices can also communicate with microcontroller boards via a serial interface to process the exchanged information. Figure 2 illustrates the matrix-based communication model with two UAVs performing the patrolling. The UAVs detect each other presence within the communication radius and open a communication channel to trade information about the visited places and update their internal matrices. Algorithm 3 describes the Decentralized NC-Drone adopting the matrix-based communication model.

The matrix-based communication model allows the UAVs to exchange information stored in the internal matrices. Synchronization methods can be used to merge information from two different matrices into a single one or combine multiple matrices to decide the next place to go. The merging methods presented in our previous work [20] were NC-Drone MAX and NC-Drone AVG. These methods compare both matrices, position by position, updating the original matrix. Table 2 presents a brief description of the synchronization methods.

The NC-Drone MULTI combines multiple matrices when deciding each move. Using this approach, the UAVs

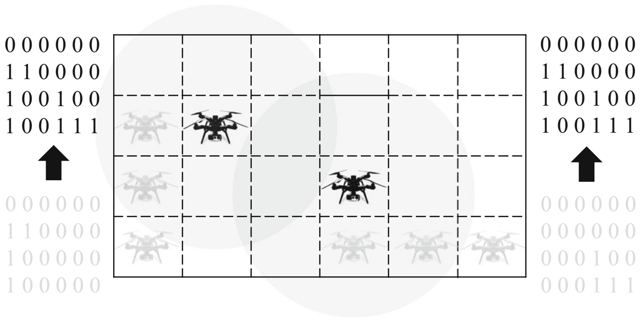


Fig. 2 Two UAVs are performing the patrolling in the scenario. When the vehicles are within the communication radius, the synchronization process starts to exchange the information stored in the current matrices (in grey). The information about the visited places of both matrices is updated (in black)

Table 2 Synchronization methods for merging and combining matrix information

Method	Description
NC-Drone MAX	Updates the matrix with the major value.
NC-Drone AVG	Computes the mean between two values and rounds the result to the nearest integer number.
NC-Drone MULTI	Superposes the matrices, but does not merge the information. UAVs can copy the matrices of other vehicles to its memory.

copy the information from each other when they are within the synchronization perimeter. They can store multiple matrices from all vehicles in the scenario. When performing a move, the UAV superposes the stored content instead of fusing the information, summing the neighbor values of its current position. Then, it chooses the least visited location as the next place to be covered. UAV updates only its matrix during coverage. Other matrices internally stored are individually updated when there is a new synchronization.

4.2 Watershed Strategy

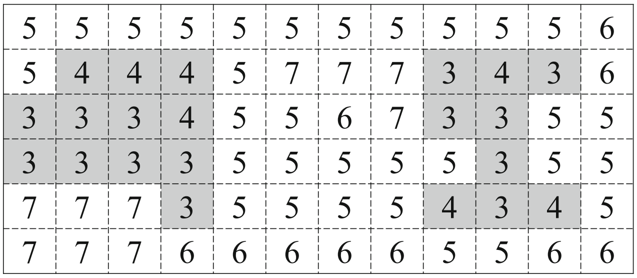
The Watershed is a mathematical morphology model widely used for segmentation in image processing field [57]. The algorithm can also be used in patrolling missions involving UAVs to represent a matrix as a topographic relief with different elevations. Depression zones correspond to the minor values in the matrix, i.e., poorly or less frequently explored areas, while higher elevations represent the greater values. A global analysis of the map instead of using only local information (neighbor cells) may be useful to guide the vehicles to the needy areas and balance coverage.

Algorithm 3 Decentralized NC-Drone.

```
Input: A set of cells $\{c_1, ...c_p\}$, matrix
Output: A set of metric values $\{m_1, ...m_p\}$, matrix updated
repeat
    Read the u-value of the neighbor cells in the matrix
    Select the neighbor cell with lowest u-value
    if there is a tie and one of the cells is aligned with the sweep direction then
        Choose the aligned cell as the next place to be visited
    else
        Randomly choose one of the cells
    end if
    Update with +1 the u-value of the cell in the matrix
    Move to the selected neighbor cell
    if sync is on and there are UAVs within the radius then
        Open a communication channel
        Merge or combine the information of the matrices
        Update the matrices with the resulting values
        Close the communication channel
    end if
until Number of cycles is reached
```

Watershed Strategy (WS) analyzes internal matrices and computes the average value of its elements. This value is rounded down and used as a threshold by the WS, as indicated in line 1 of Algorithm 4. A $clusterList$ is created to store all the resulting clusters of the matrix (line 2). Then, the algorithm chooses a random position in the matrix M whose value is under the $threshold$ (line 3). This random cell is added to a new $cluster$ (line 5), which is a structure used to store all cells belonging to the same group. The same cell cannot be part of two different clusters. A flooding process starts in the selected location, spreading to the neighboring cells (Moore neighborhood) whose values are beneath the $threshold$ (line 7 and 8). These neighbors are added to the current $cluster$ (line 9) and the propagation process repeats until there is no surrounding cell under the $threshold$. The resulting group of cells forms a $cluster$, which is added to a list of all clusters (line 12). This process forms several clusters and ends when there is no cell in the matrix under the threshold, as illustrated by Fig. 3.

The WS verifies the resulting clusters and selects the one with the larger number of cells (line 14). Then, the algorithm selects a random cell inside this cluster as the next location to be visited (line 15). The first UAV to finish the WS execution covers the cluster and sends a message to the other vehicles to prevent them from covering the same zone. A straight path with minimum turns is used by the UAV to reach the cluster zone. As can be seen in Fig. 4, the UAV in the bottom part of the scenario computes the cluster



5	5	5	5	5	5	5	5	5	5	5	6
5	4	4	4	5	7	7	7	3	4	3	6
3	3	3	4	5	5	6	7	3	3	5	5
3	3	3	3	5	5	5	5	5	3	5	5
7	7	7	3	5	5	5	5	4	3	4	5
7	7	7	6	6	6	6	6	5	5	6	6

Fig. 3 Watershed Strategy forms two clusters with a group of cells whose u-value is beneath the $threshold$ = 5. The process starts in a random cell with u-value under the $threshold$ and spreads to the neighbors until there is no cell outside a cluster

