

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO GIỮA KỲ
ĐIỆN TỬ TƯƠNG TỰ II
DỊCH TÀI LIỆU

Nhóm 49
Sinh viên 1 Tạ Hoàng Việt
20172916
Sinh viên 2 Hồ Huy Hoàng
20172581

Hà Nội, 06/2021

Mục lục

Introduction	8
1 Coordinated exploration	9
1.1 Introduction	9
1.2 Related work	9
1.2.1 Robot-to-target assignment	10
1.2.2 Utility function	11
1.3 Proposed exploration strategy	12
1.3.1 Overview	12
1.3.2 Frontier points selection	14
1.3.3 Information gain computation	15
1.3.4 Frontier points processing	16
1.3.5 Goal assignment process	24
1.3.6 Path planning and control	27
1.4 Result and discussions	28
1.4.1 Parameters tuning	29
1.4.2 Exploration strategy performances	31
1.4.3 Exploration mission using relative localization algorithm	43
1.5 Conclusion	44
2 Giao tiếp giữa các robot	46
2.1 Giới thiệu	46
2.2 Phân loại mạng	46
2.2.1 Chế độ cơ sở hạ tầng so với chế độ không có cơ sở hạ tầng	46
2.2.2 Phân loại mạng Ad Hoc	48
2.2.3 Phân loại cấu trúc liên kết mạng	50
2.3 Cấu trúc mạng	51
2.3.1 Công việc liên quan	51

2.3.2 Các tiêu chuẩn và giao thức mạng	58
2.3.3 Kết quả và thảo luận	59
2.4 Cấu trúc liên kết và chiến lược mạng cho MRS robustness	61
2.4.1 Công việc liên quan	61
2.4.2 Phương pháp giao tiếp giữa các robot	62
2.4.3 Kết quả và thảo luận	65
2.5 Conclusion	72
Conclusion	74
Tài liệu tham khảo	75

Danh sách hình vẽ

1.1	Proposed exploration process pipeline.	13
1.2	Map structure evolution during the exploration process: From 2D cells \mathcal{L} to 2D frontier cells \mathcal{F} to candidate frontier cells/points \mathcal{G} (candidate targets) to 2D target cells/points \mathcal{T}	14
1.3	Frontier cells/points selection of a 2D occupancy grid map.	15
1.4	The information gain computation of a frontier point x (in red). The hatched cells represent the 48 surrounding cells of the frontier point x . The information gain of x is: $I(x) = 25$	16
1.5	Frontier points processing example: (a) and (b) represent the frontier points of UAV_1 and UAV_2 , respectively. (c) represent the candidate targets. The process consists in removing frontier points that belong to overlapped areas, and those that are adjacent to obstacles.	18
1.6	Frontier points processing while assuming two convex map shapes in the first line and three in the second line. (a) and (d) represent two and three convex map shapes, respectively; (b) and (e) represent the frontier points in overlap (intersection); and (c) and (f) represent the obtained frontier points after processing (final shape).	20
1.7	Frontier points processing while assuming two concave map shapes in the first line and three in the second line. (a) and (d) represent two and three convex map shapes, respectively; (b) and (e) represent the frontier points in overlap (intersection); and (c) and (f) represent the obtained frontier points after processing (final shape).	22
1.8	Utility function behavior: $I(t_j) = 25, n_c = 3, d_{tot} = \sum_{k=1, k \neq i}^{n_c} (d_{min}(\mathbf{P}_k, \mathbf{t}_j))$. The average distance of other UAVs d_{tot} has a minimum value different from zero since n_c is different from zero.	24

1.9 Example of the shape function to schedule the information gain($\mathbf{I}(\mathbf{t}_k)_{t-1} = 29$, $(\mathbf{t}_g(x), \mathbf{t}_g(y)) = (1.25, -0.275)$, $\sigma_x = \sigma_y = 3$).	27
1.10 From 2D (left) to 3D (right) navigation stack. The <i>navigation z</i> is a ROS process developed to provide a control command on the z axis.	28
1.11 The impact of varying the trade off parameter λ over exploration itme.	29
1.12 Exploration time while varying the loop rate r	30
1.13 Ratio of explored space during time for one UAV.	33
1.14 Coordinated exploration using two robots. Columns 1, 2 and 3 show the evolution of the local map of the UAV ₁ , the UAV ₂ and the global map over time, respectively.	34
1.15 The evolution of candidate and final frontier points numbers during cooperative exploration.	35
1.16 Projected 2D map in a coordinated exploration with a team of one, two and three UAVs. Green, blue and red markers and arrows define, respectively, the initial position and the trajectory of UAV1, UAV2 and UAV3.	36
1.17 Goal assignment: After assigning a target to UAV ₁ , a target is assigned in a sequential manner to UAV ₂ . (a) represents the candidate frontier points with their respective information gain. (b), (c), and (d) represent, respectively, the targets assignment when: No further process is performed for the remaining candidate targets; UAV ₁ 's target is removed from the remaining candidate targets and; the information gain of the remained candidate targets are scheduled.	38
1.18 Explored and overlapped area rate using two cooperative UAVs.	39
1.19 Explored space rate with one, two and three UAVs.	40
1.20 Traveled distance by each UAV with one, two and three cooperative UAVs.	41
1.21 Traveled distance evaluation.	42
1.22 Average exploration time.	42
1.23 Two UAVs navigation using ORB-SLAM2. Rviz image (left) shows, for each UAV, the constructed 2D occupancy grid map, its estimated trajectory and the corresponding ground truth. The point cloud (middle) represent the sparse reconstruction of the environment made by each UAV. And, the green markers (right) represent the features computed by each UAV to perform localization.	43

1.24	Explored area rate evolution during exploration mission with one and two UAVs while performing ORB-SLAM2 by each UAV	44
2.1	Chế độ hạ tầng (trái) và chế độ không có hạ tầng (phải).	47
2.2	Mạng tập trung (trái), phi tập trung (giữa) và mạng phân tán (phải).	50
2.3	Mạng lưới minh họa giữa ba máy tính xách tay và một máy bay không người lái.	60
2.4	Phát sóng thông lượng được phát sóng trong giao thức mạng Ad Hoc và BATMAN.	60
2.5	Luồng dữ liệu giữa hai robot. FP và GA đứng tương ứng, để xử lý biên giới và phân công mục tiêu.	63
2.6	Sự tiến hóa vai trò trong phạm vi giao tiếp hạn chế.	64
2.7	Minh họa mạng Ad Hoc trong nhiệm vụ thăm dò.	65
2.8	Hai cấu hình ROS trong trường hợp nhiều máy. Ảnh từ [1]. .	66
2.9	Kích thước dữ liệu khi UAV trao đổi toàn bộ bản sao lưới cục bộ của họ so với các điểm biên giới của nó.	68
2.10	Tiến hóa cấu trúc liên kết mạng trong một vòng lặp với ba UAV hợp tác.	69
2.11	Hai robot hợp tác thăm dò cùng với kết nối mạng.	71
2.12	Bản đồ lưới chiếm dụng 2D toàn cầu trong nhiệm vụ thăm dò với hai UAV hợp tác.	72

Danh sách bảng

1.1	Common parameters.	31
2.1	Chế độ cơ sở hạ tầng so với chế độ không có cơ sở hạ tầng. .	47
2.2	So sánh giữa MANET, VANET, FANET và mô hình mạng của chúng tôi.	49
2.3	Ví dụ về một số tiêu chuẩn được sử dụng trong MRS.	54
2.4	Thời gian cho module giao tiếp.	70

Introduction

Chương 1

Coordinated exploration

1.1 Introduction

The exploration and mapping of large areas is an active field of research in aerial robotics. It consists in constructing a 3D model representation of the workspace as robots progress within it. Recently, the key subject in the exploration problem is the cooperative deployment of fleet of robots that promise enhanced performances compared to single robot exploration.

In this chapter, we address the problem of cooperative exploration strategies with no a- priori knowledge of the environment. We will try to answer the question: Where should each robot move next?

1.2 Related work

Recently, several works have proposed solutions for exploration using multi-robot teams to reduce the mission time and to increase the scalability [3], [26], [53]. Hence, the challenge is to have an efficient cooperation among the agents in the fleet while maintaining communication [38]. For that, existing approaches can be centralized, that is, one robot in the fleet is responsible for assigning targets [9]. In [40], a central server with increased computational resources is adopted to receive, treat, optimize and send back information to other robots. Other works, such as [54],[42], use distributed approaches where each robot chooses its own target. Another trend is to switch from individual to cooperative exploration behavior when robots are not able to converge to a local minimum at a satisfying rate [51]. In [27], authors propose to consider four possibilities taking into account robots' interactions in

order to construct a distributed map. The situations involved are: no interaction (robots are not within each others' communication range), hypothesis generation (there is an interaction between the robots but they do not know their respective location), hypothesis verification (there is an interaction between the robots and they propose an hypothesis about their location), and coordinated exploration (there is an interaction between robots and they do share their maps).

1.2.1 Robot-to-target assignment

The mapping algorithm is performed while a robot attempts to reach a target. So, for an effective environment mapping, the target should be chosen carefully. There is a wide variety of goal assignment strategies to affect one robot to a target. The majority of them are centralized and use a cost function to compute the utility of reaching a target.

In the greedy assignment [52], each robot chooses a target depending on its cost function without coordinating with other robots. Hence, one target can be visited by different robots. To solve this issue, already chosen targets can be discarded before being considered for assignment by others using Broadcast of Local Eligibility (BLE) [49] also called Iterative Assignment. But, this method does not necessarily produce the optimal solution since it depends on the order of robots.

The K-means method [44] consists in dividing the environment to explore into regions of the same number of robots, then, assigning one robot to the closest region where it will choose a target from the frontiers depending on a cost function.

The Hungarian method [16] proposed by [28], solves the worker- task assignment written in the form of $n \times n$ matrix C where $c_{i,j}$ is the cost of the task j assigned to the worker i . The optimal assignment is found with time complexity $O(n^3)$. This algorithm requires that the number of workers is equal to the number of tasks which cannot be guaranteed. Else, imaginary robots or targets can be added to satisfy the assumption, and skipped later in the selection.

Authors in [37] present an auction-based method to assign tasks to a group of robots. The distribution of tasks is accomplished by means of a first-price reverse auction which means that the auctioneer is the buyer. One task is

auctioned one at a time by priority. Then, the auctioneer selects the best bid and assigns the task to the corresponding bidder. The algorithm is well adapted to dynamic environments, where unexpected obstacles might prevent a robot from reaching its target.

In [55], [31], a genetic algorithm is used to optimally assign robot to tasks. It is an NP-hard problem since it is considered as a generalized two-dimensional multi-type bin packing problem.

Authors in [18] present a new approach called Multiple Traveling Salesman Problem (MTSP) for robot-to-target assignment in multi-robot exploration problem. It consists in clustering an environment, determining the cost of TSP distance for each pair of robot cluster, assigning goals from non empty cluster, and finally fixing goals for empty clusters. This approach was compared with greedy, iterative and Hungarian method. The MTSP presents competitive results regarding the total computational requirements and the increasing number of robots.

In [29], a comparison of some assignment strategy used for multi- robot system is done. The compared strategies include Hungarian method, Greedy, BLE method and K-means clustering. Results show that Hungarian method outperforms other approaches in majority of cases. Unlike the Hungarian method, the iterative assignment can be implemented in a distributed environment. Also the Hungarian methods are computationally heavy compared with the simple greedy algorithm which is preferred in applicable scenario.

1.2.2 Utility function

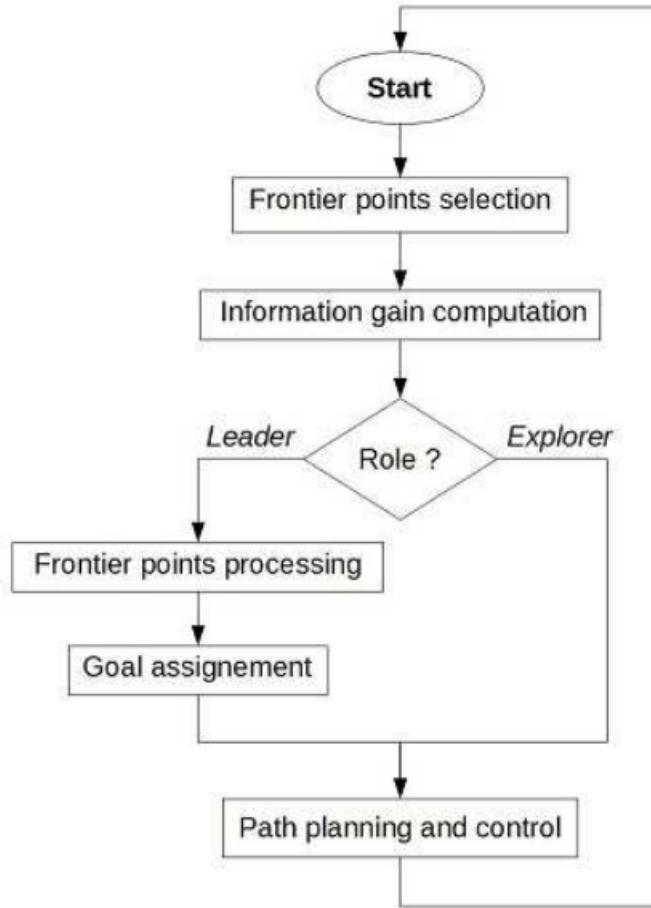
Most of the robot-to-target assignments are based on an utility function that defines the benefits that a robot have to reach this target, taking into account the mission's aim [9]. The work proposed in [5] presents a new utility function that takes into account the traveling cost to the target and the connectivity utility. This allows a trade off between minimizing the amount of exploration time and the connectivity. To speed up velocity, authors in [12] propose a rapid frontier selection technique to select goals from the robot's field of view. This approach minimizes the overall mission time by minimizing the change in velocity of the robot. Nonetheless, this approach increases the total path length traveled. In [25], maximizing the reconstructed model is favored over the mission time. Furthermore, the proposed approach solves simultaneously exploration and coverage problems in order to maximize the

completeness of the reconstructed model. Whereas in [43], the aim is the maximization of the utility of targets that minimizes the potential for overlap in information gain amongst members of the fleet. The utility of reaching a target depends basically on the aim of the mission while taking into account some additional constraints such as time, completeness of the map, limited sensor and communication range, or number of robots.

1.3 Proposed exploration strategy

1.3.1 Overview

In a Multi-UAV system, the exploration strategy needs to be cooperative to maximize the efficiency. The main objective proposed here, is to cooperatively choose specific regions to be simultaneously explored using a frontier-based approach. Commonly, this is done by selecting candidate targets and assigning them to each robot in an optimized manner. Figure 1.1 shows the proposed pipeline exploration process performed by each UAV.



Hình 1.1: Proposed exploration process pipeline.

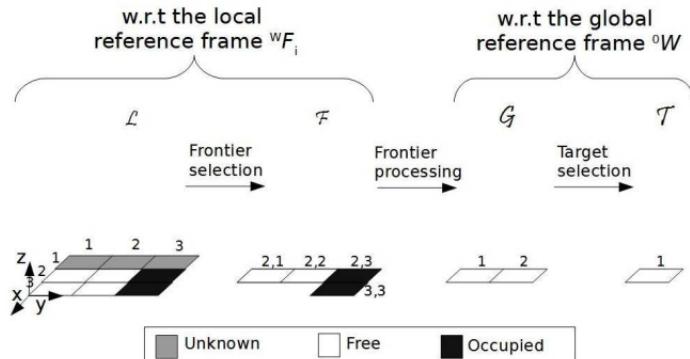
Each UAV selects the frontier points of its constructed local map during the SLAM step. Then, it computes the corresponding information gain of these points. If the UAV's role is an *explorer*, it would passively wait until receiving instructions; else, if it acts as a *leader*, it would process the collected frontier points, and would assign a target to each robot in the group. Given a target, UAVs will plan a specific path to reach it.

During these steps, the map structure evolves from a projected 2D grid map obtained from the localization and mapping module, through frontier points during the frontier selection process, then candidate targets during

the frontier processing step, to a final set of selected targets to reach. The evolution of the map structure is illustrated in Figure 1.2. Algorithm 1 describes the main steps performed during the exploration.

Algorithm 1 Exploration strategy for coordinated Multi-UAV

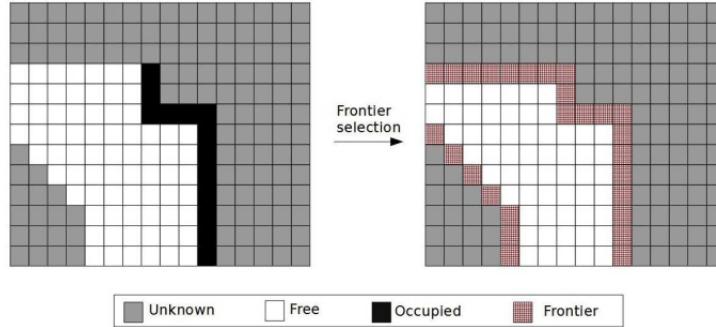
- 1: From cell $\mathcal{L}_l \in \mathcal{L}$, select frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$ and compute their respective information gain $\mathbf{I}(\mathbf{f}_{i,j})$.
 - 2: Process frontier points $\mathbf{f}_{i,j}$ to get candidate goals $\mathbf{t}_k \in \mathcal{G}$ (See Algorithm 2)
 - 3: Assign UAV $_i$ with target k (See Algorithm 2).
 - 4: Send targets to the corresponding robots.
-



Hình 1.2: Map structure evolution during the exploration process: From 2D cells \mathcal{L} to 2D frontier cells \mathcal{F} to candidate frontier cells/points \mathcal{G} (candidate targets) to 2D target cells/points \mathcal{T} .

1.3.2 Frontier points selection

The frontier selection process is used to define the frontiers of regions bounded by obstacles or unknown spaces (See Figure 1.3).



Hình 1.3: Frontier cells/points selection of a 2D occupancy grid map.

The frontier cells $\mathbf{f}_{i,j} \in \mathcal{F}$ are selected from the set of cells $\mathcal{L}(\mathcal{F} \subset \mathcal{L})$ such that they are either:

- Free \mathbf{l}_f and adjacent to unknown.
- Labeled as occupied \mathbf{l}_o . The Occupied cells \mathbf{l}_o are considered as frontier cells to be able to perform frontier processing in the next step. They could not be chosen as target and will be discarded later.