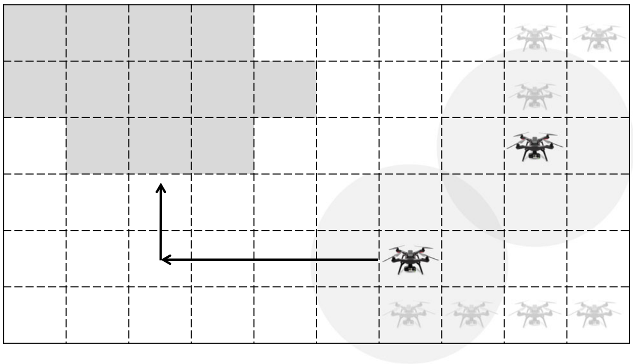


Fig. 4 Watershed Strategy computes the cluster (grey zone) when the two UAVs are within the communication radius. After synchronizing their matrices, the designated UAV follows a straight path with a turn to cover the selected zone

and goes toward it following a simple trajectory with a 90-degree turn. The other UAV remains in the surrounding area visiting the closest neighbors. Once the UAV reaches the selected location, the tendency is to remain inside the cluster exploring the whole area.

4.3 Time-based Strategy

The previously proposed NC-Drone and its variations (MAX, AVG, MULTI, WS) use the number of visits to choose the next place to be visited by UAVs. When there are two or more cells with the same minimum value, the UAV chooses the one aligned with the sweeping direction. If there is no aligned cell, the vehicle randomly chooses one of the others. The Time-based Strategy (TS) considers the time when the last visit occurs to decide among the ties during the patrolling, selecting the cell visited less recently.

Time-based Strategy (TS) can also be explored combined with the Watershed Strategy (WS). Instead of using a threshold based on the average of visits to build the clusters, the approach can employ the average time of the last visit performed in all cells as a threshold. This strategy forms clusters with a set of cells covered less recently to guide the vehicles. Table 3 presents a summary of the time-based strategies.

Algorithm 4 Watershed strategy.

```
Input: A matrix $M_{ij}$
Output: A destiny cell $c$
threshold ← Floor(Threshold(M))
Create empty clusterList
while cell $c \leftarrow$ Random ((cell $c$ in $M$ with u-value $<$ threshold) and ($c \notin clusterList$)) do
    Create new cluster
    Insert $c$ in cluster
    for all cell $c$ in cluster do
        Propagate $c$ to the neighbors
        for all ((neighbors of $c$ with u-value $<$ threshold) and
          (neighbors of $c \notin cluster$) and
          (neighbors of $c \notin clusterList$)) do
            Add neighbor to cluster
        end for
    end for
    Add cluster to clusterList
end while
Choose the largest cluster in clusterList
Returns a random cell $c$ in cluster
```

4.4 Evaporation Strategy

The uncertainty about information arises and increases over time due to the absence of visits in the scenario's locations, i.e., during the interval between two consecutive visits in a certain place. The Evaporation Strategy (ES) models the absence of visits by reducing the amount of pheromone when a place is not visited for a while. This strategy works with an evaporation factor (EF) varying from $EF = 0.1$ to $EF = 1$ in steps of 0.1. By reducing the pheromone, the less recently visited areas become more attractive to the UAVs.

The ES reduces the values stored in the matrix depending on the time of the last visit. Figure 5 illustrates the ES in three distinct cycles of simulation in a scenario represented by a matrix 3 x 3. Here we are considering only the first line highlighted in the matrix to exemplify. The last visit in each one of these places occurred at cycles 50, 40, and

Table 3 Time-based strategies for NC-Drone

Method	Description
NC-Drone TS	Selects the cell less recently visited when two or more cells have the same minimum value. It can be used with Centralized NC-Drone and Decentralized NC-Drone MAX, AVG, and MULTI.
NC-Drone WSTS	Combines the Watershed Strategy with the Time-based Strategy. It forms clusters with a set of cells less recently covered by adopting the average time of the last visit performed in all cells as a threshold.

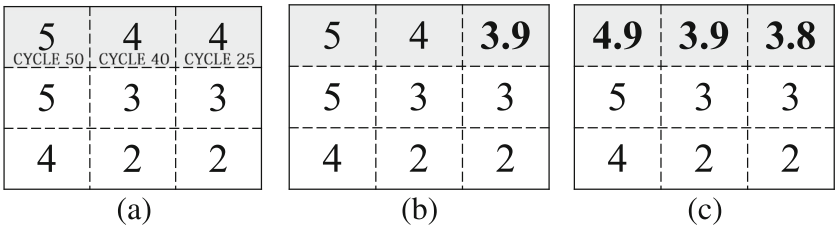


Fig. 5 Evaporation Strategy in three distinct cycles of simulation. **a** Cycle of simulation = 100, **b** Cycle of simulation = 125, **c** Cycle of simulation = 225

25, respectively. The number of visits is 5, 4, and 4 in each position. These values are consistent in the current cycle of simulation = 100 in Fig. 5a. In cycles 125 and 225, the third value is decreased by 0.1 two times, according to Fig. 5b and c. In the meantime, the first two values were also decreased at cycles 140 and 150, respectively. This example considers the pheromone dropping 0.1 at every 100 cycles in each unvisited location. The ES can explore different intervals to evaporate the pheromones, such as 50, 100, 250, 500, and 1k.

4.5 Communication-Frequency Strategy

The previously proposed approaches consider that the UAVs work with short-range communication to synchronize and exchange information. However, new technologies have been developed with increasing reach, such as XBee 3. This device works with different operating frequencies and adopts a wide variety of protocols, such as Zigbee, Thread, 802.15.4, DigiMesh, Wi-Fi, and Bluetooth. Moreover, it eliminates the need for an external microcontroller to process information. The range radius varies from 4000ft (1200m) to 2 miles (3200m) depending on the version, standard or PRO [58]. This technology has been successfully integrated with UAVs, such as the Draganfly [59]. The Communication-Frequency Strategy (CFS) explores the idea of global range, where all UAVs are within the communication radius and can synchronize their matrices. The vehicles may explore different communication frequencies, reducing the number of exchanged messages compared with the centralized approach. Each UAV stores two matrices. The matrix A indicates the last synchronization, while the matrix B is a copy which is constantly updated when the vehicle moves. All vehicles compute the difference between the matrix B and A at every synchronization. The resulting matrix C corresponds to the visits performed by the UAV. The vehicles broadcast their resulting matrices, updating matrix A with the sum of all matrices C. These matrices correspond to the visits performed by the other UAVs. Then, matrix A is copied to matrix B and the UAVs continue to count the visits in matrix B until the next synchronization.