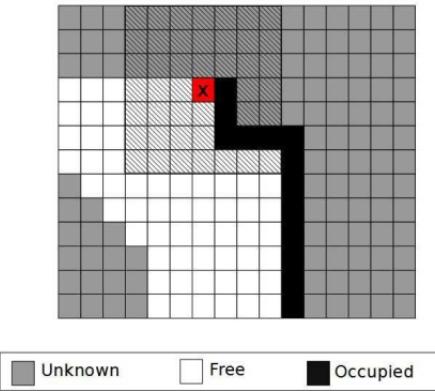
In Figure 1.2, the frontier cells are: $\mathbf{l}_f(2, 1), \mathbf{l}_f(2, 2)$, $\mathbf{l}_f(2, 3)$, and $\mathbf{l}_f(3, 3)$. Thus, for a cluster \mathcal{C} containing UAV_i , the frontiers are:

$$\mathcal{F} = \{\mathbf{f}_{i,1}(2, 1), \mathbf{f}_{i,2}(2, 2), \mathbf{f}_{i,3}(2, 3), \mathbf{f}_{i,4}(3, 3)\} \quad (1.1)$$

1.3.3 Information gain computation

In frontier-based exploration approaches, only cells adjacent to unknown ones may be defined as candidate frontier points and are likely to be chosen as targets. Thereby, the information gain is associated to each of them in order to estimate the utility of reaching each frontier. This corresponding information gain can be defined in different manners depending on the mission purpose. Authors in [10] propose to use a probability function to reduce an assigned constant value taking into account the relative distance to the UAV's pose. This strategy is general and does not take into account the updated explored cells. The approach proposed in [25] affects, to the information gain, the number of unknown and not occluded cells in the view frustum of the target. This method depends on the real estimate of information gained when visiting the considered pose. However, it requires more computation.

In the proposed strategy, the information gain is allocated so that it defines the amount of unknown cells surrounding the target (See Figure 1.4). It is a non-metric value that counts the number of cells labeled as unknown l_u from the 48 cells around the frontier point.



Hình 1.4: The information gain computation of a frontier point x (in red). The hatched cells represent the 48 surrounding cells of the frontier point x . The information gain of x is: $I(x) = 25$.

1.3.4 Frontier points processing

All frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$ of UAV _{i} in the cluster \mathcal{C} with $i \in [1..n_c]$, are collected. Points in \mathcal{F} are then processed using Algorithm 2 to get candidate frontier points considered as candidate targets $\mathbf{t}_k \in \mathcal{G}$ with $k \in [1..n_g]$ (See Figure 1.2)

Algorithm 2 Frontier processing algorithm.

Input: Frontier points $\mathbf{f}_{i,j} \in \mathcal{F}$ of UAV _{i} with $i \in [1..n_c]$.

Output: Candidate targets \mathcal{G}

- 1: $p_u = \cup_{i=1}^{n_c} \mathbf{f}_{i,j}$.
 - 2: $p_i = \cap_{i=1}^{n_c} \mathbf{f}_{i,j}$.
 - 3: $\mathcal{G} = p_u \setminus p_i$.
 - 4: Delete the obstacle frontier points $\mathbf{f}_{i,j}(x, y) = \mathbf{l}_o(x, y)$ from \mathcal{G} .
 - 5: **return** \mathcal{G} .
-

Figure 1.5 shows an example of frontier processing with two UAVs' map

to get the candidate targets. The obstacle frontier points $\mathbf{f}_{i,j}(x, y) = \mathbf{l}_o(x, y)$ – labeled as occupied – are only kept to compute the intersection of frontier points. Only the free frontier cells \mathbf{l}_f can be considered as candidate target.

When using local frontier points instead of local maps, the frontier process replaces the map matching process where the aim is to clear overlapping areas. Hence, in the frontier processing step, the frontier points that belong to overlapping areas are cleared. Therefore, using frontier points allows important memory saving. To compute the frontier points that belong to the overlapped areas (Steps 1, 2 and 3 in Algorithm 2), we propose two approaches with convex shapes and concave shapes assumptions.

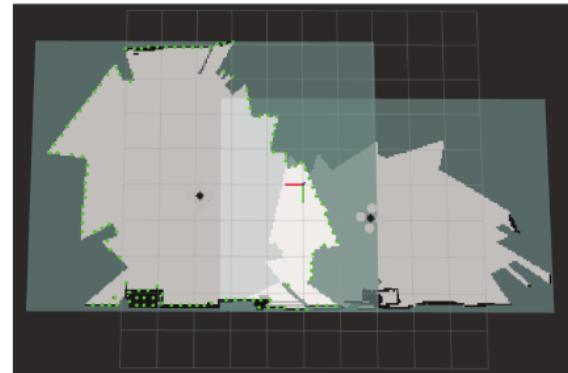
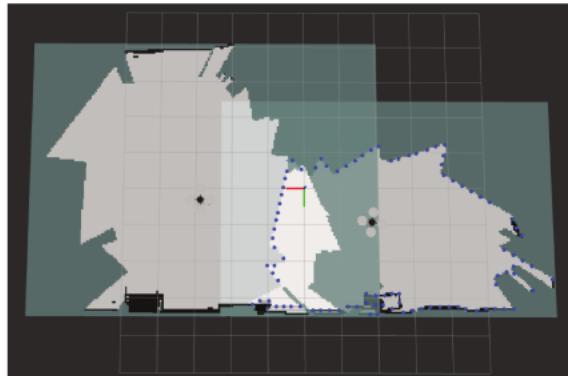
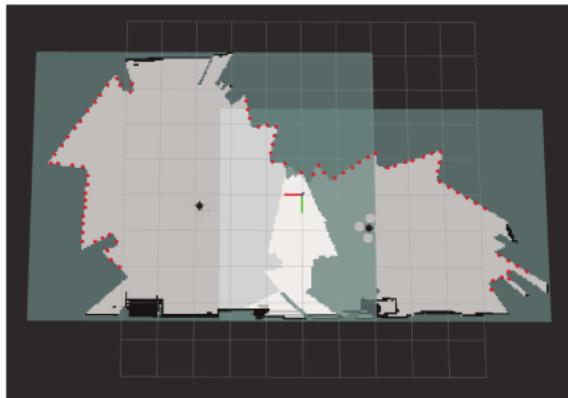
First approach: Convex shape map

In this approach, map shapes are considered convex, which simplifies the intersection computation in Algorithm 3. Figure 1.6 shows two examples of applying frontier processing algorithm while assuming convex shapes with two and three UAVs' map.

This algorithm is relatively easy to apply, but results show that it does not perform well. The convex assumption leads to some false overlapped frontier points. Consequently, some unknown areas will not be visited since their corresponding points are removed and, thus, will not be assigned as a target.

Second approach: Concave shape map

In the second approach, we make the assumption of concave shapes for the UAVs' map to compute their intersection in Algorithm 4. Figure 1.7 shows two examples of applying frontier processing algorithm while assuming convex shapes with two and three UAVs' map.

(a) Frontiers $f_{1,j}$ of UAV₁(b) Frontiers $f_{2,j}$ of UAV₂(c) Candidate targets \mathcal{G}

Hình 1.5: Frontier points processing example: (a) and (b) represent the frontier points of UAV₁ and UAV₂, respectively. (c) represent the candidate targets. The process consists in removing frontier points that belong to overlapped areas, and those that are adjacent to obstacles.

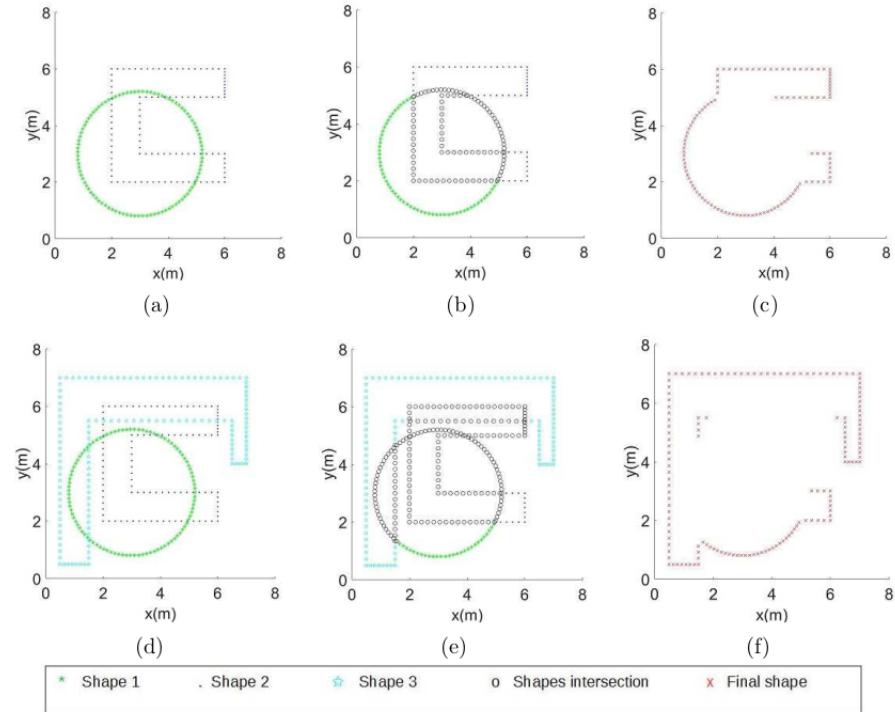
Algorithm 3 Intersection computation algorithm with convex shapes assumption.

Input: $Vect1, Vect2$
Output: $Vect_Final$

```

1: for  $i \in Vect1$  do
2:    $inf = 0, sup = 0$ 
3:   for  $j \in Vect2$  do
4:     if  $Vect1(i).x < Vect2(j).x$  then
5:        $inf ++;$ 
6:     else
7:        $sup ++;$ 
8:     end if
9:   end for
10:  if  $inf > 0$  and  $sup > 0$  then
11:     $Vect\_inter.push\_back(Vect1(i))$ 
12:  end if
13: end for
14: for  $i \in Vect\_inter$  do
15:    $inf = 0, sup = 0;$ 
16:   for  $j \in Vect2$  do
17:     if  $|Vect\_inter(i).x - Vect2(j).x| < 0.5$  then
18:        $inf ++;$ 
19:     else
20:        $sup ++;$ 
21:     end if
22:   end for
23:   if  $inf > 0$  and  $sup > 0$  then
24:      $Vect\_Final.push\_back(Vect\_inter(i))$ 
25:   end if
26: end for

```



Hình 1.6: Frontier points processing while assuming two convex map shapes in the first line and three in the second line. (a) and (d) represent two and three convex map shapes, respectively; (b) and (e) represent the frontier points in overlap (intersection); and (c) and (f) represent the obtained frontier points after processing (final shape).

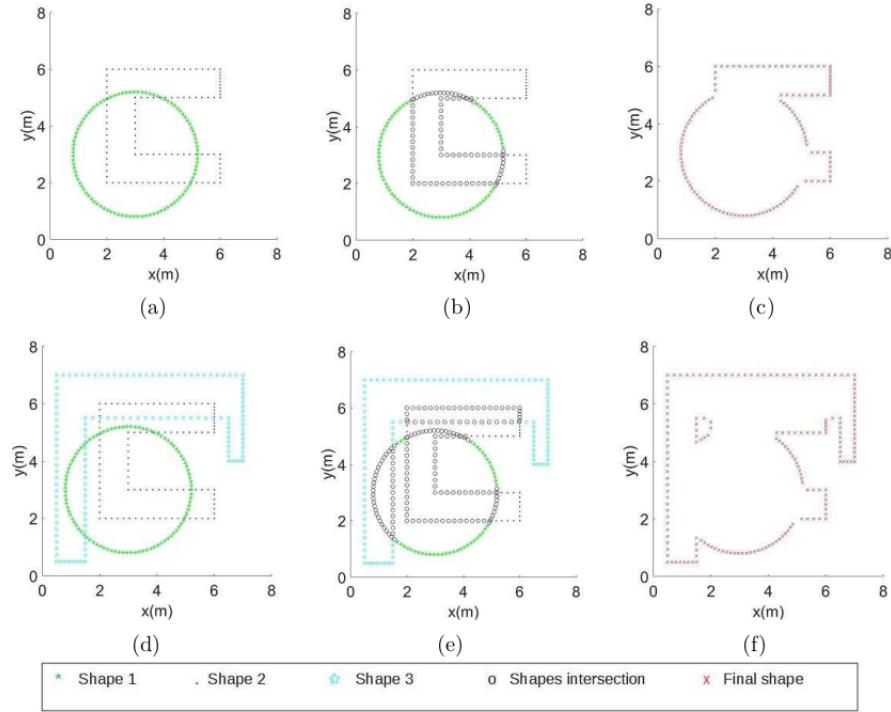
Algorithm 4 Intersection computation algorithm with convex shapes assumption.

Input: $Vect1, Vect2$
Output: $Vect_Final$

```

1: for  $i \in Vect1$  do
2:    $inf = 0, sup = 0$ 
3:   for  $j \in Vect2$  do
4:     if  $Vect1(i).x < Vect2(j).x - 0.5$  then
5:        $inf ++;$ 
6:     else if  $Vect1(i).x > Vect2(j).x + 0.5$  then
7:        $sup ++;$ 
8:     else
9:        $eq ++;$ 
10:    end if
11:   end for
12:   if  $inf > 0$  and  $sup > 0$  or  $eq > 0$  then
13:      $Vect\_inter.push\_back(Vect1(i))$ 
14:   end if
15: end for
16: for  $i \in Vect\_inter$  do
17:    $inf = 0, sup = 0, eq = 0;$ 
18:   for  $j \in Vect2$  do
19:     if  $|Vect\_inter(i).y - Vect2(j).y| < 0.5$  then
20:       if  $Vect\_inter(i).x < Vect2(j).x - 0.5$  then
21:          $inf ++;$ 
22:       else if  $Vect1(i).x > Vect2(j).x + 0.5$  then
23:          $sup ++;$ 
24:       else
25:          $eq ++;$ 
26:       end if
27:     end if
28:   end for
29:   if  $(inf > 0$  and  $sup > 0)$  or  $eq > 0$  then
30:      $Vect\_Final.push\_back(Vect\_inter(i))$ 
31:   end if
32: end for

```



Hình 1.7: Frontier points processing while assuming two concave map shapes in the first line and three in the second line. (a) and (d) represent two and three convex map shapes, respectively; (b) and (e) represent the frontier points in overlap (intersection); and (c) and (f) represent the obtained frontier points after processing (final shape).

Results show that only the frontier points within overlapping areas are removed. Even if some ambiguous frontier points are located within/inside the global shape – which can be confusing –, they are not deleted. This algorithm, assuming concave shapes, performs better than the previous one that assumes convex shapes. Hence, for the frontier processing, the shape of local maps are assumed concave.

Utility function

The proposed utility function in Eq. ?? aims to simultaneously increase the explored area rate and to reduce the distance of each UAV to its corresponding target. The function also considers the average of distances between each robot in the group and this target in order to maximize distances among

robots.

$$\mathbf{U}(UAV_i, t_j) = \mathbf{I}(t_j) \exp(-\lambda \cdot (d_{min}(\mathbf{P}_i, \mathbf{t}_j) + \frac{n_c - 1}{d_{tot}})), \quad (1.2)$$

where UAV i is the considered robot, $\mathbf{t}_j \in \mathcal{G}$ and $\mathbf{I}(\mathbf{t}_j)$ are respectively the candidate target and its corresponding information gain, $\lambda \in [0, 1]$ is a trade-off parameter, n_c is the number of UAVs in the cluster \mathcal{C} , and $d_{tot} = \sum_{k=1, k \neq i}^{n_c} (d_{min}(\mathbf{P}_k, \mathbf{t}_j))$ is the sum of the minimum distance from UAV k 's pose \mathbf{P}_k to the candidate target j . The proposed utility function is inspired from [25] and it has been presented in our works [32],[33]. This function performs a trade-off between rapid exploration and a precise filling the map using a tuning parameter λ . From Figure 1.8, we can notice that the larger λ , the less important the distance d_{tot} . Thus, a precise filling is favored over rapid exploration and vice versa.

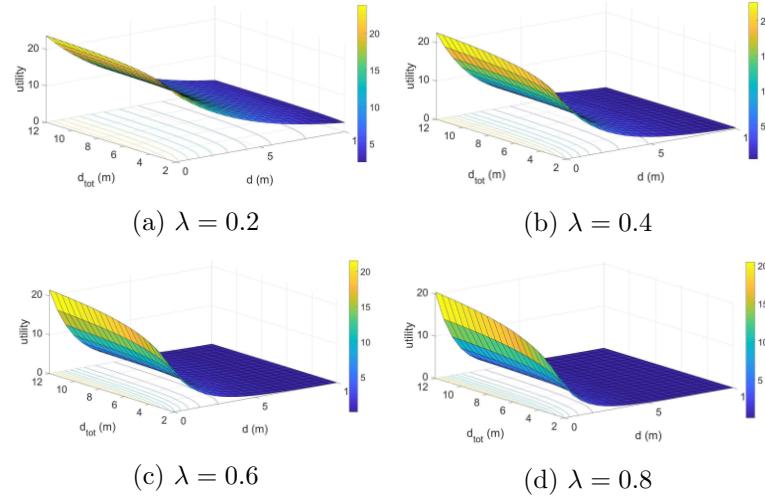
Regarding the Multi-UAV case, the utility function is based on the average of neighbors distances. As shown in Figure 1.8, with an information gain of $\mathbf{I}(\mathbf{t}_j) = 25$ and three UAVs in the cluster ($n_c = 3$); an increasing distance of UAV to the target will reduce the utility function. Whereas, the larger the average distance from other UAVs w.r.t. to the target, the more the utility. So the function tends to choose the closest target to the considered UAV; but at the same time, the farthest one from the others.

In the case of a single UAV, the utility function (See Eq. 1.3) tends to choose the closest target with the maximum of information gain:

$$\mathbf{U}(UAV_i, \mathbf{t}_j) = \mathbf{I}(\mathbf{t}_j) \exp(-\lambda \cdot (d_{min}(\mathbf{P}_j, \mathbf{t}_j))), \quad (1.3)$$

The parameter $d_{tot} = \sum_{k=1, k \neq i}^{n_c} (d_{min}(\mathbf{P}_k, \mathbf{t}_j))$ represents the sum of the minimum distance between the target \mathbf{t}_j and the neighbors' poses \mathbf{P}_k with $k \in [1..n_c] \setminus i$. So, if \mathbf{t}_j has neighboring UAVs that are too far, the utility function will increase, so \mathbf{t}_j is more likely to be chosen.

The aim in the utility function is to maximize d_{tot} . We distinguish two cases where d_{tot} can be too close to zero or equal to it:



Hình 1.8: Utility function behavior: $\mathbf{I}(t_j) = 25, n_c = 3, d_{tot} = \sum_{k=1, k \neq i}^{n_c} (d_{min}(\mathbf{P}_k, \mathbf{t}_j))$. The average distance of other UAVs d_{tot} has a minimum value different from zero since n_c is different from zero too.

- There are no neighbors. In this case, there is one UAV in the fleet: $n_c = 1$. The utility function in Eq. ?? is used.
- The candidate target \mathbf{t}_j and the neighbors' poses \mathbf{P}_k with $k \in [1..n_c] \setminus i$ are almost confused (too close to each others). In this case \mathbf{t}_j is less likely to be chosen.

Yet, this parameter d_{tot} could have a high value when:

- There is one UAV too far from \mathbf{t}_j .
- There are several UAVs with relatively short distance from \mathbf{t}_j .

So, in order to avoid these ambiguous situations, the average of d_{tot} is considered by using $\frac{n_c}{d_{tot}}$.

1.3.5 Goal assignment process

In order to make appropriate UAV-to-target assignment, the utility of reaching each candidate frontier is considered. The goal assignment process is described in Algo. 5.

For each UAV_i , the utilities of reaching all the candidate targets are computed. The target \mathbf{t}_g that maximizes the utility is computed and, the assignment $\theta(\text{UAV}_i, \mathbf{t}_g)$ is performed. Then, the remaining candidate targets $\mathcal{G} \setminus \mathcal{T}$ are scheduled in order to avoid to select a target too close to \mathbf{t}_g , in the next iteration. Also, \mathbf{t}_g is removed from \mathcal{G} to prevent from assigning the same target to different robots. This assignment process is performed for the available UAVs in \mathcal{C} in a sequential manner until getting all assigned targets $\mathbf{t}_g \in \mathcal{T}$ with $g \in [1..n_t]$ (See Figure 1.2)

Algorithm 5 Goal assignment algorithm.

Input: Candidate targets $\mathbf{t}_k \in \mathcal{G}, k \in [1..n_g]$ and their respective information gain $\mathbf{U}(\mathbf{t}_k)$, poses \mathbf{P}_i of all robots in the considered cluster \mathcal{C} .

Output: $\theta(\text{UAV}_i, \mathbf{t}_g)$ assignment of UAV_i with target g .

- 1: $\mathcal{T} = \emptyset$
 - 2: **while** no goal for UAV_i **do**
 - 3: Compute its corresponding utility of reaching each remaining candidate goal $\mathbf{U}(\text{UAV}_i, \mathbf{t}_k)$ with $\mathbf{t}_k \in \mathcal{G} \setminus \mathcal{T}$.
 - 4: $t_g = \text{argmax}_{\mathbf{t}_k \in \mathcal{G} \setminus \mathcal{T}} (\mathbf{U}(\text{UAV}_i, \mathbf{t}_k))$.
 - 5: Schedule the information gain of the remaining candidates $\mathbf{t}_k \in \mathcal{G} \setminus \mathcal{T}$.
 - 6: $\mathcal{T} = \mathcal{T} \cup t_g$.
 - 7: **end while**
 - 8: **return** $\theta(\text{UAV}_i, \mathbf{t}_g)$ assignment.
-

Stop condition

The goal selection process is realized by each cluster/group-*leader* (if $n > n - c$) or the Fleet-*leader* (if $n = n_c$). This assignment aims to, cooperatively, distribute the robots in the environment to explore simultaneously different unknown regions. As long as candidate frontier points are still available, the *leader* continues to assign targets to *explorers* and they attempt to reach their assigned goals. When the *leader* notices that no candidate targets are left, that means that all the environment has been explored successfully and the mission is accomplished. Thus, it has to send back to the *explorers* an acknowledgment to prevent them assuming a communication loss.

Loop rate

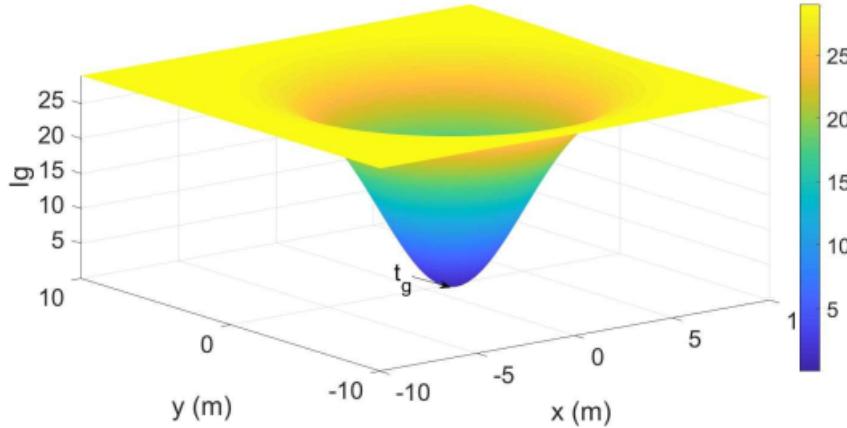
The target assignment process is performed at each loop. The frequency of assigning targets impacts the duration and the efficiency of the mission. In a distributed approach, as soon as the UAV reaches its current target, it selects a new one without consulting the others. In a centralized approach, the first UAV to reach its current target has to wait until the others reach their respective targets. This can be a problem as soon as one of them fails or leaves the mission. Another possibility is to begin to assign targets once one UAV reaches its target. But this may generate incomplete tasks. In the proposed strategy, the frequency of assignment or loop rate r is predefined depending on the average of time to reach a target (See Eq. 1.4).

$$r \in \left[\frac{s}{v_{i,max}}, \frac{s}{v_{i,min}} \right] \quad (1.4)$$

where s is the maximum sensor range and v_i is the UAV's velocity.

Scheduling the information gain

The information gain of each remaining candidate target $\mathbf{I}(\mathbf{t}_k)_{t-1}$ at time $t - 1$ with $\mathbf{t}_k \in \mathcal{G} \setminus \{t_g\}$, that belongs to the threshold range $[r_{min}, r_{max}]$, is scheduled at time t depending on its distance w.r.t. the target t_g , using Eq. 1.5. Figure 1.9 represents an example of the function shape used to schedule the information gain. The example shows a Gaussian function with an amplitude that corresponds to the information gain maximum value; a center that corresponds to the target position $(\mathbf{t}_g(x), \mathbf{t}_g(y))$; and σ_x and σ_y that spread the blob in x and y axis, respectively.



Hinh 1.9: Example of the shape function to schedule the information gain($\mathbf{I}(\mathbf{t}_k)_{t-1} = 29, (\mathbf{t}_g(x), \mathbf{t}_g(y)) = (1.25, -0.275), \sigma_x = \sigma_y = 3$).

$$\mathbf{I}(\mathbf{t}_k)_t = \mathbf{I}(\mathbf{t}_k)_{t-1} \left(1 - \exp\left(-\left(\frac{(\mathbf{t}_k(x) - \mathbf{t}_g(x))^2}{2.\sigma_x^2} + \frac{(\mathbf{t}_k(y) - \mathbf{t}_g(y))^2}{2.\sigma_y^2}\right)\right)\right), \quad (1.5)$$

where $\mathbf{t}_k(x)$ and $\mathbf{t}_k(y)$ are the remaining candidate target coordinates; $\mathbf{t}_g(x)$ and $\mathbf{t}_g(y)$ are the target coordinates; and σ_x and σ_y are the spreads of the blob. The smaller the distance of the frontier point \mathbf{t}_k w.r.t. the target \mathbf{t}_g , the smaller the information gain. When reducing the information gain, the candidate targets are less likely to be chosen and thus, robots ensure a certain distance among their future targets.

1.3.6 Path planning and control

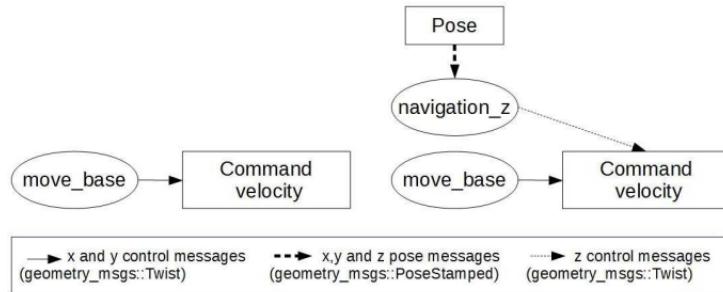
As explained in Section 5.2 of Chapter 1, UAVs are assumed to navigate in a simplified 2D environment with a fixed z value. Block 6 in Figure 1.6 is responsible for planning a path to the selected target and attempting to reach it. These tasks are ensured by the *move base*¹ package.

For the navigation task, each UAV maintains a local and a global planner along with a local and a global costmap, respectively. The costmap is a 2D cell grid \mathcal{L} with additional inflation that consists in propagating cost values out from occupied cells and decreasing them with distance. The global

¹Source: http://wiki.ros.org/move_base

costmap has the size of the UAV's map whereas, the local costmap has a fixed size moving window. Given a starting point – the current pose – and an endpoint – the assigned target – in the global costmap, the global planner produces a plan using a navigation function computed with Dijkstra algorithm [17]. It consists in following the adjacent free cells until reaching the goal. Then taking into account the local costmap, the local planner generates velocity commands for the UAV's mobile base. A recovery rotational behavior is also performed when needed in order to clear the robot's field of view.

The target assigned by the *leader* is ensured to belong to an unknown area using the exploration strategy. The trajectory planning process is performed locally on each robot. And since the UAVs do not exchange their local maps nor fuse them, they are likely to revisit already explored areas while following the planned path. To minimize these overlapped regions during navigation, a priority is given to frontier points $\mathbf{f}_{i,j}$ to be a target for UAV_i over UAV_k with $k \neq i$. This helps the UAV to maintain the same direction during exploration. The *move base* package is a 2D navigation stack. However, to avoid drifting on the z axis, a control command is added to keep a static z altitude (See Figure 1.10).



Hình 1.10: From 2D (left) to 3D (right) navigation stack. The *navigation z* is a ROS process developed to provide a control command on the z axis.

1.4 Result and discussions

Simulations have been performed to evaluate the proposed exploration strategy. Additional tests while using relative localization have been done to measure the system performances. The simulations are performed using Robot Operating System (ROS) running on a 2.60GHz i7 Linux machine. For the

quad-rotor simulation, the AR-drone model² equipped with an RGB-D camera in a forward-looking configuration, is used. A bounded unknown environment is generated using *Gazebo* simulator. The number of robots used for evaluation is limited to three, however, the proposed system architecture is not constrained to a fixed number of robots.

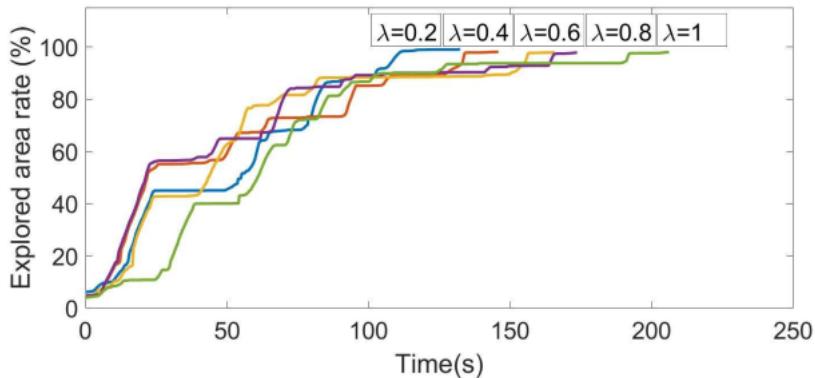
1.4.1 Parameters tuning

For an effective evaluation of the exploration strategy, we run some tests to set the most adequate parameters configuration.

Trade-off parameter λ

The utility function (See Eq. ??) used in the exploration strategy can be tuned, using a trade off parameter λ , between fast exploration and filling in details the map.

Figure 1.11 shows different runs while varying this parameter. By increasing λ , the information gained when reaching the goal is favored over the distance and thus, the cost to it, and vice versa. So, when λ is small, the traveled distance is small and so the exploration time. Though, some times during the mission, high values of λ are noticed to reach higher exploration rate than smaller ones.

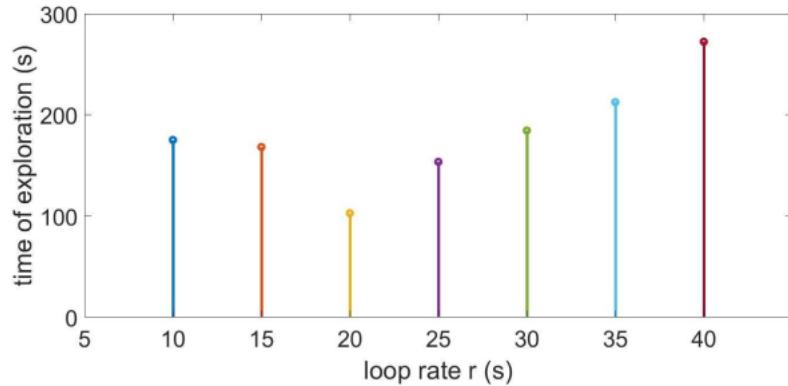


Hinh 1.11: The impact of varying the trade off parameter λ over exploration time.

²Source: http://wiki.ros.org/ardrone_autonomy

Loop rate r

The frequency or loop rate r of target assignment may also affect exploration time performance. The values of r vary to take into account the robot velocity \mathbf{v}_i and the sensor's maximum range s . The impact of varying the loop rate is evaluated in Figure 1.12.



Hình 1.12: Exploration time while varying the loop rate r .

Given a robot velocity $\mathbf{v}_i = \{0.1, 0.3\} m.s^{-1}$ and a maximum sensor range $s = 4m$, the loop rate varies in $r \in [10, 40]$. This parameter should not be too small to allow the robot to reach its target; nor too big to prevent long waiting times for the next goal assignment.

Common parameters

Depending on results in Figure 1.11 and Figure 1.12, respectively, λ is set to 0.2 and r to $20s$ in order to maximize the explored area rate while minimizing the mission time. The simulation parameters are summarized in Table 1.1.

Bảng 1.1: Common parameters.

Parameter	Value
RGB-D horizontal FoV	$\pi/3$
Trade-off parameter λ	0.2
RGB-D maximum range $s(m)$	4
Min distance among frontiers $d(m)$	0.3
Occupancy grid resolution (m)	0.05
Range to schedule the $lg[\sigma_x, \sigma_y](m)$	[3, 3]
Loop rate $l(s)$	20
Environment dimension (m^2)	8×8
Linear velocity $v_i(m.s^{-1})$	[0.1, 0.3]
Angular velocity $w_i(rad.s^{-1})$	[0.1, 0.3]

1.4.2 Exploration strategy performances

The proposed exploration strategy has been evaluated in terms of distribution of the robots in the environment, overlap rate, exploration time, and total traveled distance by each robot.

Maps evolution during the mission

While reaching their respective assigned goals, each robot is in charge of creating a detailed grid map of the visited area in order to get a global map of the environment. Figure 1.13 illustrates the growth of the reconstructed 3D occupancy grid map at different times during the mission. Note that to reach 99% of coverage, a lot of time is spent.

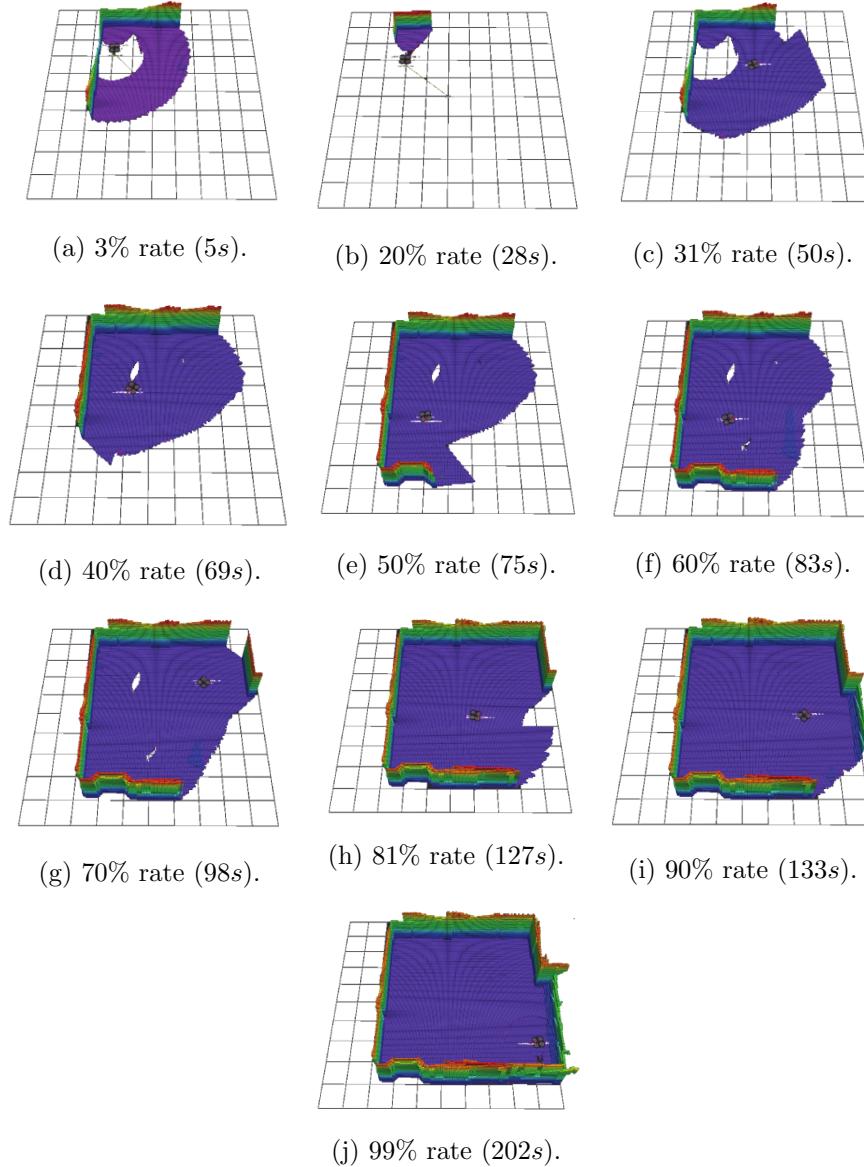
Figure 1.14 shows the evolution of the respective projected 2D local grid map of two robots during a cooperative exploration mission. The global projected 2D grid map is also created and represented for evaluation (the occupancy grid matching process introduced in Algorithm 2.1 in Chapter 2 is used). The robots' initial positions are $(1, 0, 0)$ for UAV_1 and $(1, -3, 0)$ for UAV_2 . Despite a relatively close initial position, the proposed strategy effectively spreads the robots so that UAV_1 is in charge of the left side of the environment and UAV_2 of the right one.