5 Experiments and Results

The proposed approaches were implemented in the NetLogo platform, a multi-agent programmable modeling environment. UAVs were modeled as agents and were able to navigate and interact with each other in a 50 × 50 grid scenario. Following the centralized approach, agents were able to read and write information in the cells of the scenario - the cells are also known as patches in the platform. In the decentralized variations, agents can exchange matrix information by communicating among themselves.

We employed four UAVs during the experiments. All vehicles start the patrolling task at the lower left corner of the area and move to a cell in a single cycle of simulation. Thirty simulations with 10.000 cycles were executed for each approach. We compute the mean (M) and the standard deviation (SD) of the simulations and employ the Student's T-Test to check if there is a statistical difference among the approaches. We also present the best and worst results of each strategy.

Three different performance metrics were adopted during the experiments, such as Quadratic Mean of the Intervals (QMI), Standard Deviation of the Frequencies (SDF), and Number of 90° Turning Maneuvers (NTM). The performance metrics highlight different application requirements depending on the main goal of the task. We are seeking a uniform spatial and temporal distribution of visits while minimizing the number of turns. Furthermore, we need to keep the unpredictable behavior of the path for an external observer or intelligent target while minimizing energy consumption.

The QMI and SDF performance metrics were introduced by [60] for patrolling missions. QMI corresponds to frequency-regularity equilibrium, an application requirement which combines frequency of visits and evenness of intervals between these visits. This performance metric balances values of average, maximum, and standard deviation of intervals. Equation 1 presents QMI performance metric:

$$QMI = \left[\frac{1}{N_{intervals}} \sum_{x \in cells} \left(\sum_{j=1}^{visits(x)+1} (i_j^x)^2 \right) \right]^{1/2} \quad (1)$$

where i_j^x represents the intervals between the visits j to the cells x and $N_{intervals}$ corresponds to the total number of intervals.

SDF computes the distribution equality of visits over the cells of the scenario, satisfying the application requirement of uniform visitation. The lower the value, the more uniform the frequency is. Zero is the ideal condition, where all cells have the same frequency of visits. QMI and SDF do not depend on each other, having potential applications separately. Equation 2 illustrates the SDF performance metric:

$$SDF = \left[\frac{1}{|cells|} \sum_{x \in cells} \left(freq(x) - F_{avg} \right)^2 \right]^{1/2} \tag{2}$$

where $freq(x)$ corresponds to the frequency of visits on each cell x, F_{avg} is the average frequency of all cells and $cells$ is the total number of cells present in the scenario.

NTM is usually correlated with the energy spent to perform a coverage path. Every time a UAV needs to execute a turning maneuver, it must decelerate, rotate, and accelerate again. These maneuvers increase covering time and, consequently, the necessary energy to carry out a mission. Thus, the lower the NTM, the lower the energy consumption is. Equation 3 shows the NTM performance metric:

$$NTM = \sum_{u \in uavs} \left(\sum_{k=1}^{turns(u)} t_k^u \right) \tag{3}$$

where t_k^u denotes the number of turning maneuvers performed by each UAV u.

The Centralized NC-Drone and its decentralized variations (MULTI, MAX, and AVG) were originally experimented and evaluated in our previous work [20], regarding the same set of metrics.

One can download all algorithms implemented in NetLogo and the complete set of results of the simulations through the link https://github.com/tauacabreira/NCDrone.

5.1 Watershed Strategy

In this section, experiments performed with the Watershed Strategy (WS) applied to the original approaches are presented. Table 4 presents the results for these strategies considering the QMI metric with the mean (M), standard deviation (SD), the best and the worst results.

The Centralized Watershed strategy outperforms the original centralized version with a difference considered to be statistically significant. Furthermore, the Watershed strategy drastically reduced the SD in 76%. With a low SD, we can provide a more stable behavior for the algorithm, as can be seen in the best and the worst results in the last two columns of the table. The decentralized variations (MULTI, MAX, and AVG) present no improvements by adopting the Watershed strategy and its results have been omitted from the aforementioned table.

Moreover, the QMI improvement in the centralized algorithm provided by the novel strategy does not impact neither negatively nor positively the SDF metric. In other words, there is no significant change in statistical terms in the results achieved by the centralized and distributed variations of the algorithm adopting the Watershed strategy. We also omitted here the results of the SDF metric for brevity.

The NTM metric is depicted in Table 5. One can observe an increase in the value achieved in this metric considering the Watershed strategy, which states a correlation between the QMI improvement and the elevated number of turns in those algorithms.

5.2 Time-based Strategies

In this section, we show the experiments performed with the Time-based Strategies (TS and WSTS) applied to the original approaches. Table 6 presents the results for these strategies considering the QMI metric with the mean (M), standard deviation (SD), the best and the worst results.

Both TS and WSTS outperform the original centralized version, presenting a difference considered to be statistically significant. Furthermore, the Time-based Watershed (WSTS) strategy drastically reduced the SD in 88%. As we previously mentioned, with a low SD, we can provide a more stable behavior for the algorithm, as can be seen in the best and the worst results in the last two columns of the table. Once again, the decentralized variations (MULTI, MAX, and AVG) present no improvements by adopting the Time-based strategy and have no impact in the SDF metric, whose results were omitted from the table.