Frontier points evolution during the mission

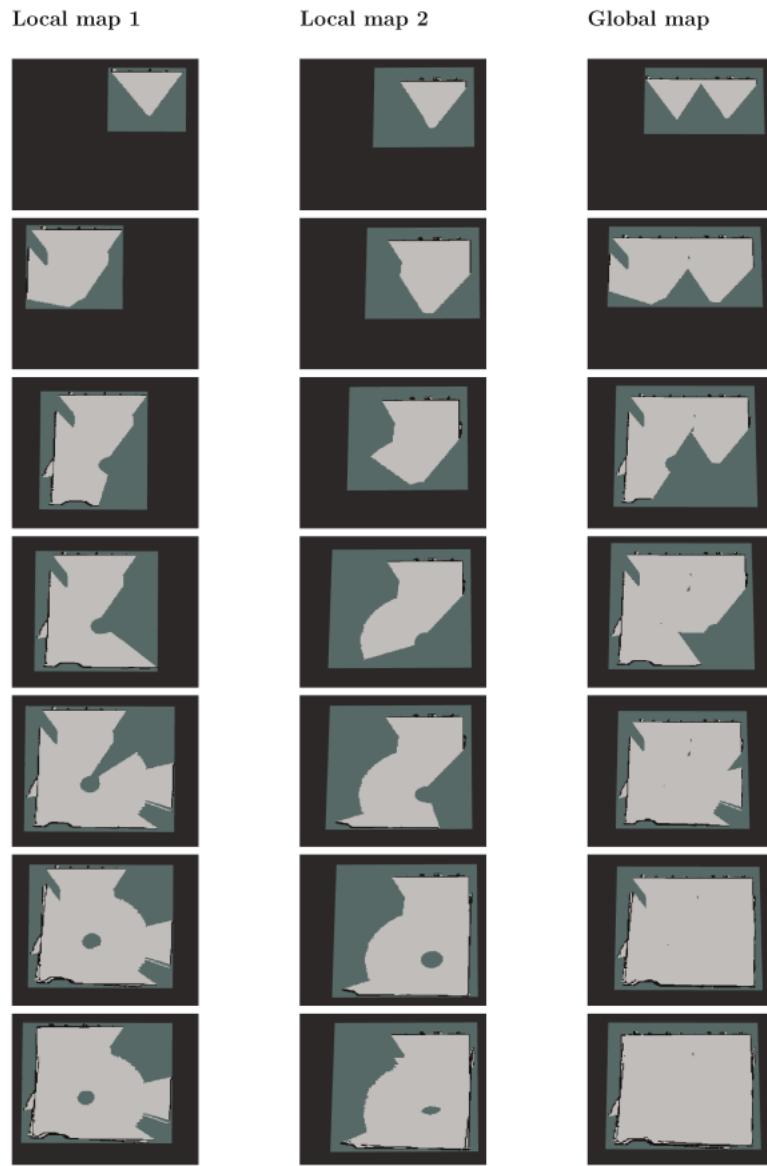
The target is chosen from the candidate frontier points that define the edges of an environment not previously explored. These candidates are selected from the final frontier points of each UAV in the fleet (See Figure 1.15). During the exploration mission, the local map size increases, which leads to an increasing number of local frontier points. At the beginning of the exploration, the number of candidate frontier points increases, but as soon as the exploration evolves in time, their number decreases. At the end of the mission, when all the environment is explored, no candidate frontier points should be left.

UAV's trajectories during the mission

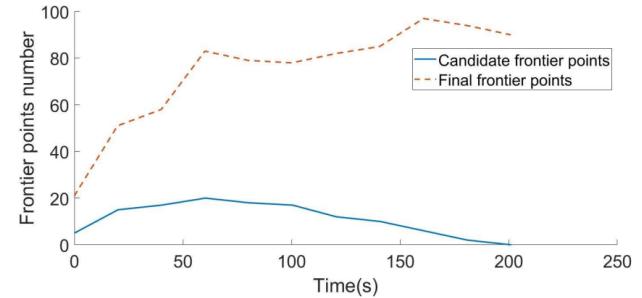
Figure 1.16 shows the explored map with the trajectories using one, two and three UAVs. The UAVs try to explore the full environment while avoiding already explored areas. In a cooperative way, each UAV is in charge of visiting an area by reaching a target that belongs to a non-explored environment. These targets are assigned by the *Leader* which, even if the initial poses of the UAVs are relatively close, effectively spreads them into unknown areas. The global map is composed of the superposition of all UAVs' local maps.



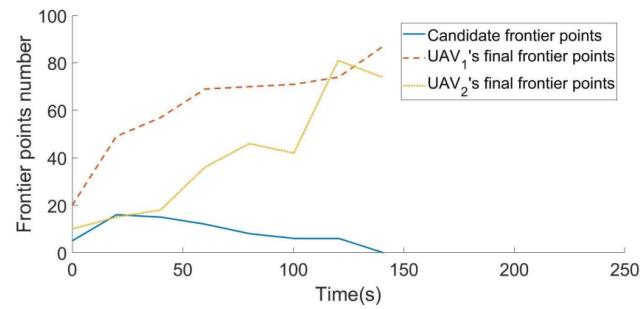
Hình 1.13: Ratio of explored space during time for one UAV.



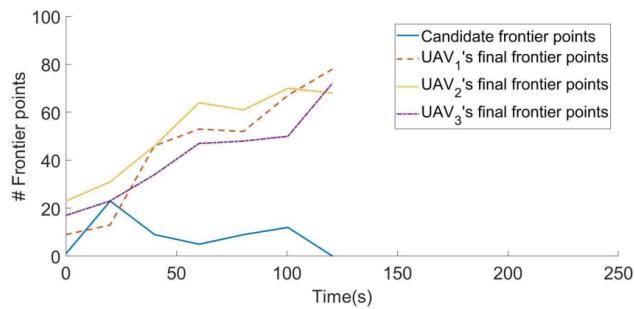
Hình 1.14: Coordinated exploration using two robots. Columns 1, 2 and 3 show the evolution of the local map of the UAV₁, the UAV₂ and the global map over time, respectively.



(a) One UAV.



(b) Two cooperative UAVs.

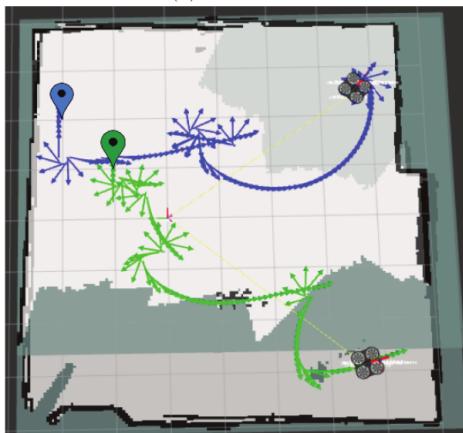


(c) Three cooperative UAVs

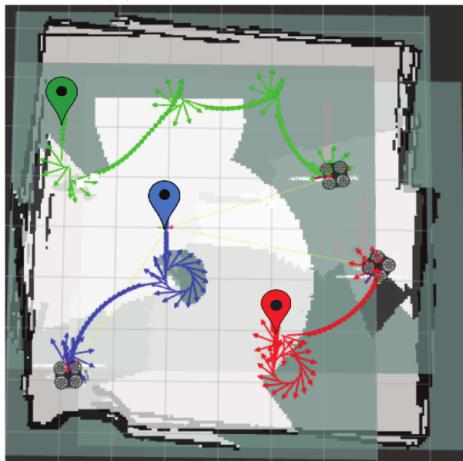
Hình 1.15: The evolution of candidate and final frontier points numbers during cooperative exploration.



(a) One UAV.



(b) Two cooperative UAVs.



(c) Three cooperative UAVs

Hình 1.16: Projected 2D map in a coordinated exploration with a team of one, two and three UAVs. Green, blue and red markers and arrows define, respectively, the initial position and the trajectory of UAV1, UAV2 and UAV3.

Goal assignment evaluation: Distribution of the robots in the environment

The goal assignment process is performed according to the algorithm described in Section 3.6. Nevertheless, after assigning a target to the first robot in the list, the same target or another one close to it may be assigned to the second robot in the list. To overcome these issues, the information gains of the remaining candidate targets are scheduled. This allows to discard an already assigned target and keep a certain distance between the new target and the previous one assigned.

Suppose that a target is assigned to the first robot in the cluster list. Figure 1.17 shows the goal selected for the second robot when a sequential assignment is performed:

- Without further frontier points processing (See Figure 1.17b). Consequently, the same target is assigned to two different robots.
- While removing the assigned target from the remaining candidate frontier points (See Figure 1.17c). Consequently, the second target is relatively close to the first one assigned.
- While scheduling the information gain after each target assignment (See Figure 1.17d). Consequently, the assigned targets are spaced out into the environment. The information gain is scheduled following Eq. 1.5. The information gain value increases with distance to the candidate target \mathbf{t}_g .

The goal assignment process may sometimes be not optimal since it depends on the robots' order in the list. For example, suppose that robots UAV_i and UAV_j have the same best target assignment \mathbf{t}_k such that it offers the maximum utility over candidate frontier points where Eq.1.6 and Eq. 1.7 are verified.

$$\mathbf{t}_k = \operatorname{argmax}_{\mathbf{t}_m} \mathbf{U}(\text{UAV}_i, \mathbf{t}_m) \quad (1.6)$$

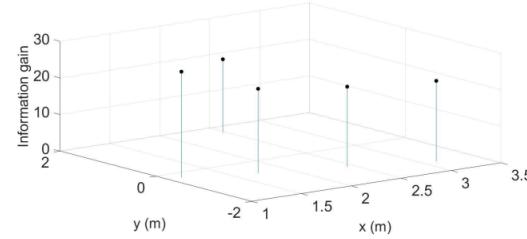
with $\mathbf{t}_m \in \mathcal{G}$.

$$\mathbf{t}_k = \operatorname{argmax}_{\mathbf{t}_n} \mathbf{U}(\text{UAV}_j, \mathbf{t}_n) \quad (1.7)$$

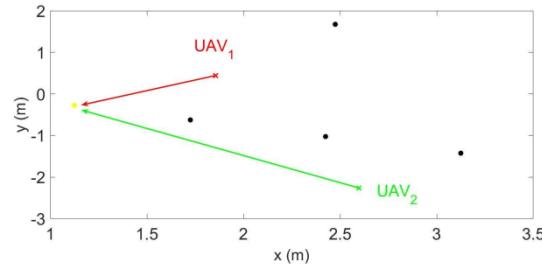
with $\mathbf{t}_n \in \mathcal{G}$.

UAV_i have another candidate frontier point \mathbf{t}_l with:

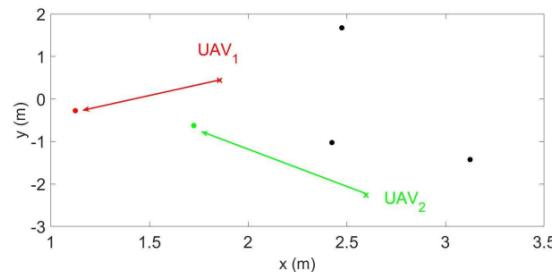
$$U(\text{UAV}_i, \mathbf{t}_k) > U(\text{UAV}_i, \mathbf{t}_l) > U(\text{UAV}_j, \mathbf{t}_k) \quad (1.8)$$



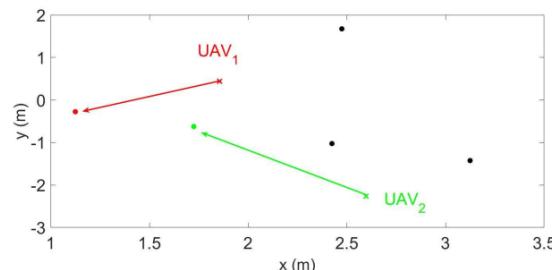
(a) Candidate targets.



(b) Case 1.



(c) Case 2.



(d) Case 3.

Hình 1.17: Goal assignment: After assigning a target to UAV_1 , a target is assigned in a sequential manner to UAV_2 . (a) represents the candidate frontier points with their respective information gain. (b), (c), and (d) represent, respectively, the targets assignment when: No further process is performed for the remaining candidate targets; UAV_1 's target is removed from the remaining candidate targets and; the information gain of the remained candidate targets are scheduled.

So the optimal solution would be to assign \mathbf{t}_l to UAV_{*i*} and \mathbf{t}_k to UAV_{*j*}. But, if UAV_{*i*} is the first in the list, \mathbf{t}_k is assigned to it and another candidate frontier point with less utility than \mathbf{t}_k , is assigned to UAV_{*j*}. Thus, the solution with sequential goal assignment is not always optimal.

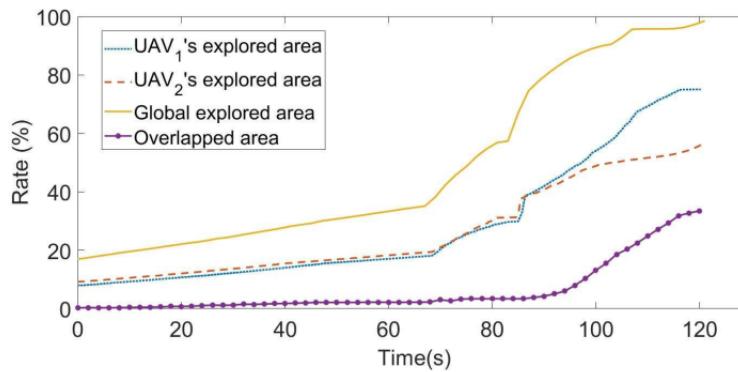
To overcome this problem, all the numbers of possible combination $\frac{n_g!}{n_g!(n_g-n_c)!}$ with n_g the number of candidate targets and n_c the number of robots, need to be considered. This increases considerably the computation time by increasing the number of robots. Therefore, in the proposed algorithm, sequential assignment is favored over computing all possible permutations.

Explored space rate evaluation

This evaluation aims at quantifying the amount of explored spaces during the mission. Figure 1.19 shows the explored space rate performed by each UAV in the fleet. The more UAVs in the environment, the less the exploration rate demanded by each one. A robot has no need to continue exploring an area if it has already been explored by another one. Thus, the mission time is considerably reduced.

Overlap rate evaluation

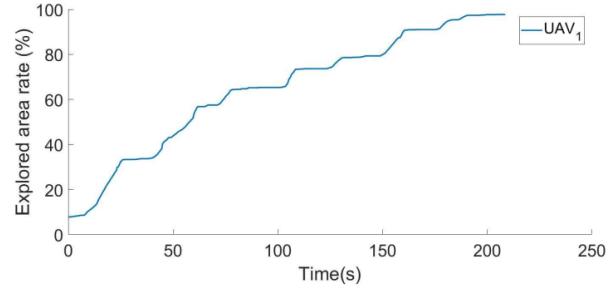
The use of an effective goal assignment process should limit the generated overlap. In Figure 1.18, the time evolution of overlap is evaluated using two cooperative robots. The overlap undergoes a significant increase at the end of the exploration to reach 33%. This is explained by the closeness of the local maps at the end of the mission to precisely fill the global grid map.



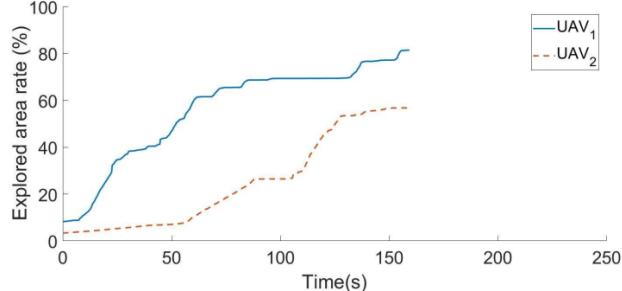
Hinh 1.18: Explored and overlapped area rate using two cooperative UAVs.

Traveled distance evaluation

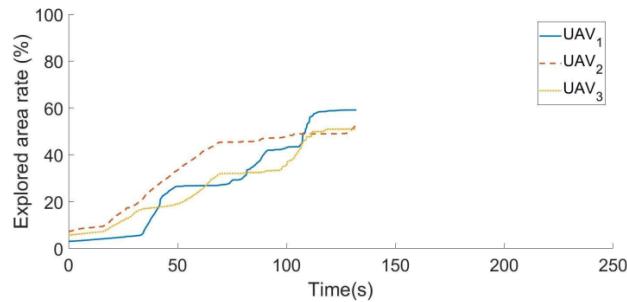
To effectively evaluate the exploration strategy performance in terms of distance traveled by each UAV, different runs with one, two and three UAVs have been conducted where explored area rate reaches almost 99%. Figure 1.20 shows the distance traveled by each UAV in the fleet during the mission.



(a) One UAV.

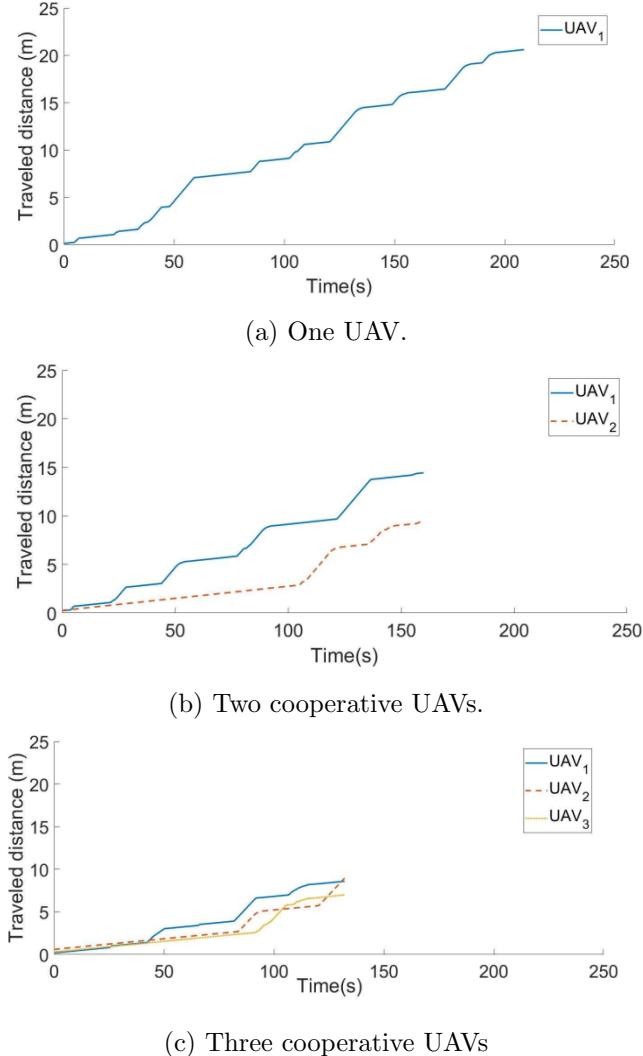


(b) Two cooperative UAVs.



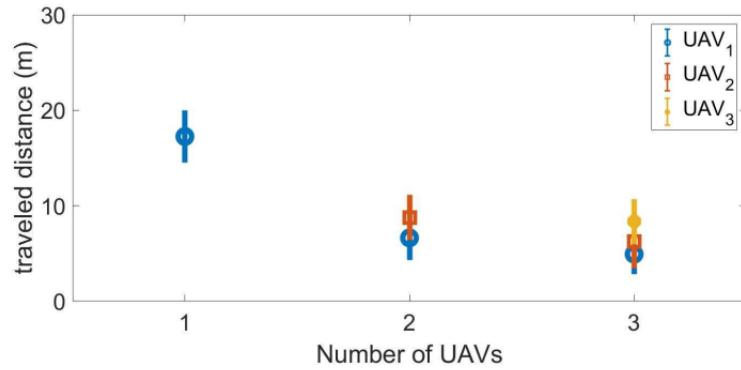
(c) Three cooperative UAVs

Hình 1.19: Explored space rate with one, two and three UAVs.



Hinh 1.20: Traveled distance by each UAV with one, two and three cooperative UAVs.

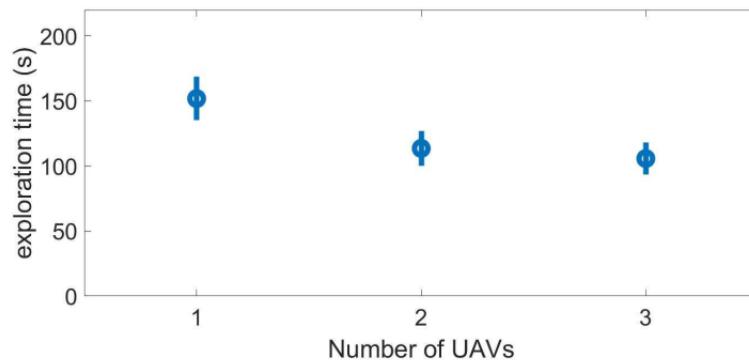
The distances traveled by each UAV using one, two and three UAVs in the fleet are compared in Figure 1.21. This distance decreases with the number of UAVs. The average distance traveled by each UAV is reduced by 55% for 2 UAVs and by 62% for 3 UAVs. The error of the traveled distance is slightly reduced from one to two and three UAVs.



Hình 1.21: Traveled distance evaluation.

Exploration time evaluation

For the exploration time evaluation, different runs have been performed using 1, 2 and 3 UAVs. Figure 1.22 shows that the average of exploration time decreases when the number of robots in the fleet increases. The computed error decreases as well. The time is reduced by 25% for 2 UAVs and by 30% for 3 UAVs. The exploration time and distance are not divided by 2 or 3 when multiplying by 2 or 3 the number of robots, respectively. During these simulations, the robots' initial positions are: (1, 0, 0) for one UAV; (1, 0, 0) and (1, -3, 0) for two UAVs; and (1, 1, 0), (1, -1, 0) and (1, -3, 0) for three UAVs.



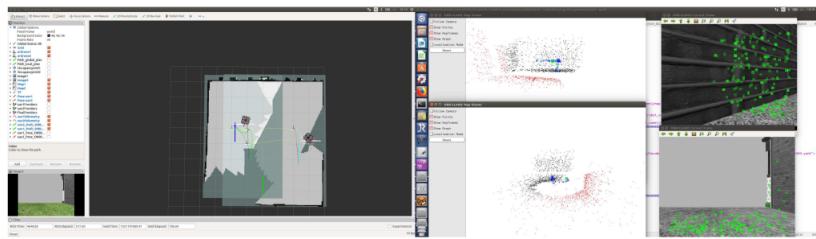
Hình 1.22: Average exploration time.

The results presented in Section 4.2 were evaluated without a relative

localization. So, for a more challenging realistic scenario, runs with relative localization algorithm have been performed to evaluate system performances using SLAM.

1.4.3 Exploration mission using relative localization algorithm

Toward a more realistic scenario, the ORB-SLAM2 approach (introduced in Section 5 of Chapter 2) has been implemented to perform relative localization (See Figure 1.23).



Hình 1.23: Two UAVs navigation using ORB-SLAM2. Rviz image (left) shows, for each UAV, the constructed 2D occupancy grid map, its estimated trajectory and the corresponding ground truth. The point cloud (middle) represent the sparse reconstruction of the environment made by each UAV. And, the green markers (right) represent the features computed by each UAV to perform localization.

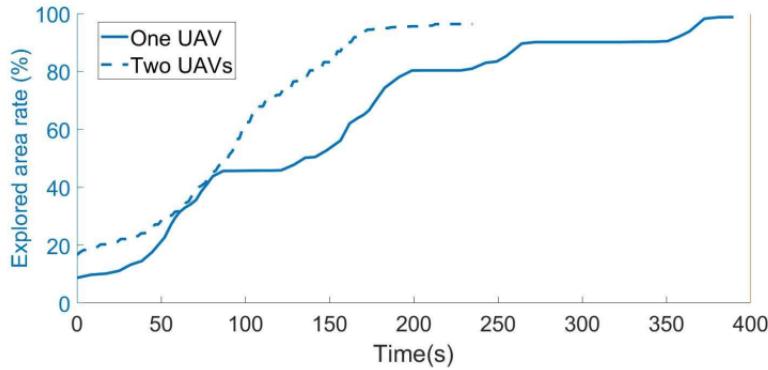
Each robot performs SLAM where it constructs its map in its local reference frame ${}^W F_i$, and estimates its relative pose \mathbf{p}_i within it. Then, the fleet performs cooperative exploration using some specific information exchanged among UAVs. But, these information have to be in a common reference frame. Therefore, information such as the pose \mathbf{p}_i and the frontier points $\mathbf{f}_{i,j}$ are necessarily transformed into the global reference frame ${}^0 W$ before being exchanged. Hence, the *leader* makes all the needed computation and sends back to the *explorers* the targets in ${}^0 W$. When a robot receives its assigned goal, it transforms it into ${}^W F_i$ to plan a path to it.

To perform a transformation from local reference ${}^W F_i$ to global one ${}^0 W$, the robot has to know – at least – its initial pose w.r.t. ${}^0 W$. As explained in Section 5.2 of Chapter 1, the global reference frame of the environment is initialized such that it coincides with the local reference frame of the

first group-*leader* in the fleet which is UAV₁ in the considered example of Equation 1.9.

$${}^0W \equiv {}^WF_1, \quad (1.9)$$

Then, by detecting this robot using tags mounted on it, the other robots are able to estimate their respective transform to it ${}^{F_j}[\mathbf{R} \ t]_{F_1}, j \in [2..n_c]$. For simulation evaluations, the information of transform – computed while detecting the tag – are assumed to be known. Figure 1.24 shows the exploration rate evolution during the exploration mission while using ORB-SLAM2 as the relative localization approach. The mission time using ORB-SLAM2 is reduced by 43% for 2 UAVs instead of 1 UAV.



Hình 1.24: Explored area rate evolution during exploration mission with one and two UAVs while performing ORB-SLAM2 by each UAV.

The exploration time when using a relative localization (See Figure 1.24) is relatively important compared to the exploration without SLAM (See Figure 1.19) since the velocity has been considerably reduced.

1.5 Conclusion

In this chapter, we presented a state of the art of multi-robot exploration mission including the strategy used to assign a robot to a target and an utility function adopted to estimate the interest of reaching it. Then, we introduced an exploration strategy based on the group-*leader* decision making. The robot-to-target assignment is performed using a novel utility function. This function makes a trade-off between fast exploration and getting a detailed grid map, and also takes into account the distance of each robot in the

group from the unexplored set of targets. Also, we propose to schedule the information gain in order to efficiently spread the UAVs into the environment. Moreover, the strategy adopted exchanges the frontier points instead of a whole copy of the local map.

Results show that the proposed cooperative exploration strategy minimizes the global exploration time by 25% for 2 UAVs and by 30% for 3 UAVs, while minimizing the average traveled distance by each UAV by 55% for 2 UAVs and by 62% for 3 UAVs. Furthermore, the strategy was evaluated using a relative localization algorithm where the exploration time was reduced by 43% for 2 UAVs instead of 1 UAV.

Chương 2

Giao tiếp giữa các robot

2.1 Giới thiệu

Một chủ đề quan trọng trong các hệ thống nhiều robot là giao tiếp giữa các robot. Tính năng này được sử dụng để điều phối và chia sẻ thông tin cụ thể. Một đội có thể được triển khai trong một số nhiệm vụ trong các khu vực tương đối khác biệt và thù địch. Tuy nhiên, những môi trường này có thể không có cơ sở hạ tầng mạng sẵn có, ngoài các liên kết truyền thông không lý tưởng.

Chương này nhấn mạnh một trong những điểm thách thức nhất trong MRS, đó là giao tiếp giữa các robot. Vấn đề này có thể được giải quyết từ các quan điểm khác nhau; nhưng, chúng tôi đã lựa chọn nghiên cứu hai vấn đề nhỏ, đó là: định dạng mạng, cấu trúc liên kết mạng và chiến lược cho sự mạnh mẽ của MRS.

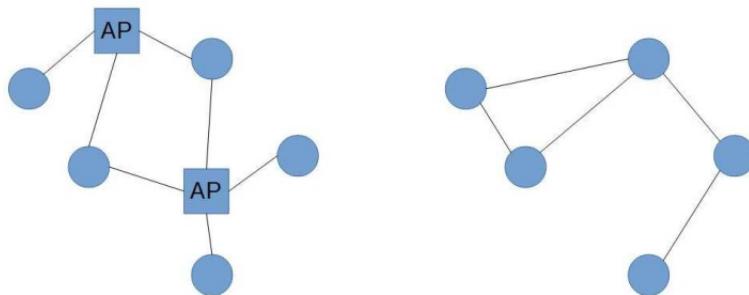
2.2 Phân loại mạng

Mạng được sử dụng để liên kết giữa các thực thể khác nhau và để thiết lập giao tiếp có thể có giữa chúng. Một số loại mạng tồn tại và có thể được phân loại theo nhiều cách khác nhau.

2.2.1 Chế độ cơ sở hạ tầng so với chế độ không có cơ sở hạ tầng

Để quản lý cơ sở hạ tầng mạng, các chế độ hiện có có thể được phân loại thành hai loại chính [4],[24]. Loại đầu tiên là chế độ cơ sở hạ tầng, có thể

được gọi là chế độ Điểm truy cập (AP) và loại thứ hai là chế độ không có cơ sở hạ tầng, gọi là Ad Hoc mode (Xem Hình 2.1). Trong chế độ AP, giao tiếp UAV-to-UAV được thực hiện thông qua cơ sở hạ tầng (AP, bộ định tuyến, v.v.), không giống như chế độ Ad Hoc trong đó các nút giao tiếp trực tiếp với nhau. Chế độ Ad Hoc còn được gọi là chế độ ngang hàng. Mạng Ad Hoc có thể phát triển thành một chế độ ngang hàng khác được gọi là mạng lưới bằng cách cho phép khả năng đa bước. Thật vậy, mạng Ad-Hoc không có bất kỳ khả năng cố hữu nào cho đa bước. Các nút giao tiếp với nhau khi chúng nằm trong phạm vi giao tiếp của nhau. Trong khi đó, trong mạng lưới, các nút có thể giao tiếp trực tiếp hoặc thông qua một hoặc nhiều nút trung gian.



Hình 2.1: Chế độ cơ sở hạ tầng (trái) và chế độ không có cơ sở hạ tầng (phải).

Bảng 2.1: Chế độ cơ sở hạ tầng so với chế độ không có cơ sở hạ tầng.

	Chế độ cơ sở hạ tầng	Chế độ không có cơ sở hạ tầng
Ưu điểm	Giao tiếp đáng tin cậy	Kết nối trực tiếp với nhau, dễ thiết lập, mạnh mẽ với lỗi nút, cho phép mở rộng và điều chỉnh trong cấu trúc liên kết mạng.
Nhược điểm	Phản ứng đắt, phức tạp, phạm vi bị hạn chế.	Thừa trong kết nối mạng, khó bảo trì.
Tiêu chuẩn	802.11	802.11, 802.15, 802.16

Ban đầu, robot phải thực hiện nhiệm vụ của chúng trong môi trường không có cơ sở hạ tầng mạng. Trên thực tế, chúng trao đổi thông tin khi

chúng hoạt động như một bộ định tuyến và như một AP. Vì vậy, hệ thống MRS có thể được xem như một mạng Ad Hoc di động theo quan điểm giao tiếp.

2.2.2 Phân loại mạng Ad Hoc

Mạng Ad Hoc là mạng được sử dụng phổ biến để quản lý giao tiếp đa robot. Với mạng này, mỗi robot có thể di chuyển tự do và chuyển tiếp các gói tin đến và từ mỗi robot khác tùy thuộc vào phương thức phân phối dữ liệu [6]. Mạng này được đặc trưng bởi chất lượng dịch vụ phức tạp, trong đó giao thức định tuyến xác định đường đi tối ưu của thông tin có tính đến sự thay đổi thường xuyên của cấu trúc liên kết.Thêm vào đó, sẽ rẻ hơn nếu nhận ra loại mạng này thay vì các loại khác (vệ tinh, di động, v.v.).

Mạng Ad Hoc có thể được phân loại thành ba loại chính là Mobile Ad hoc NETwork (MANET), Vehicle Ad hoc NETwork (VANET) và Flying Ad hoc NETworks (FANET). Nói chung, đối với hệ thống đa UAV, mô hình FANET được sử dụng. Bảng 2.2 mô tả chi tiết mỗi đặc điểm của danh mục [4], [34], phân tích dự đoán của chúng tôi.

Bảng 2.2: So sánh giữa MANET, VANET, FANET và mô hình mạng của chúng tôi.

Đặc điểm	MANET	VANET	FANET	Mô hình mạng của chúng tôi
Tính di động	Người ở một số địa hình nhất định	Phương tiện ở cao tốc	Máy bay trong kế hoạch 3D	3D
Mức độ di động	+	+	++	++
Chế độ di động	Điểm cách ngẫu nhiên với hướng và tốc độ ngẫu nhiên	Có khả năng dự đoán cao trên đường	Không được xác định trước (mô hình chuyển động ngẫu nhiên của UAV, mô hình dựa trên pheromone)	Có thể dự đoán
Thay đổi cấu trúc liên kết	+	+	++	++
Khoảng cách giữa các nút	+	+	++	+
Mật độ nút	++	++	+	+
Mô hình truyền thanh	Sự hiện diện hiếm hoi của đường ngầm	Sự hiện diện hiếm hoi của đường ngầm	Sự hiện diện thường xuyên của đường ngầm	Luôn có mặt của đường ngầm
Công suất tiêu thụ và công suất tính toán thời gian sống	+	+	++	++
Tiêu chuẩn	IEEE 802.11, 802.15, 802.15.4, 802.16 and 802.20	IEEE 802.11p	IEEE 802.11	?

So sánh cho thấy FANET là mạng gần nhất với kỳ vọng của mô hình của chúng tôi. Do đó, trong số các tiêu chuẩn hiện có cho loại mạng này, IEEE 802.11b và IEEE 802.11g cung cấp tốc độ dữ liệu cao và thường được sử dụng cho các hệ thống nhiều robot.

2.2.3 Phân loại cấu trúc liên kết mạng

Cấu trúc liên kết mạng xác định điểm đầu, điểm cuối và đường dẫn của thông tin trong mạng. Để cho các điểm mục tiêu, cấu trúc liên kết xác định tất cả các con đường có thể để đạt được chúng.

Từ quan điểm phần mềm, mạng chủ yếu có thể được phân loại theo ba cấu trúc liên kết (Xem Hình 2.2¹):



Hình 2.2: Mạng tập trung (trái), phi tập trung (giữa) và mạng phân tán (phải).

- **Mạng tập trung:** Đây là cấu trúc liên kết được sử dụng nhiều nhất [36], [19]. Tất cả các nút - còn được gọi là các trạm - được kết nối với nhau thông qua một trung tâm tập trung. Đó là khi một nút cần gửi một thông tin đến một nút khác, dữ liệu nhất thiết phải đi qua nút trung tâm để được chuyển đến đích của nó. Nếu nút trung tâm không hoạt động hoặc không thể truy cập được, không có dữ liệu nào có thể được truyền đi, điều này làm cho cấu trúc liên kết này dễ bị lỗi nút.
- **Mạng phân cấp:** Cấu trúc liên kết này được coi là một tập hợp các mạng tập trung được kết nối với nhau [27], [7]. Một trung tâm chính được sử dụng cho mỗi nhóm nút nhỏ để gửi và nhận thông tin. Nếu một trung tâm bị lỗi, các nút liên quan trực tiếp đến nó sẽ không thể truy cập được.
- **Mạng phân tán:** Cấu trúc liên kết này chứa các nút được kết nối với nhau. Không giống như cấu trúc liên kết tập trung, một số đường dẫn

¹Source: From user 1983 on Wikipedia. Licensed CC BY-SA

có thể cùng đến một điểm. Ngoài ra, các nút không bị ảnh hưởng bởi lỗi của một nút. Điều này cho phép tăng khả năng sống sót và giảm hư hỏng cho mạng. Cấu trúc liên kết phân tán có tính hấp dẫn để đảm bảo mạng đáng tin cậy [14]. Tuy nhiên, khá khó để tiếp cận và nhận ra trong thực tế, và mô phỏng không được phân phối đầy đủ [46], [11], [39]. Điều này là do tất cả các quá trình xử lý và ra quyết định phải ở trên robot vốn đòi hỏi một bộ nhớ lớn và khả năng tính toán nặng.

2.3 Cấu trúc mạng

Cấu trúc của mạng thể hiện kiểu của nó cũng như các đặc điểm đã được giới thiệu.

2.3.1 Công việc liên quan

Giao tiếp MRS là một vấn đề quan trọng cần giải quyết trong cộng đồng robotics. Trong thập kỷ qua, sự tiến bộ vượt bậc trong công nghệ không dây và MRS đã tạo ra mối quan tâm ngày càng tăng đối với giao tiếp giữa các robot. Do đó, các phương pháp khác nhau đã được đề xuất.

Đối mặt với các tình huống thăm họa chặng hạn như ở trong một nhà kho công nghiệp, tác giả của [50] đề xuất sử dụng mạng LAN không dây, Bluetooth và ZigBee để tạo thành mạng Ad-Hoc. Một số robot của hạm đội tạo thành cơ sở hạ tầng mạng để hỗ trợ giao tiếp.

Tác giả của [36] sử dụng hai tiêu chuẩn mạng để xây dựng một hệ thống đa UAV. Mạng lưới không dây IEEE 802.11s được xây dựng bằng cách sử dụng UAV mang các nút lưới được kết nối trực tiếp với thiết bị điện tử. Mỗi nút lưới này hoạt động như một AP để tạo thành IEEE 802.11g.

Phương pháp luận hỗ trợ giảm thiểu các liên kết không dây cho các phương tiện bay được nối mạng dựa trên Linux được trình bày trong [**kuschnig2012profiling**.] Cách tiếp cận bao gồm hai tùy chọn để thu thập thông tin chất lượng liên kết và giám sát giao diện không dây 802.11.