Table 4 QMI results for the Watershed Strategy

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	772.33	33.24	737.95	860.15
Centralized WS	**745.87**	**7.96**	**736.84**	**764.53**

The results shown in bold text empathize the best results on the table to provide an easier reading

Table 5 NTM results for the watershed strategy

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	**5.7k**	**523.18**	**4.0k**	**6.5k**
Centralized WS	7.4k	133.96	7.1k	7.7k

The results shown in bold text empathize the best results on the table to provide an easier reading

Table 6 QMI results for the Time-based Strategy

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	772.33	33.24	737.95	860.15
Centralized TS	752.39	29.74	718.46	840.64
Centralized-WSTS	**747.30**	**4.12**	**739.50**	**754.56**

The results shown in bold text empathize the best results on the table to provide an easier reading

The NTM metric performance results is illustrated in Table 7. As one can see, Centralized-TS emerges as a promising strategy once it improves the QMI performance while keeping the SDF and the NTM metrics stable. In these two metrics, the results are similar to the ones in the original Centralized NC-Drone.

5.3 Evaporation Strategies

We have run four different experiments for each algorithm with intervals of 100, 250, 500, and 1k between each evaporation. Since the evaporation process adds a computational cost to the original algorithm every time it runs, we empirically tested four values to evaluate the performance of the ES and finds out the ideal interval. The evaporation factor (EF) is 0.1, which means that the amount of pheromone drops 0.1 at every x intervals. Table 8 presents the ES results considering the QMI metric in the centralized NC-Drone and its decentralized variations.

The ES improved the Centralized NC-Drone in the QMI metric in all intervals with a difference considered extremely significant in statistical terms. Centralized-ES-100 presented an improvement of around 10% over the original algorithm. Moreover, the ES reduced the standard deviation (SD) around 75% in all runs, making the algorithm more stable. The ES also improved the Decentralized NC-Drone MULTI (NC-D. MULTI) in all runs and the Decentralized NC-Drone MAX (NC-D. MAX) in the first interval (MAX-ES-100). Despite the improvement of around 5% in the D. MULTI, the D. MAX is still the best algorithm among the decentralized variations in the QMI metric. There was no improvement in the Decentralized NC-Drone AVG (NC-D. AVG).

Table 7 NTM results for the Time-based Strategy

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	5.7k	523.18	4.0k	6.5k
Centralized TS	**5.4k**	**449.35**	**4.2k**	**6.3k**
Centralized-WSTS	8.1k	92.18	7.9k	8.3k

The results shown in bold text empathize the best results on the table to provide an easier reading

Table 8 QMI results for the evaporation strategies

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	772.33	33.24	737.95	860.15
Centralized-ES-100	**700.97**	**8.59**	**683.4**8	**717.42**
Centralized-ES-250	702.49	8.22	683.69	723.09
Centralized-ES-500	706.60	8.66	689.79	726.23
Centralized-ES-1k	705.69	8.32	682.12	719.87
NC-D. MULTI	849.80	16.24	816.43	877.47
MULTI-ES-100	806.84	14.18	782.82	844.45
MULTI-ES-250	807.71	15.23	767.34	833.86
MULTI-ES-500	809.62	12.45	790.97	836.98
MULTI-ES-1K	820.37	15.89	795.05	875.16
NC-D. MAX	797.58	10.97	787.56	825.92
MAX-ES-100	**789.98**	**10.59**	**773.25**	**827.16**
MAX-ES-250	794.31	9.40	777.21	816.73
MAX-ES-500	794.13	8.87	773.47	810.93
MAX-ES-1K	793.98	7.98	782.45	811.41
NC-D. AVG	795.57	9.60	780.80	828.88
AVG-ES-100	792.50	9.27	777.38	812.09
AVG-ES-250	793.26	9.13	773.47	810.61
AVG-ES-500	791.13	8.60	776.04	810.62
AVG-ES-1K	794.95	10.60	776.01	824.40

The results shown in bold text empathize the best results on the table to provide an easier reading

The QMI results for the Centralized NC-Drone and D. MULTI worsened as the interval grows. The majority of cells do not stay without a visit for periods longer than 1k cycles of simulation. In this way, an evaporation process with this interval is rarely triggered and does not improve the algorithm's performance as it does with smaller intervals.

While this strategy improves both Centralized NC-Drone and NC-D. MULTI in the QMI metric, the SDF metric performance drops in all runs by adopting the ES. On the other hand, there was a slight improvement in the NC-D. MAX and NC-D. AVG. Despite this improvement, the MULTI remains the best algorithm among the decentralized variations in the SDF metric. The improvements provided by ES in the QMI metric also present a negative impact on the NTM metric. The total number of turning maneuvers almost triplicate in the centralized version of the NC-Drone, drastically decreasing the performance in this metric. Once more, we can observe the trade-off and correlation between the QMI and NTM metrics. It is necessary to perform a large number of turns to obtain a more spatially distributed patrolling. The number of turns also increase in the MAX and AVG, but not so much as in the Centralized NC-Drone and MULTI. We have omitted here the results of both SDF and NTM metrics.

5.4 Communication-Frequency Strategies

We have run five different experiments for each algorithm with intervals of 50, 100, 250, 500, and 1000 cycles between every synchronization among the vehicles. Since communication presents an additional cost to the NC-Drone, we investigated the impact of different communication ranges empirically selected to evaluate the performance of the CFS. Figure 6 shows the results of the QMI metric considering the CFS.

The original Centralized NC-Drone behaves as a CFS-1, where all vehicles have access to all available information in the scenario at every step of the simulation. Thus, we can set Centralized NC-Drone as the upper bound solution for the experiments. CFS-50 presents the best results regarding the QMI metric considering this strategy. It obtains an outcome equivalent to the one presented by the Centralized NC-Drone, even sharing the matrices only at every 50 cycles. According to the Student's T-test, the difference between the two approaches is considered to be not statistically significant. However, as the interval between synchronizations doubles, metric performance falls between 7% and 8%. This behavior can be seen in all variations. The standard deviation also increases as the interval grows.

The results of this strategy regarding the SDF metric are illustrated in Fig. 7. In this metric, the performance drop of the CFS is not linear and increases more and more as the interval grows.

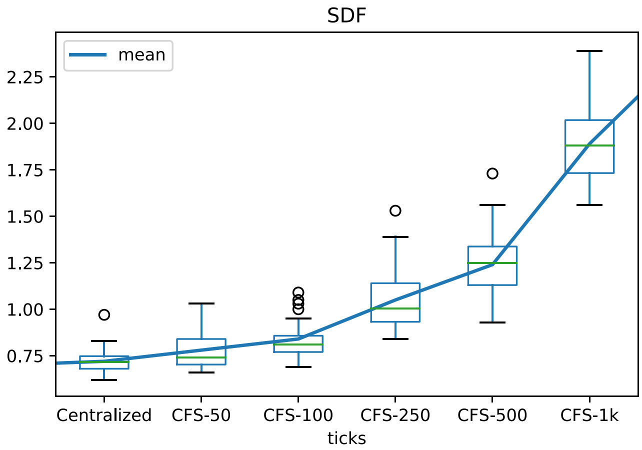


Fig. 7 SDF results for the Communication-Frequency strategies

The NTM results for the experiment with the CFS strategy is presented in Fig. 8. In this strategy, the number of turning maneuvers increases compared to Centralized NC-Drone. Considering only the CFS strategies, the NTM continues to drop as the interval of synchronization grows from CFS-100 to CFS-1000. In these cases, the vehicles tend to follow straight paths for more extended periods and perform less turning maneuvers, since they have less information available about the visits. A UAV ends up visiting areas already covered by other vehicles, once in its internal map, these areas are not covered yet. This particular behavior leads to a performance drop of the QMI metric. Thus, it is necessary to perform more turns to obtain a more uniform distribution of visits.

Moreover, one can verify that there is a correlation between the presented metrics. The Pearson Correlation coefficient between QMI and NTM is -0.7713, which is a strong negative correlation, meaning that whenever QMI values grow, NTM tends to fall, and vice versa. The coefficient between SDF and NTM is -0.88963, and there is also a strong negative correlation. However, the coefficient between QMI and SDF is positive (0.9716), which means that the two metrics tend to grow together over time.

5.5 Combination of Strategies

Considering the improvements provided by the individual algorithms CFS, WS, TS, and ES in previous sections,

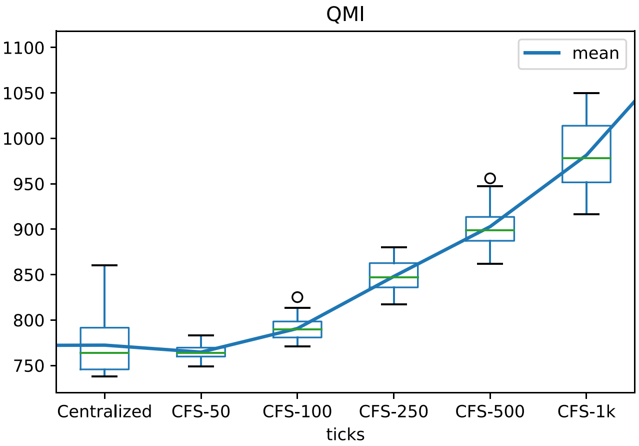


Fig. 6 QMI results for the Communication-Frequency Strategies

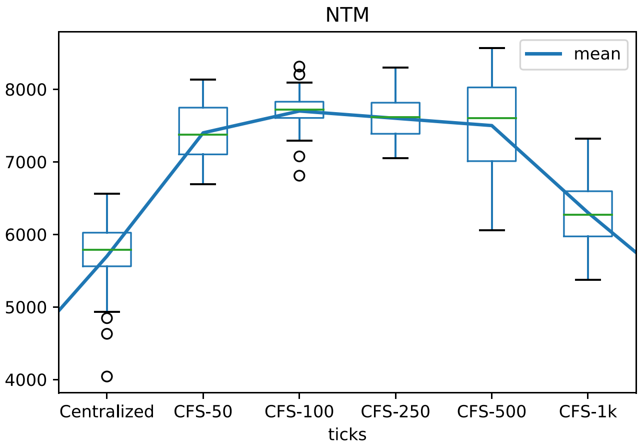


Fig. 8 NTM results for the Communication-Frequency Strategies

Table 9 QMI results for the combination of strategies

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	772.33	33.24	737.95	860.15
CFS-WS-50	747.41	5.58	736.92	756.99
CFS-WS-100	772.92	8.50	756.09	797.12
CFS-WS-ES-50	746.65	6.74	730.91	758.80
CFS-WS-ES-100	763.29	6.30	750.81	782.08
CFS-WS-ES-Time-50	742.64	4.56	736.29	751.75
CFS-WS-ES-Time-100	761.89	7.17	750.84	781.67
CFS-WSTS-ES-50	743.79	3.77	734.30	750.69
CFS-WSTS-ES-100	760.15	5.69	748.47	772.23
CFS-WSTS-ES-Time-50	**739.40**	**3.62**	**731.20**	**747.21**
CFS-WSTS-ES-Time-100	760.64	6.95	745.13	784.05

The results shown in bold text empathize the best results on the table to provide an easier reading

Table 11 NTM results for the combination of strategies

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	**5.7k**	**523.18**	**4.3k**	**6.6k**
CFS-WS-50	7.7k	188.25	7.3k	8.0k
CFS-WS-100	7.9k	243.28	7.4k	8.4k
CFS-WS-ES-50	9.1k	199.17	8.7k	9.5k
CFS-WS-ES-100	9.4k	235.31	9.0k	9.9k
CFS-WS-ES-Time-50	8.9k	149.82	8.5k	9.1k
CFS-WS-ES-Time-100	9.3k	236.48	8.8k	9.7k
CFS-WSTS-ES-50	9.5k	187.33	9.0k	9.9k
CFS-WSTS-ES-100	9.7k	235.46	9.3k	10.1k
CFS-WSTS-ES-Time-50	9.6k	231.12	9.2k	10.2k
CFS-WSTS-ES-Time-100	9.8k	239.99	9.3k	10.4k

The results shown in bold text empathize the best results on the table to provide an easier reading

we present here a few combinations and a final algorithm composed of all strategies together:

- CFS-WS-X explores global communication at regular intervals to exchange information and computes clusters of less visited cells to balance coverage;
- CFS-WS-ES-X combines the previous strategy with Evaporation Strategy reducing the amount of pheromone in certain places not visited for a while;
- CFS-WS-ES-Time-X includes tie-break between two places with the same minimum number of visits, selecting the less recently visited place;
- CFS-WSTS-ES-X adopts a Watershed Strategy based on the time of the last visit, instead of using the number of visits to form the clusters; and
- CFS-WSTS-ES-Time-X combines the last two strategies with the Evaporation Strategy. In all strategies, the evaporation factor drops 0.1 at every 100 cycles of simulation.