Việc đánh giá tiêu chuẩn 802.11a được sử dụng giữa UAV và AP được thực hiện trong [30]. Kết quả cho thấy cường độ tín hiệu nhận (RSS) cho downlink giảm theo khoảng cách đến AP nhưng vẫn ở mức chấp nhận được cũng như thông lượng.

Biết rằng phô ánh sáng nhìn thấy rộng hơn nhiều so với phô Tần số vô tuyến, có vẻ thú vị khi sử dụng Visible Light Communication (VLC) dựa trên Light Emitting Diodes (LED). IEEE đã phát triển tiêu chuẩn 802.15.7 cho giao tiếp tầm ngắn sử dụng ánh sáng khả kiến. Tác giả của [48] giới thiệu giao tiếp hai chiều bằng Open VLC. Phần cứng giải pháp này cần bo mạch Beagle Bone Black (BBB) và bộ thu phát phòng chữ sử dụng một đèn LED duy nhất để truyền và nhận. Giải pháp được triển khai trên trình điều khiển Linux giao tiếp trực tiếp với đèn LED front-end và ngăn xếp mạng Linux. Ý tưởng chính để sử dụng lại cùng một đèn LED cho cả tín hiệu Truyền (chế độ TX) và Tín hiệu ánh sáng nhận (chế độ RX) là khi một nút đang truyền dữ liệu, nút kia có thể mong đợi biểu tượng LOW của bit 0 và sử dụng thời gian này để chuyển chế độ của nó từ RX sang TX và truyền dữ liệu.

Zigbee - dựa trên IEEE 802.15.4 - là một giao thức ở mức cao, đặc biệt hữu ích cho phạm vi liên lạc ngắn và mức tiêu thụ thấp. Một số ứng dụng của UAV sử dụng mạng này [2]. Giao tiếp có thể được thực hiện dưới ba dải tần tùy thuộc vào loại chipset: 868MHz cho tốc độ bit 20 kbps, 915MHz cho 40 kbps và 2,4 GHz cho 250 kbps.

Tác giả trong [45] thông qua các nhóm có quy mô khác nhau bao gồm các UAV chiến thuật và UAV di chuyển trong các khu vực địa lý được phân định. Các kỹ thuật ảo hóa được sử dụng để điều chỉnh việc nâng cấp bất kỳ chức năng nào, cho phép khả thi trong các dịch vụ không đồng nhất.

Trong một nhiệm vụ tìm kiếm và cứu hộ, ví dụ, các tác giả trong [39] đề xuất phát trực tuyến thời gian thực giữa các UAV trên một khoảng cách lớn. Nhiệm vụ được chia thành 5 giai đoạn chính: Lập kế hoạch trước để xác định các con đường tốt nhất trong khu vực tìm kiếm cho phép giảm thời gian của nhiệm vụ sau đó tìm kiếm chỉ theo các điểm chỉ dẫn trước; việc phát hiện mục tiêu và gửi các kế hoạch mới cho những người khác, tiếp theo là định vị lại bằng cách thiết lập một liên kết đa bước để đánh giá tình hình bởi trạm gốc và cuối cùng là phát trực tuyến video.

Đa UAV cần hệ thống truyền dẫn phức tạp để đảm bảo tính khả thi khi di chuyển trong không gian ba chiều. Trong [39], một ăng ten đẳng hướng đa hướng được bố trí trên trạm gốc và trên tất cả các UAV không đồng nhất được triển khai. Để trao đổi dữ liệu, công nghệ lưới chuẩn IEEE 802.11s được sử dụng.

Các tác giả trong [24] so sánh hiệu suất giữa IEEE 802.11n tiêu chuẩn và IEEE 802.11ac trong mạng đa bước. Các mạng thử đã được thử nghiệm trong môi trường trong nhà và ngoài trời, và ở hai chế độ, chế độ điểm truy cập (AP) / chế độ cơ sở hạ tầng và chế độ lưới liên quan đến thông lượng và công bằng. Trong các thử nghiệm trong nhà, đối với chế độ cơ sở hạ tầng, 802.11ac cho thấy hiệu suất được cải thiện trong cả thông lượng TCP và UDP, và mất gói đối với đường truyền UDP. Trong các thí nghiệm ngoài trời, đối với chế độ cơ sở hạ tầng, thông lượng của 802.11n cao hơn gấp ba lần so với 802.11a, tuy nhiên, chất lượng liên kết giảm mạnh hơn. Đối với chế độ lưới, 802.11n đạt được thông lượng cao hơn ở phạm vi gần nhưng giảm nhanh hơn ngay khi tốc độ ngày cao hơn và phạm vi dài hơn. Thông lượng được ghi lại cho mạng lưới thấp hơn chế độ cơ sở hạ tầng do thời gian truyền liên gói dài hơn.

Đối mặt với các vấn đề về bảo trì kết nối, tránh va chạm, tính chắc chắn khi gặp lỗi và cải thiện vùng phủ sóng, các tác giả trong [21] đề xuất một mô hình mới cung cấp nhiều cấu trúc liên kết mạng tốt hơn.

Trong [23], một lớp giao tiếp mới được đề xuất để đối phó với các mạng yêu cầu băng thông cao. Vì vậy, một số cơ chế được sử dụng để gửi thông báo, nén dữ liệu hoặc phản ứng với các tình huống bất ngờ.

Bảng 2.3 liệt kê một số tiêu chuẩn được sử dụng trong hệ thống nhiều robot. Đối với mỗi tiêu chuẩn, chúng tôi trình bày chi tiết lớp liên quan trong mô hình OSI, các đặc điểm của nó, kết quả hoạt động cũng như phần cứng và phần mềm được sử dụng cho các thí nghiệm.

Bảng 2.3: Ví dụ về một số tiêu chuẩn được sử dụng trong MRS.

Bài báo	Tiêu chuẩn	Lớp	Đặc trưng	Hiệu năng	Phần cứng	Phần mềm
[Scherer et al., 2015]	IEEE 802.11s mesh technology	Lớp Network	Tương thích với ché độ luối và ché độ AP	Range: 100m, communication delay: 5ms, Throughput: 10-20Mbit/s	Heterogeneous UAVs + laptop BS + wifi module	Middleware Robot Operating System ROS
[Hayat et al., 2015]	IEEE 802.11n	Lớp PHY và MAC	Thông lượng cao ở cả hai ché độ (AP và lƣới), mức độ công bằng chấp nhận được	Outdoor mesh mode + single hop; Range: 500m; TCP throughput: 35Mbit/s (50m); FB: 40Mhz	Two Pelican UAVs + laptop + Compex WLE300NX 802.11abgn mini PCIe modules	Ubuntu Linux Kernel 3.2. with ath9k driver
	IEEE 802.11ac	Lớp PHY và MAC	Không hỗ trợ ché độ luối	Outdoor mesh mode + single hop; Range: 500m; TCP throughput: 10Mbit/s (50m); FB: 80Mhz	Two Pelican UAVs + laptop + Compex WLE900N518 802.11ac 5Ghz miniPCIe modules	Ubuntu Linux Kernel 3.2. with ath10k driver

2.3. CẤU TRÚC MẠNG

55

[Kuschnig et al., 2012]	IEEE 802.11a	Lớp PHY và MAC	RSS và thông lượng giảm dần theo khoảng cách nhưng vẫn ở mức chấp nhận được	Outdoor over campus Range: 10m; Throughput: 54Mbit/s (theo) and 27Mbit/s (prac); 5Ghz	AP: Netgear WNDR3700 + Atheros AR9280 based wireless cards + UAV with Intel Atom Processor + SparkLAN WPEA110N wireless card + antenna WIMO 18720.11 for UAV and AP	Linux-based OpenWRT Backfire 10.03.1RC5
[Asadpour et al., 2014]	IEEE 802.11n multi hop	Lớp PHY và MAC		Thời gian hội tụ lâu và chỉ phủ định tuyến cao (với giao thức BATMAN cho Lớp mạng)	In-flight experiment; Throughput for 200m: 5.95Mbit/s; Convergence time (20-100m): 28s; Routing overhead (20-100m): 10msg/s	Two Arducopter + GB + WLAN IEEE 802.11n + XBEE-PRO

[Morgen-thaler et al., 2012]	IEEE 802.11s mesh nodes for UAV to UAV and IEEE 802.11g for UAV to AP	Lớp Network	Trong một bước nhảy, các UAV đặt được thông lượng cao hơn các UAV trên mặt đất. Các kết quả này được thực hiện với chẽ độ định vị dựa trên vị trí thấp hơn so với chẽ độ định vị cưỡng độ tín hiệu. Những màn trình diễn này cao hơn so với multi-hop.	Single hop (1UAV altitude 3-5m and AP-AP = 75m) in location positioning mode; TCP throughput = 6.5Mbit/s + in signal strength positioning mode; TCP throughput = 8.1Mbit/s	One UAVNet quadcopters Professional OM1P + two notebooks	Linux 2.6.37.6 Kernel generated by ADAM (embedded Linux distribution) + driver ath5k
[Muzaffar and Yannaz, 2014]	IEEE 802.11ab	Lớp PHY và MAC	Thông giảm khi số lượng nút tăng lên	Range: up to 1000m × 1000m × 50m; Throughput: 54Mbit/s (prac); FB: 5Ghz	Simulation: UAVs + Ground Station	Omnnet++

[Wang et al., 2014]	IEEE 802.15.7	Lớp PHY và MAC	Truyền hai chiều, kết hợp lọc và khôi phục lỗi thời gian có thể tăng phạm vi liên lạc và độ ổn định	<p><i>One hop:</i> throughput: 1.6kb/s, packet loss ratio 5%;</p> <p><i>Two hop:</i> throughput 0.65kb/s, packet loss ratio 15%</p>	<p>Embedded BBB board + LED front end</p>	Linux Kernel 3.8.13
---------------------	---------------	----------------	---	---	---	---------------------

2.3.2 Các tiêu chuẩn và giao thức mạng

Kể từ khi trao đổi thông tin là cần thiết để phối hợp, mạng được sử dụng cho truyền thông đa UAV phải đảm bảo rằng thông tin lưu thông đúng cách trong các nút. Do đó, chúng ta phải vượt qua một số ván đe như:

- Lỗi nút: Khi liên kết bị hỏng hoặc bị lỗi, mạng phải tìm đường dẫn để đến nút.
- Thay đổi cấu trúc liên kết: Các nút di chuyển trong môi trường và tạo ra những thay đổi trong cấu trúc liên kết mà mạng phải đối phó nhanh chóng.
- Băng thông liên lạc: UAV sở hữu một số thông tin nhất định để trao đổi với các thông tin khác và cần một số băng thông hỗ trợ trao đổi dữ liệu.

Có tính đến các ván đe được trích dẫn ở trên và để đảm bảo tích hợp tốt giữa SLAM trực quan và giao tiếp, mạng lưới không dây phân phối và hợp tác đường như là kiểu chữ mạng đầy đủ nhất. Nó cho phép giới thiệu những lợi thế sau:

- Cải thiện độ tin cậy của mạng vì mạng không có cơ sở hạ tầng yêu cầu một số đường dẫn, để thông tin có thể đến đích ngay cả khi chỉ có một liên kết bị hỏng.
- Cho phép khả năng mở rộng của mạng bằng cách thích ứng nhanh với các thay đổi cấu trúc liên kết.
- Cho phép triển khai nhanh chóng với chi phí back-haul.
- Cung cấp bảo hiểm dễ dàng trong các khu vực có quyền truy cập khó khăn.
- Tiết kiệm tuổi thọ pin do tiêu thụ điện năng thấp hơn.
- Cho phép đổi mặt với một số lượng lớn các UAV trong một hạm đội.

Trong số các tiêu chuẩn mạng lưới hiện có, sửa đổi 802.11, liên quan đến lớp Mac, thú vị cho các ứng dụng MRS. Nó là một mạng thiết lập dịch vụ mở rộng hỗ trợ giao tiếp phát sóng, phát đa hướng và unicast. Nó chứa giao thức Hybrid Wireless Mesh Protocol (HWMP) làm giao thức định tuyến mặc định. HWMP là một giao thức định tuyến lai lấy cảm hứng từ AODV (Ad-hoc On-demand Distance Vector: một phần theo yêu cầu và phần phản

ứng) và giao thức dựa trên cây (một phần chủ động). Qua đó, nó chứa các ưu điểm của giao thức phản ứng vì nó chuẩn bị bảng định tuyến khi các nút thay đổi vị trí của chúng và do đó cung cấp đường dẫn an toàn hơn. Mặt khác, nó chứa những ưu điểm của giao thức chủ động vì bảng định tuyến đã sẵn sàng cho phép tiết kiệm thời gian khi cần thiết. Ngoài giao thức định tuyến mặc định HWMP, mạng lưới 802.11S hỗ trợ các giao thức khác như Optimized Link State Routing (OLSR), Better Approach to Mobile Ad hoc Networking (BATMAN), Wireless Distribution System (WDS), Open Shortest Path First (OSPF) và BABEL. Theo [47], BATMAN đã chứng minh khả năng tốt hơn HWMP và OLSR. Do đó, các thí nghiệm sử dụng HWMP và BATMAN đã được thực hiện để chỉ ra dữ liệu đã lưu.

Định nghĩa giao thức BATMAN Giao thức BATMAN² là một giao thức định tuyến chủ động lấy cảm hứng từ AODV và OLSR. Nó được hỗ trợ bởi các mạng lưới multi-hop Ad Hoc. Nó sử dụng các phương pháp khác nhau để định tuyến nút chọn bằng cách gửi OriGinator Messages (OGM) cho hàng xóm với thông tin nút đến next hop và đích đến tiếp theo vì các quyết định định tuyến được phân phối trên các nút. Trong giao thức này, mỗi nút quyết định cho next hop và không cho toàn bộ tuyến đường để các nút không sử dụng hoặc thậm chí biết cấu trúc liên kết của mạng. Trong trường hợp phát hiện các nút khác, giao thức BATMAN sẽ dẫn con đường tốt nhất cho họ. Nó cũng theo dõi các nút mới và thông báo cho hàng xóm về sự tồn tại của họ.

2.3.3 Kết quả và thảo luận

Để đánh giá hệ thống nối mạng bằng cách sử dụng giao thức BATMAN³, Chúng tôi sử dụng các nút không đồng nhất gồm ba máy tính xách tay: 2.40GHz dual core Linux machine, 2.27GHz i3 Linux machine, 2.50GHz i5 Linux machine and a Parrot AR-Drone 2.0 (Xem Hình 2.3).

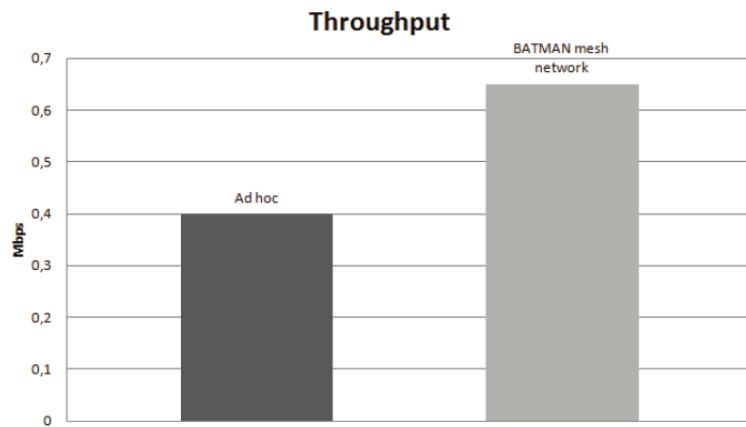
²Nguồn: <https://www.open-mesh.org/projects/open-mesh/wiki>

³The BATMAN-adv version used for the testbed is available since 2013



Hình 2.3: Mạng lưới minh họa giữa ba máy tính xách tay và một máy bay không người lái.

Chúng tôi mô phỏng việc phát sóng dữ liệu giữa các nút trong cả mạng Ad Hoc và lưới với giao thức BATMAN để nhấn mạnh dữ liệu đã lưu. Máy bay không người lái đã được điều khiển từ máy tính xách tay bằng cách sử dụng biên dịch chéo. Đầu tiên, chúng tôi thực hiện một mạng Ad Hoc giữa các điểm cuối và phát dữ liệu từ nút A đến nút B, C và D trong mạng. Sau đó, chúng tôi phát sóng - trong cùng điều kiện - từ nút A đến B, C và D với giao thức mạng lưới BATMAN. Kết quả như Hình 2.4 cho thấy thông lượng đạt được trung bình 0.4 Mbits/s; trong khi đó, thông lượng được đánh giá trong mạng lưới với giao thức BATMAN, đạt mức trung bình 0.65 MBits/s. Giao thức mạng lưới BATMAN cải thiện 1,5 lần thông lượng của mạng so với mạng Ad Hoc cơ bản.



Hình 2.4: Phát sóng thông lượng được phát sóng trong giao thức mạng Ad Hoc và BATMAN.

2.4 Cấu trúc liên kết và chiến lược mạng cho MRS robustness

Cấu trúc liên kết của mạng xác định các thuộc tính hình học của các nút (ở đây là robot). Chiến lược đề xuất trong công việc này thể hiện hành vi được áp dụng để đổi mới với các tình huống quan trọng và để đạt được MRS robustness.

2.4.1 Công việc liên quan

Thách thức trong giao tiếp MRS là duy trì mạng đáng tin cậy trong nhiệm vụ làm cho robot có thể thực hiện hợp tác thăm dò [38], [22]. Chiến lược được sử dụng để thăm dò ảnh hưởng đến việc trao đổi dữ liệu giữa các robot bao gồm loại, điểm đến và tần số.

Thông tin trao đổi giữa robot và một máy chủ có thể là khung hình chính và điểm bản đồ [40], hoặc chỉ các tính năng của các khung hình chính được chọn và ước tính relative-pose giữa robot và trạm mặt đất [19]. Nhưng chủ yếu, robot trao đổi các bản sao địa phương của họ về bản đồ và các pose của họ [20], [8], [41].

Lượng dữ liệu trao đổi có thể tăng nhanh về kích thước, có thể gây tắc nghẽn mạng và mất dữ liệu. Để giảm yêu cầu băng thông, các tác giả của [35] Đề xuất chỉ gửi các key-frames và các vị trí key-frames được cập nhật. Ngoài ra, các tác giả trong [14] đề xuất một cách tiếp cận Decentralized Data Fusion-Smoothing And Mapping (DDF-SAM), trong trường hợp mỗi robot tuyên truyền tới các robot khác, đồ thị cục bộ cô đọng của nó để đạt được khả năng mở rộng và khả năng chống lại lỗi nút.

Hầu hết các công việc liên quan đến giải quyết các vấn đề giao tiếp trong khi giả định một mạng lưới lý tưởng hoặc mục đích giữ cho các thành viên trong phạm vi của nhau để tập trung sự chú ý vào các vấn đề quan trọng hơn [39], [10].