Table 10 SDF results for the combination of strategies

Approaches	M	SD	BEST	WORST
Centralized NC-Drone	**0.72**	**0.07**	**0.62**	**0.97**
CFS-WS-50	0.73	0.05	0.63	0.85
CFS-WS-100	0.79	0.06	0.68	0.92
CFS-WS-ES-50	0.84	0.05	0.76	1.02
CFS-WS-ES-100	0.95	0.08	0.85	1.20
CFS-WS-ES-Time-50	0.85	0.05	0.75	0.98
CFS-WS-ES-Time-100	0.92	0.07	0.81	1.11
CFS-WSTS-ES-50	0.93	0.06	0.83	1.05
CFS-WSTS-ES-100	0.97	0.08	0.84	1.15
CFS-WSTS-ES-Time-50	1.02	0.10	0.86	1.30
CFS-WSTS-ES-Time-100	1.09	0.10	0.91	1.38

The results shown in bold text empathize the best results on the table to provide an easier reading

Table 9 presents the results of the QMI metric for the combination of strategies. The achieved results are slightly improved as more combinations of strategies are adopted. The standard deviation also drops as the combinations are formed, making the solution more stable. Combinations, containing the tie-break based on time, present the best results in the QMI metric, CFS-WS-ES-Time-X, and CFS-WSTS-ES-Time-X. These solutions presented an improvement of 4% and 4.5% over the Centralized NC-Drone, despite sharing matrix information only at every 50 cycles of simulation. The combined solution with a larger interval of 100 cycles also overcomes the centralized algorithm.

The results obtained by the CFS-WS-50 are similar to the ones obtained by the Centralized NC-Drone in the SDF metric. However, as the combination of strategies improves QMI results, SDF scores become slightly worse. The SDF metric varies from 0.73 in the CFS-WS-50 to 1.02 in the CFS-WSTS-ES-Time-50. Since QMI and SDF are independent metrics from each other, it is possible to choose the most appropriate combination of strategies depending on the application requirement. The NTM metric also presents a performance drop as more strategies are combined, varying from $7.7k$ in the CFS-WS-50 to $9.6k$ in the CFS-WSTS-ES-Time-50. Once more, a larger number of turning maneuvers is necessary to obtain uniform patrolling. Tables 10 and 11 show the SDF and NTM results considering the combination of strategies.

6 Conclusion

This paper proposed a set of strategies aiming at improving the performance of our previously proposed NC-Drone algorithm. The novel strategies dealt with important aspects of the patrolling problem with multi-UAVs, such as time, uncertainty, and communication. The approaches improved the QMI metric and drastically reduced the standard deviation, providing a more stable behavior for the algorithm.

There was a trade-off and a correlation between QMI and NTM metrics in all strategies. It was necessary to perform a large number of turns to obtain a more spatially distributed patrolling. The CFS obtained an outcome equivalent to the one presented by the Centralized NC-Drone, even sharing the matrices only at every 50 cycles. The achieved results are slightly improved as more combinations of strategies are adopted. The combination of all strategies was the CFS-WSTS-ES-Time-50 algorithm. This algorithm reduced the communication frequency by 50 times and outperformed the centralized version of the NC-Drone in 4.5%.

As future work, we intend to apply a Genetic Algorithm to set different parameters for the combination of strategies to improve the algorithm's performance, such as the communication frequency, the evaporation factor, and the clusters threshold.

References

1. Lottes, P., Khanna, R., Pfeifer, J., Siegwart, R., Stachniss, C.: UAV-Based crop and weed classification for smart farming. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3024–3031. IEEE (2017)
2. Barrientos, A., Colorado, J., del Cerro, J., Martinez, A., Rossi, C., Sanz, D., Valente, J.: Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. J. Field Robot. **28**(5), 667–689 (2011). ISSN 1556-4959
3. Maza, I., Caballero, F., Capitán, J., Martínez-de Dios, J.R., Ollero, A.: Experimental results in multi-UAV coordination for disaster management and civil security applications. J. Intell. Robot. Syst. **61**(1-4), 563–585 (2011)
4. Pham, H.X., La, H.M., Feil-Seifer, D., Deans, M.: A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6648–6653, IEEE (2017)
5. Casbeer, D.W., Kingston, D.B., Beard, R.W., McLain, T.W.: Cooperative forest fire surveillance using a team of small unmanned air vehicles. Int. J. Syst. Sci. **37**(6), 351–360 (2006)
6. Renzaglia, A., Reymann, C., Lacroix, S.: Monitoring the Evolution of Clouds with UAVs./ In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 278–283. IEEE (2016)
7. Hefferan, B., Cliff, O.M., Fitch, R.: Adversarial patrolling with reactive point processes. In: Proceedings of the Australasian Conference on Robotics and Automation (ACRA), Brisbane, Australia, pp. 5–7 (2016)
8. Basilico, N., Carpin, S.: Deploying teams of heterogeneous UAVs in cooperative two-level surveillance missions. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 610–615. IEEE (2015)
9. Andersen, H.L.: Path planning for search and rescue mission using multicopters, Master's Thesis, Institutt for teknisk kybernetikk, Norway (2014)
10. Nattero, C., Recchiuto, C.T., Sgorbissa, A., Wanderlingh, F.: Coverage Algorithms for Search and Rescue with UAV Drones. In: Workshop of the XIII AI*IA Symposium on Artificial Intelligence (2014)
11. Cesetti, A., Frontoni, E., Mancini, A., Ascani, A., Zingaretti, P., Longhi, S.: A visual global positioning system for unmanned aerial vehicles used in photogrammetric applications. J. Intell. Robot. Syst. **61**(1-4), 157–168 (2011)
12. Choset, H.: Coverage for robotics – a survey of recent results. Ann. Math. Artif. Intell. **31**(1), 113–126 (2001). ISSN 1573-7470
13. Chevaleyre, Y.: Theoretical analysis of the multi-agent patrolling problem. In: Intelligent Agent Technology, 2004. Proceedings. IEEE/WIC/ACM International Conference on (IAT 2004), pp. 302–308. IEEE (2004)
14. Vincent, P., Rubin, I.: A framework and analysis for cooperative search using UAV swarms. In: Proceedings of the 2004 ACM Symposium on Applied Computing, SAC '04, pp. 79–86, New York, ACM. ISBN 1-58113-812-1
15. Stalmakou, A.: UAV/UAS path planning for ice management information gathering. Master's Thesis, Institutt for teknisk kybernetikk, Norway (2011)
16. Acevedo, J.J., Arrue, B.C., Maza, I., Ollero, A.: Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions. J. Intell. Robot. Syst. **70**(1), 329–345 (2013). ISSN 1573-0409
17. Lim, S., Bang, H.: Waypoint guidance of cooperative UAVs for intelligence, surveillance, and reconnaissance. In: 2009 IEEE International Conference on Control and Automation, Pages 291–296. IEEE (2009)
18. Lim, S., Bang, H.: Waypoint planning algorithm using cost functions for surveillance. Int. J Aeronaut. Space Sci. **11**(2), 136–144 (2010)
19. Sampaio, P., Sousa, R., Rocha, A.: New patrolling strategies with short-range perception. In: Proceedings of the 2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (2016)
20. Cabreira, T.M., Kappel, K.S., Ferreira, P.R., de Brisolara, L.B.: An energy-aware real-time search approach for cooperative patrolling missions with multi-UAVs. In: 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE), pp. 254–259. IEEE (2018)
21. Araujo, J.F., Sujit, P.B., Sousa, J.B.: Multiple UAV area decomposition and coverage. In: 2013 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), pp. 30–37. IEEE (2013)
22. Ramasamy, M., Ghose, D.: A heuristic learning algorithm for preferential area surveillance by unmanned aerial vehicles. J. Intell. Robot. Syst. **88**(2-4), 655–681 (2017)
23. Maza, I., Ollero, A.: Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. In: Distributed Autonomous Robotic Systems 6, pp. 221–230. Springer, Japan (2007)
24. Torres, M., Pelta, D.A., Verdegay, J.L., Torres, J.C.: Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. Expert Syst. Appl., pp. 441–451, 2016. ISSN 0957-4174
25. Jiao, Y.-S., Wang, X.-M., Chen, H., Li, Y.: Research on the coverage path planning of UAVs for polygon areas. In: 2010 5Th