Xem xét tồn thất giao tiếp và/hoặc băng thông hạn chế giúp ngăn ngừa thất bại nhiệm vụ và để đảm bảo một kịch bản thực tế hơn. Thật vậy, trong các kịch bản thực tế, nhiều vấn đề có thể phát sinh như có khoảng cách giữa các robot vượt quá phạm vi giao tiếp, mất thông tin chính trong một liên kết giao tiếp bị hỏng, tốn thời gian trong việc gửi thông tin do băng thông hạn chế, v.v.. Chiến lược thăm dò phải tính đến các vấn đề được đề cập để

tránh nhiệm vụ thất bại trong kịch bản thế giới thực. Một số công việc đã bắt đầu giải quyết vấn đề thăm dò trong khi xem xét các giới hạn giao tiếp [13], [40].

Trong [15], mục đích là để cảm nhận một môi trường phức tạp về hình học bằng cách gán các mục tiêu cho robot khi độ phân giải không gian và thời gian là hợp lý. Cách tiếp cận này sử dụng thuật toán lập kế hoạch đường dẫn năng lượng min-max tuân theo thời hạn.

Trong công việc này, chúng tôi lựa chọn để cho UAV trao đổi với nhau chỉ qua các điểm biên giới, robot đặt ra và các mục tiêu được chỉ định. Trao đổi này xảy ra ở mỗi lần lặp trong khi xem xét vai trò của UAV, được điều chỉnh theo cấu trúc liên kết mạng. Thích ứng này cũng cho phép đối phó với những hạn chế giao tiếp.

2.4.2 Phương pháp giao tiếp giữa các robot

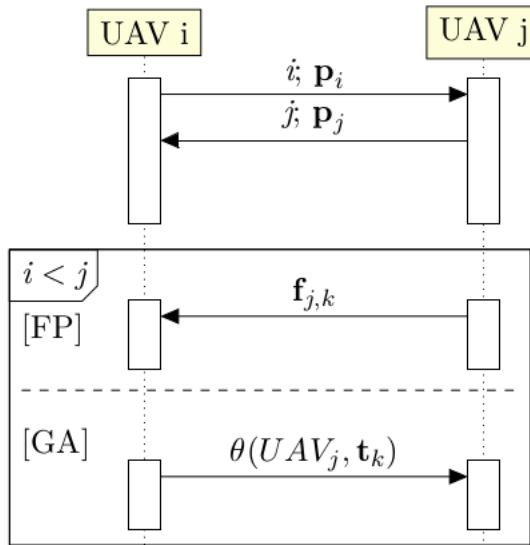
Tương tác giữa các thành viên của đội tàu rất quan trọng đặc biệt là trong các nhiệm vụ thăm dò để ngăn chặn UAV khám phá các khu vực giống nhau và cho phép họ hợp tác khám phá các khu vực chưa biết nhanh hơn và một cách tối ưu hóa. Tuy nhiên, liên lạc giữa các UAV là một vấn đề đầy thách thức đòi hỏi phải trả lời một số câu hỏi thực tế: Loại nút dữ liệu nào phải trao đổi? Làm thế nào để dữ liệu nên được chia sẻ một cách thường xuyên? Chúng ta có nên xem xét một trao đổi dữ liệu nhiều hop? Nếu vậy, làm thế nào để xác định điểm cuối của trao đổi dữ liệu? Làm thế nào để đối phó với những hạn chế giao tiếp? Những câu hỏi này được giải quyết trong các tiểu mục sau.

Tương tác đa UAV và trao đổi dữ liệu

Trong chiến lược hợp tác thăm dò đề xuất, điểm biên giới địa phương $\mathbf{f}_{i,j} \in \mathcal{F}$, pose hiện tại \mathbf{p}_i , và điểm mục tiêu hiện tại \mathbf{t}_m được trao đổi thay vì toàn bộ bản sao của bản đồ địa phương. Điều này dự kiến sẽ tạo ra sự giảm đáng kể khối lượng dữ liệu trao đổi, và do đó, giảm tiêu thụ bộ nhớ. Sơ đồ trình tự⁴ trong Hình 2.5 thể hiện chi tiết (thời gian và thông tin) các tin nhắn trao đổi giữa hai UAV. UAV_i, với $i \in [1..n_c]$ và UAV_j với $j \in [2..n_c]$ ($i < j$) là các robot trong cụm \mathcal{C} . Họ chuyển tiếp ID tương ứng và poses hiện tại của họ \mathbf{p}_i và \mathbf{p}_j . Từ $i < j$, explorer UAV_j gửi đến leader UAV_i điểm biên cục bộ của nó $\mathbf{f}_{j,k}$ trong bước xử lý biên giới (FP).

⁴This diagram uses Unified Modeling Language's sequence diagram notation.

2.4. CẤU TRÚC LIÊN KẾT VÀ CHIẾN LƯỢC MẠNG CHO MRS ROBUSTNESS63



Hình 2.5: Luồng dữ liệu giữa hai robot. FP và GA đứng tương ứng, để xử lý biên giới và phân công mục tiêu.

Sau đó, *leader* thực hiện quy trình phân công mục tiêu (GA) và gửi lại UAV_j điểm mục tiêu đã chọn t_m . Ví dụ được trích dẫn đại diện cho hai UAV trong cụm \mathcal{C} . Trong trường hợp nhiều UAV trong \mathcal{C} , các chuỗi tương tự sẽ được thực hiện giữa một *leader* và nhiều *explorers*.

Chiến lược exploration để đổi mặt với mốc giao tiếp

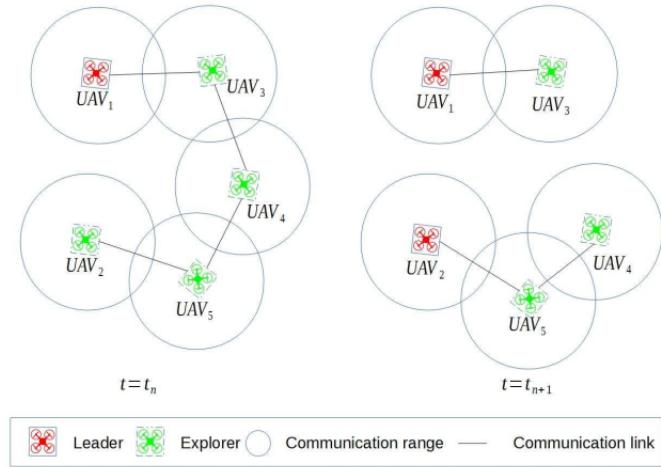
Trong hệ thống đề xuất, xem xét các hạn chế truyền thông rất quan trọng để đảm bảo tính liên tục của nhiệm vụ. Trong trường hợp mốc liên lạc với *leader* do thất bại giao tiếp hoặc UAV bị kẹt, một *leader* khác được tự chọn trong lần lặp tiếp theo để nhiệm vụ có thể tiếp tục. Trong Hình 2.6, tại $t = t_n$, hạm đội bao gồm một cụm nơi UAV có thể giao tiếp với nhau. Một *leader* xử lý các quyết định cho người khác. Tại $t = t_{n+1}$, liên kết truyền thông thất bại giữa UAV_3 và UAV_4 . Hạm đội được chia thành hai cụm với mỗi một *leader*.

Trường hợp đặc biệt Trong trường hợp mốc liên lạc với *leader* và một *leader* đã được chọn, *explorers* để một bộ đếm thời gian τ hết hạn trong khi chờ chuyển nhiệm vụ mục tiêu. Nếu không có mục tiêu được nhận, *explorer* chọn mục tiêu riêng của nó theo thông tin địa phương.

Sử dụng chiến lược này, miễn là - ít nhất - một UAV tồn tại trong hạm đội, nhiệm vụ sẽ tiếp tục cho đến khi tất cả các môi trường giới hạn được khám phá (không có điểm biên giới ứng cử viên nào còn lại).

Thảo luận chiến lược trao đổi dữ liệu

Trong chiến lược được đề xuất, trao đổi dòng dữ liệu được lặp lại tại mỗi lần lặp trong khi có tính đến các thay đổi cấu trúc liên kết mạng để xác định các cụm. Các điểm bắt đầu và điểm cuối được xác định theo các vai trò này. Vai trò của UAV cũng chỉ định loại dữ liệu trao đổi. Ngoài việc trao đổi pose hiện tại \mathbf{p}_i và id số i , nếu UAV $_i$ là một *explorer*, nó sẽ thu động chia sẻ thông tin về chính nó và môi trường xung quanh với *leader* (điểm biên giới $\mathbf{f}_{i,j} \in \mathcal{F}$); khác, vai trò của nó sẽ là gửi mục tiêu để truy cập đến *explorers* (điểm mục tiêu $\mathbf{t}_k \in \mathcal{G}$).



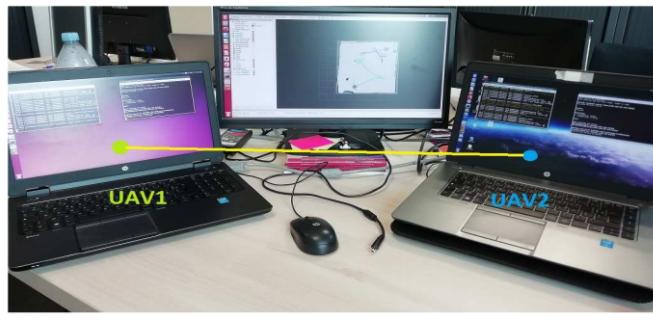
Hình 2.6: Sự tiến hóa vai trò trong phạm vi giao tiếp hạn chế.

Chiến lược đề xuất đảm bảo tính một nhiệm vụ liên tục trong trường hợp mất liên lạc. Tuy nhiên, UAV có thể khám phá các khu vực đã được các nút khác khám phá, vì không có bản đồ địa phương nào được trao đổi cũng không được hợp nhất để theo dõi các khu vực truy cập. Do đó, trong trường hợp mất truyền thông, thành tựu được ưa chuộng đối với việc giảm thiểu tiêu thụ tài nguyên, chẳng hạn như thời gian và pin.

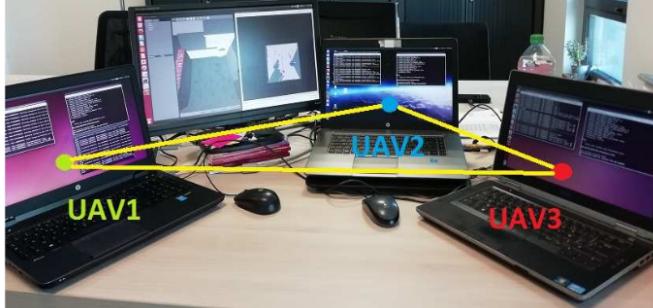
2.4. CẤU TRÚC LIÊN KẾT VÀ CHIẾN LƯỢC MẠNG CHO MRS ROBUSTNESS65

2.4.3 Kết quả và thảo luận

Vì UAV được trang bị thẻ không dây IEEE 802.11b,g, chúng tôi thiết lập một mạng lưới cơ sở hạ tầng trong tập hợp các robot để định lượng trao đổi dữ liệu giữa các thành viên của đội tàu, cũng như, để xác định hiệu suất của mạng robot. Thực hiện chạy với 2 và 3 UAV (Xem Hình 2.7). Mạng được bao gồm hai thiết bị là một 2.60GHz i7 Linux machines và một 2.50GHz i7 Linux machine. Số lượng robot được sử dụng để đánh giá được giới hạn ở 3, tuy nhiên, kiến trúc hệ thống được đề xuất không bị hạn chế với một số robot cố định.



(a) Hai UAV (Laptops)



(b) Ba UAV (Laptops).

Hình 2.7: Minh họa mạng Ad Hoc trong nhiệm vụ thăm dò.

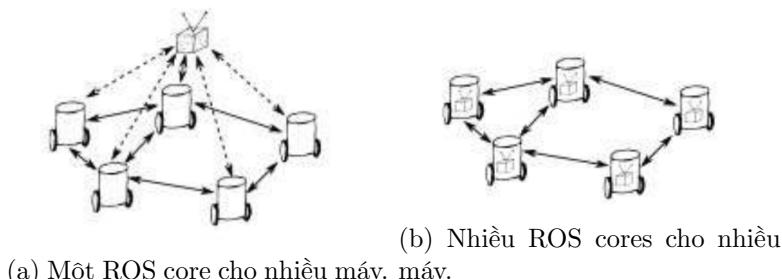
Cài đặt mạng: Từ một đến nhiều máy

Khi chạy nhiều UAV trên một máy duy nhất (như trong các mô phỏng trước đó trong Chương 3), một ROS master có trách nhiệm quản lý giao tiếp intra và inter-processes bằng cách sử dụng publisher/subscriber. Trong trường hợp nhiều máy, hai cấu hình có thể:

- Một ROS core cho nhiều máy (Xem Hình 2.8a): Ngay cả với nhiều UAV trên các máy khác nhau, một ROS master/core có thể được áp dụng bằng cách chỉ định máy chạy ROS core. Trong trường hợp này, SoS sẽ quản lý giao tiếp inter-processes theo cùng một cách như thể chúng nằm trên cùng một máy. Ngoại lệ là giao tiếp trong thế giới thực được sử dụng thay vì bộ nhớ dùng chung để chạy nhiều UAV trên một máy.
- Nhiều ROS cores cho nhiều máy (Xem Hình 2.8b): Khi chạy các ROS core khác nhau trên các máy khác nhau, mỗi chiếc UAV quản lý chủ của chính nó. Trong hệ thống đa lõi này - còn được gọi là hệ thống đa chủ -, đồng bộ hóa trong số các master này cần phải được thực hiện.

Tuy nhiên, đối với cả hai trường hợp, môi trường ảo sẽ được khám phá trong *Gazebo* phải giống nhau để UAV khám phá cùng một môi trường cùng một lúc. Cụ thể, điều này có nghĩa là IP client của *Gazebo* phải phù hợp với IP server của máy chạy *Gazebo* thế giới bằng cách thiết lập *GAZEBO_MASTER_URI*.

Khi chạy một ROS core cho nhiều UAV, một mạng đáng tin cậy là cần thiết, quá trình ROS khác sẽ không hoạt động đúng khi kết nối mạng không ổn định. Ngoài ra, chạy hệ thống multi-master là thực tế hơn bởi vì trong thế giới thực, mỗi UAV phải là thuộc về chức năng, độc lập và hợp tác để đạt được các mục tiêu nhiệm vụ, trong công việc này, việc khám phá một môi trường không xác định. Một hệ thống multi-master đại diện cho cấu hình hệ thống phân tán, có những ưu điểm khác nhau như khả năng mở rộng cho khả năng chịu lỗi.



Hình 2.8: Hai cấu hình ROS trong trường hợp nhiều máy. Ảnh từ [1].

Hệ thống Multi-master yêu cầu đồng bộ hóa. Đối với điều này, các gói

2.4. CẤU TRÚC LIÊN KẾT VÀ CHIẾN LUỢC MẠNG CHO MRS ROBUSTNESS67

khác nhau tồn tại như gói *multi-master fkie*⁵ cho phép truyền tải unicast cũng như phát đa hướng bằng giao thức UDP. Gói *wifi_comm*⁶ thực hiện Optimized Link State Routing (OLSR), nhưng có thể được sử dụng với các thuật toán định tuyến khác nhau. Gói *recon_multimaster*⁷ là một hệ thống đa chủ tập trung thực hiện các khồi xây dựng xung quanh lớp giao tiếp ROS nhưng không thực hiện giao tiếp với chính nó. Tác giả trong [1] đề xuất một cách tiếp cận phân tán nơi mỗi robot chạy một quản lý master giao tiếp địa phương bằng cách sử dụng gói *Adhoc_communication*⁸ ở trường hợp giao thức AODV được thực hiện. Một gói multi-master khác là gói *Nimbro_network*⁹ cung cấp vận chuyển mạnh mẽ các chủ đề và dịch vụ của ROS trên các mạng không đáng tin cậy. Các gói multi-master được trích dẫn ở trên không phải là một danh sách toàn diện và các gói đồng bộ hóa khác tồn tại.

Để đơn giản, như một cách tiếp cận đầu tiên, gói *multi-master fkie*¹⁰ được sử dụng để chạy hệ thống đa lõi được thông qua. Gói này cho phép chúng tôi sử dụng và đồng bộ hóa nhiều lõi bằng giao thức UDP mặc định. Đối với trao đổi dữ liệu chủ đề của ROS, giao thức TCP được sử dụng. Đối với một đánh giá hiệu quả, đặc biệt là liên quan đến thời gian, đồng bộ hóa đồng hồ cần phải được đảm bảo. Network Time Protocol (NTP) được sử dụng để đồng bộ hóa máy tính xách tay trong vòng vài mili giây của thời gian phổ cập phối hợp (UTC).

Sự đánh giá kích thước dữ liệu được trao đổi

Dánh giá đầu tiên nhắm vào việc chỉ ra số lượng dữ liệu trao đổi bằng cách chia sẻ các điểm biên giới cục bộ thay vì bản đồ lưới cục bộ (Xem Hình 2.9).

⁵Nguồn: http://wiki.ros.org/multimaster_fkie

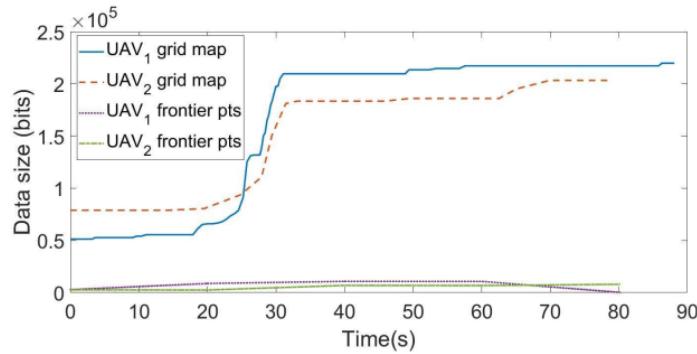
⁶Nguồn: http://wiki.ros.org/wifi_comm

⁷<http://wiki.ros.org/rocon>

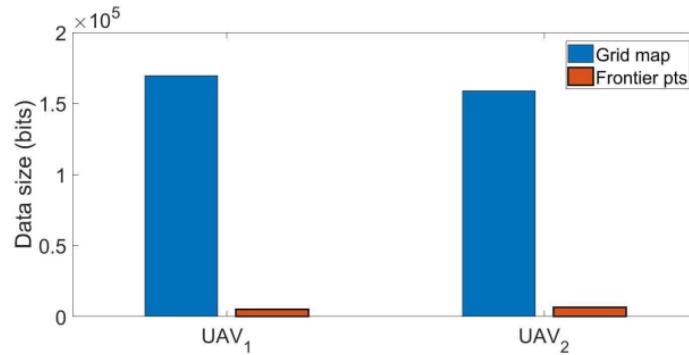
⁸Nguồn: http://wiki.ros.org/adhoc_communication

⁹Nguồn: https://github.com/AIS-Bonn/nimbro_network

¹⁰Nguồn: http://wiki.ros.org/multimaster_fkie



(a) Sự tiến hóa của số lượng dữ liệu trao đổi trong nhiệm vụ.