IEEE Conference on Industrial Electronics and Applications, pp. 1467–1472. IEEE (2010)

26. Li, D., Wang, X., Sun, T.: Energy-optimal coverage path planning on topographic map for environment survey with unmanned aerial vehicles. Electron. Lett. **52**(9), 699–701 (2016)
27. Di Franco, C., Buttazzo, G.: Coverage path planning for UAVs photogrammetry with energy and resolution constraints. Journal of Intelligent & Robotic Systems, pp. 1–18 (2016)
28. Milech Cabreira, T., Di Franco, C., Ferreira, P.R. Jr., Buttazzo, G.C.: Energy-aware spiral coverage path planning for UAV photogrammetric applications. IEEE Robot. Autom. Lett. **3**(4), 3662–3668 (2018). ISSN 2377-3766. https://doi.org/10.1109/LRA.2018.2854967
29. Balampanis, F., Maza, I., Ollero, A.: Area decomposition, partition and coverage with multiple remotely piloted aircraft systems operating in coastal regions. In: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 275–283. IEEE (2016)
30. Balampanis, F., Maza, I., Ollero, A.: Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 617–624. IEEE (2017)
31. Balampanis, F., Maza, I., Ollero, A.: Coastal areas division and coverage with multiple UAVs for remote sensing. Sensors **17**(4), 808 (2017)
32. Balampanis, F., Maza, I., Ollero, A.: Area partition for coastal regions with multiple UAS. J. Intell. Robot. Syst. **88**(2-4), 751–766 (2017)
33. Koenig, S., Liu, Y.: Terrain coverage with ant robots: a simulation study. In: Proceedings of the Fifth International Conference on Autonomous Agents, pp. 600–607 (2001)
34. Cabreira, T., Ferreira, P.R. Jr.: Terrain coverage with UAVs: Real-time search and geometric approaches applied to an abstract model of random events. In: Proceedings of the 13rd Latin American Robotics Symposium. IEEE (2016)
35. Zelenka, J., Kasanickỳ, T.: Insect pheromone strategy for the robots coordination. Appl. Mech. Mater. **613**, 163–171 (2014)
36. Zelenka, J., Kasanickỳ, T.: Outdoor UAV Control and Coordination System Supported by Biological Inspired Method. In: 2014 23Rd International Conference on Robotics in Alpe-Adria-Danube Region (RAAD), pp. 1–7 (2014)
37. Floreano, D., Mattiussi, C.: Bio-Inspired Artificial Intelligence: Theories, Methods, and Technologies. MIT Press (2008)
38. Zelenka, J., Kasanickỳ, T.: Insect pheromone strategy for the robots coordination - reaction on loss communication. In: 2014 IEEE 15Th International Symposium on Computational Intelligence and Informatics (CINTI), pp. 79–83. IEEE (2014)
39. Albani, D., Nardi, D., Trianni, V.: Field coverage and weed mapping by UAV swarms. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4319–4325. IEEE (2017)
40. Pfeifer, R., Lungarella, M., Iida, F.: Self-organization, embodiment, and biologically inspired robotics. Science **318**(5853), 1088–1093 (2007)
41. Pirzadeh, A., Snyder, W.: A Unified Solution to Coverage and Search in Explored and Unexplored Terrains Using Indirect Control. In: Proceedings., 1990 IEEE International Conference on Robotics and Automation, 1990, pp. 2113–2119. IEEE (1990)
42. Korf, R.E.: Real-time heuristic search. Artif. Intell. **42**(2-3), 189–211 (1990)
43. Thrun, S.B.: Efficient exploration in reinforcement learning (1992)
44. Wagner, I., Lindenbaum, M., Bruckstein, A.: On-line graph searching by a smell-oriented vertex process. In: Proceedings of the AAAI Workshop on On-Line Search, pp. 122–125 (1997)
45. Koenig, S., Simmons, R.G.: Easy and hard testbeds for real-time search algorithms. In: AAAI/IAAI, Vol. 1, pp. 279–285. Citeseer (1996)
46. Cannata, G., Sgorbissa, A.: A minimalist algorithm for multirobot continuous coverage. IEEE Trans. Robot. **27**(2), 297–312 (2011)
47. 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: ICRA Workshop on Open Source Software, Vol. 3, pp. 5. Kobe (2009)
48. Piaggio, M., Sgorbissa, A., Zaccaria, R.: Programming real-time distributed multiple robotic systems. In: Robot Soccer World Cup, pp. 412–423. Springer (1999)
49. Khan, A., Yanmaz, E., Rinner, B.: Information merging in multi-UAV cooperative search. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 3122–3129. IEEE (2014)
50. Ramasamy, M., Ghose, D.: Learning-based preferential surveillance algorithm for persistent surveillance by unmanned aerial vehicles. In: 2016 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1032–1040. IEEE (2016)
51. Kuiper, E., Nadjm-Tehrani, S.: Mobility models for UAV group reconnaissance applications. In: 2006 International Conference on Wireless and Mobile Communications (ICWMC'06), pp. 33–33. IEEE (2006)
52. Rosalie, M., Danoy, G., Chaumette, S., Bouvry, P.: From random process to chaotic behavior in swarms of UAVs. In: 6Th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications (2016)
53. Rosalie, M., Dentler, J.E., Danoy, G., Bouvry, P., Kannan, S., Olivares-Mendez, M.A., Voos, H.: Area exploration with a swarm of UAVs combining deterministic chaotic ant colony mobility with position MPC. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1392–1397. IEEE (2017)
54. Rohmer, E., Singh, S.P.N., Freese, M.: V-REP: a versatile and scalable robot simulation framework. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1321–1326. IEEE (2013)
55. Cheng, C.-T., Fallahi, K., Leung, H., Tse Chi, K.s.: Cooperative path planner for UAVs using ACO algorithm with gaussian distribution functions. In: 2009 IEEE International Symposium on Circuits and Systems, pp. 173–176. IEEE (2009)
56. Paradzik, M. et al.: Multi-Agent search strategy based on digital pheromones for UAVs. In: Signal Processing and Communication Application Conference (SIU), 2016 24Th, pp. 233–236. IEEE (2016)
57. Beucher, S., Meyer, F.: The morphological approach to segmentation: the watershed transformation. Optical Engineering-New York-Marcel Dekker Incorporated- **34**, 433–433 (1992)
58. Digi International Inc. (Digi). Digi xbee3 zigbee 3. https://www.digi.com/products/embedded-systems/rf-modules/2-4-ghz-modules/xbee3-zigbee-3, 2019 [Online; accessed April-02-2019]
59. Digi International Inc. (Digi). Drone technologies disrupting industries, saving lives. https://www.digi.com/customersuccesses/draganfly, 2019 [Online; accessed April-02-2019]
60. Sampaio, P.A.: Patrulha temporal: taxonomia, métricas e novas solućões. PhD thesis Universidade Federal de Pernambuco (2013)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Kristofer S. Kappel is graduated in Computer Science at the Federal University of Pelotas (UFPel) in 2019 and then, in the same year, started as a Ph.D. student in Computer Science at Federal University of Pelotas (UFPel). He is member of the research group on Cyber-Physical Systems Engineering and his research studies focus on Domestic Robots and Autonomous Navigation.

Tauã M. Cabreira is a Ph.D student in Computer Science at the Federal University of Pelotas (UFPel). Before that, he obtained a MSc in Computational Modelling from the Federal University of Rio Grande (FURG), Rio Grande, Brazil in 2013 and he graduated in Internet Systems from the Sul-rio-grandense Federal Institute (IFSul), Pelotas, Brazil in 2010. He attended a Ph.D. International Scholarship at the Scuola Superiore Sant'anna, Pisa, Italy, from June to November 2017, working on energy models and trajectories for Unmanned Aerial Vehicles. He is member of the research group on Cyber-Physical Systems Engineering and his research interest is in Energy-aware Coverage Path Planning with Unmanned Aerial Vehicles.

João L. Marins is a Ph.D student in Computer Science at the Federal University of Pelotas (UFPel). Before that, he obtained a MSc in Electrical Engineering in the areas of Control and Signal Processing, along with the Electrical Engineer Degree at Naval Postgraduate School (NPS) in 2000. He graduated in Electrical Engineering from the University of São Paulo (USP), São Paulo, Brazil in 1991, in the areas of Automation and Control. He worked for twenty years at the Brazilian Navy Research Institute. He is member of the research group on Cyber-Physical Systems Engineering and his research interest is in energy models for quadcopters.

Lisane B. de Brisolara has received the Ph.D degree in Computer Science from Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil in 2007. In 2002, she obtained a MSc also from UFRGS and she is graduated in Computer Science from the Catholic University of Pelotas, Pelotas, Brazil, in 1999. Since 2008, she is a permanent faculty member of Federal University of Pelotas (UFPel) and has worked on several research projects, coordinating her own projects and as well as collaborating with researchers from other institutions. She stayed as a Post-Doctorate at University of York, York, United Kingdom from September 2014 to March 2015, working on modeling and simulation of wireless sensor networks. Her research interests include embedded systems and cyber-physical systems engineering.

Paulo R. Ferreira Jr. has received the Ph.D. degree in Computer Science from Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil in 2008. Before that, he obtained a MSc in Computer Science from the State University of Campinas (UNICAMP), Campinas, Brazil in 2000 and he graduated in Computer Science from the Federal University of Pelotas (UFPel), Pelotas, Brazil in 1998. He has participated in several research projects, collaborating with researchers from Federal University of Rio Grande do Sul (UFRGS) and managing his own projects. Since 2009, he is a permanent faculty member of Federal University of Pelotas (UFPel). He stayed as a Post-Doctorate at University of York, York, United Kingdom from September 2014 to March 2015, working on bio-inspired strategies for dynamic management of wireless sensor networks. His research interests also include cyber-physical systems, wireless sensor networks, and path planning for unmanned aerial vehicle.