(b) Trung bình của số lượng dữ liệu trao đổi trong nhiệm vụ.

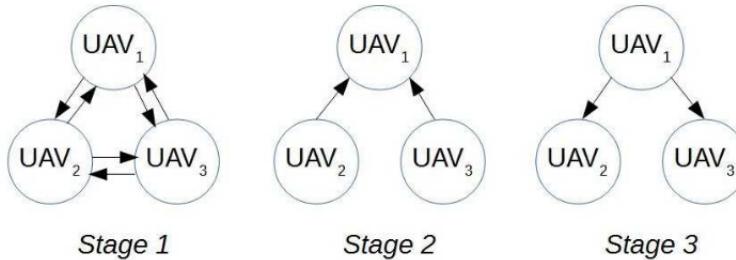
Hình 2.9: Kích thước dữ liệu khi UAV trao đổi toàn bộ bản đồ lối đi với các UAV khác nhau so với các điểm biên giới của nó.

Hình 2.9a cho thấy kích thước của bản đồ lối đi tăng theo thời gian so với kích thước của các điểm biên giới gần như không đổi trong nhiệm vụ. Theo kết quả trong Hình 2.9b, kích thước của dữ liệu được lưu, khi trao đổi điểm biên giới thay vì bản đồ kẹp, được chia cho 10.

Đánh giá thời gian trung bình của dữ liệu được trao đổi

Tùy thuộc vào kích thước và tần số của dữ liệu được trao đổi, thời gian được phân bổ cho giao tiếp có thể tăng với số lượng robot ngày càng tăng. Do đó, đánh giá thời gian hành vi và tác động tiềm năng của nó đối với các buổi biểu diễn thăm dò đã được tiến hành. Hình 2.10 hiển thị tiến hóa cấu trúc liên kết mạng trong quá trình trao đổi dữ liệu.

2.4. CẤU TRÚC LIÊN KẾT VÀ CHIẾN LƯỢC MẠNG CHO MRS ROBUSTNESS69



Hình 2.10: Tiến hóa cấu trúc liên kết mạng trong một vòng lặp với ba UAV hợp tác.

Thông tin được trao đổi trong ba giai đoạn như sau:

- Giai đoạn 1: id và poses hiện tại \mathbf{p}_i với $i \in [1..n_c]$.
- Giai đoạn 2: Các điểm biên giới $\mathbf{f}_{i,j}$ với $i \in [1..n_c]$ và $j \in [1..n_i]$.
- Giai đoạn 3: Phân công các điểm mục tiêu $\theta(UAV_i, \mathbf{t}_k)$ với $i \in [1..n_c]$ và $k \in [1..n_c]$.

Bảng 2.4 cho thấy thời gian trung bình dành cho việc trao đổi dữ liệu trong quá trình thăm dò. Một sự tăng nhẹ trong thời gian tính toán trung bình xảy ra khi tăng số lượng robot. Thời gian dành cho giao tiếp là tương đối không đáng kể so với tổng thời gian thăm dò.

Bảng 2.4: Thời gian cho module giao tiếp.

UAVs		Time spent in stage 1 (s)			Time spent in stage 2 (s)			Time spent in stage 3 (s)			Time for exploration (s)
		UAV ₁	UAV ₂	UAV ₃	UAV ₁	UAV ₂	UAV ₃	UAV ₁	UAV ₂	UAV ₃	
Two UAVs	UAV ₁	θ	0.136 ± 0.139		θ	θ		θ	0.022 ± 0.017		120, 1
	UAV ₂	0.065 ± 0.068			0.026 ± 0.008			θ	θ		
Three UAVs	UAV ₁	θ	0.056 ± 0.065	0.575 ± 0.769	θ	θ	θ	0.335 ± 0.407	0.765 ± 0.678		86
	UAV ₂	0.107 ± 0.111		0.483 ± 0.678	0.185 ± 0.244	θ	θ	θ	θ		
	UAV ₃	0.267 ± 0.165	0.616 ± 0.549		0.251 ± 0.109	θ	θ	θ	θ		

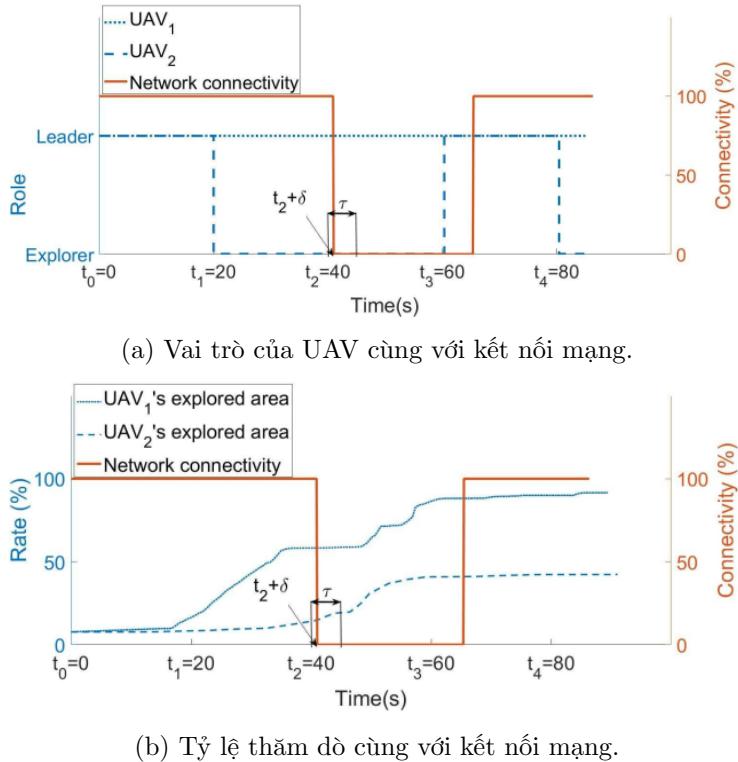
Dánh giá gián đoạn mạng

Để đánh giá hành vi hệ thống trong khi giao tiếp thất bại, kết nối mạng đã tự nguyện bị gián đoạn trong quá trình thăm dò. Hình 2.11 cho thấy vai trò của robot và hiệu suất tỷ lệ thăm dò khi kết nối mạng bị gián đoạn và sau đó phục hồi.

Hệ thống này thực hiện phát hiện hàng xóm, lựa chọn vai trò và chuyển nhượng mục tiêu tại mỗi vòng lặp của $t_{t+1} = t_i + i.r$ với $t_0 = 0s$ và $r = 20s$. Trong Hình 2.11a, tại $t = t_0$, cả hai robot bắt đầu với vai trò *leader*. Khi phát hiện ra nhau (tại $t = t_1$), UAV₁ chọn chính nó như là *leader* và UAV₂ trở thành *explorer*. Hậu quả là, UAV₁ gán một mục tiêu đến UAV₂. UAV₂ nhận được mục tiêu và cố gắng để đạt được nó. Tại $t = t_2 + \delta$, kết nối được tự nguyện bị gián đoạn ngay sau khi nhập vai trò và trước khi thông tin mục tiêu được gán cho UAV₂. Sau một khoảng thời gian τ , UAV₂ chọn một mục tiêu có tính đến dữ liệu cục bộ của riêng nó. Trong vòng lặp tiếp theo (tại $t = t_3$), vì kết nối vẫn bị gián đoạn, UAV₂ không tìm thấy hàng xóm và chọn chính nó như một *leader*. Cả hai robot thực hiện thăm dò độc lập, nghĩa là, không hợp tác. Ngay sau $t = t_3$, kết nối được thiết lập lại. Do đó, robot có thể hợp tác một lần nữa và UAV₂ đảm nhận vai trò của *explorer*. Trong Hình 2.11b, ngay cả sau khi kết nối mạng bị gián đoạn, cuộc thám hiểm sẽ tiếp tục được thực hiện bởi cả hai UAV. Khi kết nối được thiết lập

2.4. CẤU TRÚC LIÊN KẾT VÀ CHIẾN LƯỢC MẠNG CHO MRS ROBUSTNESS71

lại, *leader* thu thập các điểm biên giới và thực hiện xử lý biên giới nơi tìm thấy rằng không có mục tiêu ứng cử viên nào còn lại, đó là tất cả các môi trường hiện đã được khám phá. Do đó, nhiệm vụ được thực hiện.

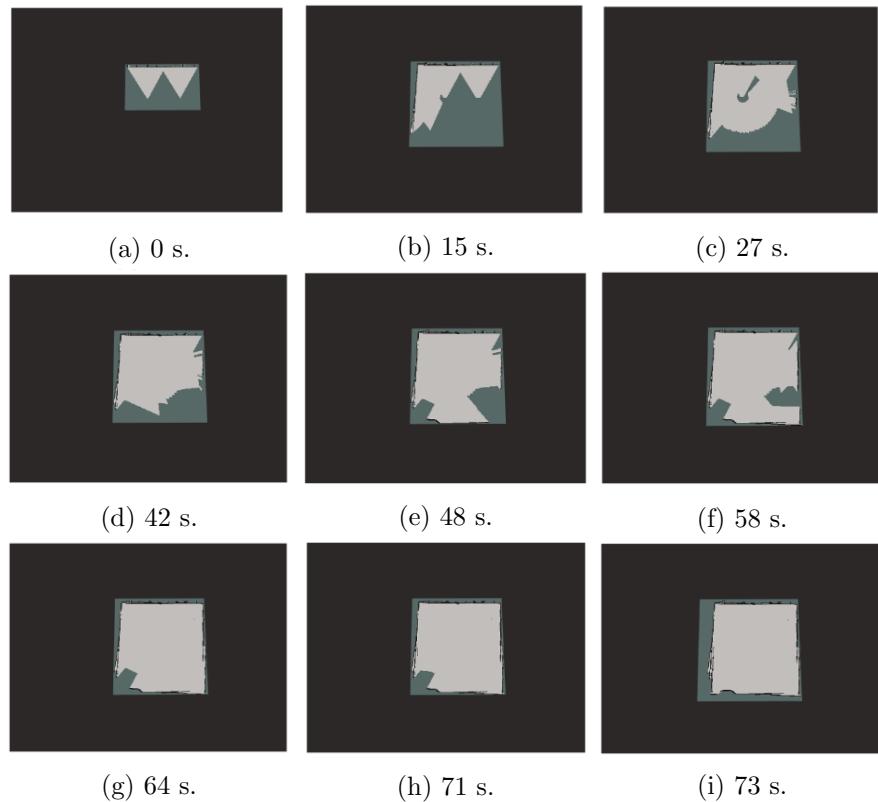


Hình 2.11: Hai robot hợp tác thăm dò cùng với kết nối mạng.

Đánh giá bản đồ toàn cầu

Nhiệm vụ thăm dò nhằm mục đích tạo ra một bản đồ toàn cầu về môi trường, một cách hiệu quả. Do đó, chúng tôi đã đánh giá bản đồ toàn cầu có được trong một nhiệm vụ hợp tác đang chạy với mạng lưới cơ sở hạ tầng.

Hình 2.12 minh họa sự phát triển của bản đồ lưới chiếm dụng 2D toàn cầu được xây dựng lại trong nhiệm vụ thăm dò.



Hình 2.12: Bản đồ lưới chiếm dụng 2D toàn cầu trong nhiệm vụ thăm dò với hai UAV hợp tác.

2.5 Conclusion

Trong chương này, chúng tôi đã giải quyết hai vấn đề con của mạng MRS. Điều đầu tiên liên quan đến cấu trúc mạng là các đặc điểm cần được đảm bảo bởi mạng; trong khi đó, vấn đề thứ hai chỉ ra sự phân phối địa lý của các nút trong mạng, được gọi là cấu trúc liên kết. Chúng tôi cũng đề nghị đổi mới với các tình huống quan trọng.

Cấu trúc mạng được đề xuất dựa trên việc áp dụng mạng lưới cùng với giao thức BATMAN. Kết quả của dữ liệu phát sóng được thực hiện trong thử nghiệm với các nút không đồng nhất bao gồm ba máy tính xách tay và một AR-Drone cho thấy giao thức mạng lưới BATMAN cải thiện thông lượng gấp 1,5 lần so với mạng Ad Hoc cơ bản.

Trong một phạm vi giao tiếp hạn chế, cấu trúc liên kết của mạng khác nhau trong nhiệm vụ. Kết quả là, Các hành vi của UAV được điều chỉnh bởi luôn cập nhật vai trò của họ thành *leader* hoặc *explorer* trong quá trình thực hiện nhiệm vụ. Ngay cả với một vai trò *explorer*, UAV có thể tiếp tục nhiệm vụ nếu mạng có kinh nghiệm ở một số vấn đề. Hơn nữa, chúng tôi đề xuất trao đổi các điểm biên giới của bản đồ địa phương thay vì toàn bộ bản sao của nó, cho phép giảm khối lượng dữ liệu dùng chung và do đó giảm tiêu thụ bộ nhớ. Kết quả của thử nghiệm được thực hiện với ba UAV, cho thấy module giao tiếp được đề xuất có thể đối phó với các giới hạn mạng. Họ cũng cho thấy rằng chiến lược được đề xuất sử dụng dữ liệu ít hơn gấp 10 lần so với chiến lược khiến robot trao đổi toàn bộ bản đồ địa phương.

Conclusion

Tài liệu tham khảo

- [1] Torsten Andre, Daniel Neuhold, and Christian Bettstetter. “Coordinated multi-robot exploration: Out of the box packages for ROS”. In: *2014 IEEE Globecom Workshops (GC Wkshps)*. IEEE. 2014, pp. 1457–1462.
- [2] Mahdi Asadpour et al. “Routing in a fleet of micro aerial vehicles: First experimental insights”. In: *Proceedings of the third ACM workshop on Airborne networks and communications*. 2014, pp. 9–10.
- [3] Antoine Bautin, Olivier Simonin, and François Charpillet. “Stratégie d’exploration multirobot fondée sur les champs de potentiels artificiels”. In: *Revue des Sciences et Technologies de l’Information-Série RIA: Revue d’Intelligence Artificielle* 26.5 (2012), pp. 523–542.
- [4] Ilker Bekmezci, Ozgur Koray Sahingoz, and Samil Temel. “Flying ad-hoc networks (FANETs): A survey”. In: *Ad Hoc Networks* 11.3 (2013), pp. 1254–1270.
- [5] Facundo Benavides et al. “Multi-robot Cooperative Systems for Exploration: Advances in dealing with constrained communication environments”. In: *2016 XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium (LARS/SBR)*. IEEE. 2016, pp. 181–186.
- [6] Ons Bouachir. “Conception et mise en oeuvre d’une architecture de communication pour mini-drones civils”. PhD thesis. Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), 2014.
- [7] Christoph Brand et al. “Stereo-vision based obstacle mapping for indoor/outdoor SLAM”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 1846–1853.

- [8] Guillaume Bresson, Romuald Aufrère, and Roland Chapuis. “A general consistent decentralized Simultaneous Localization And Mapping solution”. In: *Robotics and Autonomous Systems* 74 (2015), pp. 128–147.
- [9] Wolfram Burgard et al. “Collaborative multi-robot exploration”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 1. IEEE. 2000, pp. 476–481.
- [10] Wolfram Burgard et al. “Coordinated multi-robot exploration”. In: *IEEE Transactions on robotics* 21.3 (2005), pp. 376–386.
- [11] Stephen Cameron and Niki Trigoni. “Collaborative sensing by unmanned aerial vehicles”. In: (2009).
- [12] Titus Cieslewski, Elia Kaufmann, and Davide Scaramuzza. “Rapid exploration with multi-rotors: A frontier selection method for high speed flight”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 2135–2142.
- [13] Micael S Couceiro et al. “Darwinian swarm exploration under communication constraints: Initial deployment and fault-tolerance assessment”. In: *Robotics and Autonomous Systems* 62.4 (2014), pp. 528–544.
- [14] Alexander Cunningham, Manohar Paluri, and Frank Dellaert. “DDF-SAM: Fully distributed SLAM using constrained factor graphs”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, pp. 3025–3030.
- [15] Rui Dai et al. “Quality-aware UAV coverage and path planning in geometrically complex environments”. In: *Ad Hoc Networks* 73 (2018), pp. 95–105.
- [16] Kalyanmoy Deb. “Multi-objective genetic algorithms: Problem difficulties and construction of test problems”. In: *Evolutionary computation* 7.3 (1999), pp. 205–230.
- [17] Edsger W Dijkstra et al. “A note on two problems in connexion with graphs”. In: *Numerische mathematik* 1.1 (1959), pp. 269–271.
- [18] Jan Faigl, Miroslav Kulich, and Libor Přeučil. “Goal assignment using distance cost in multi-robot exploration”. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 3741–3746.

- [19] Christian Forster et al. “Collaborative monocular slam with multiple micro aerial vehicles”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 3962–3970.
- [20] Dieter Fox et al. “Distributed multirobot exploration and mapping”. In: *Proceedings of the IEEE* 94.7 (2006), pp. 1325–1339.
- [21] Cinara Ghedini, Carlos HC Ribeiro, and Lorenzo Sabattini. “Toward efficient adaptive ad-hoc multi-robot network topologies”. In: *Ad Hoc Networks* 74 (2018), pp. 57–70.
- [22] Lav Gupta, Raj Jain, and Gabor Vaszkun. “Survey of important issues in UAV communication networks”. In: *IEEE Communications Surveys & Tutorials* 18.2 (2015), pp. 1123–1152.
- [23] Hannes Harms et al. “Development of an adaptable communication layer with qos capabilities for a multi-robot system”. In: *Iberian Robotics conference*. Springer. 2017, pp. 782–793.
- [24] Samira Hayat, Evsen Yanmaz, and Christian Bettstetter. “Experimental analysis of multipoint-to-point UAV communications with IEEE 802.11 n and 802.11 ac”. In: *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE. 2015, pp. 1991–1996.
- [25] Lionel Heng et al. “Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments”. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1071–1078.
- [26] Elizabeth A Jensen and Maria Gini. “Rolling dispersion for robot teams”. In: *Twenty-Third International Joint Conference on Artificial Intelligence*. 2013.
- [27] Kurt Konolige et al. “Map merging for distributed robot navigation”. In: *Proceedings 2003 IEEE/RSJ international conference on intelligent robots and systems (IROS 2003)(Cat. No. 03CH37453)*. Vol. 1. IEEE. 2003, pp. 212–217.
- [28] Harold W Kuhn. “The Hungarian method for the assignment problem”. In: *Naval Research Logistics (NRL)* 52.1 (2005), pp. 7–21.
- [29] Miroslav Kulich, Tomáš Juchelka, and Libor Přeučil. “Comparison of exploration strategies for multi-robot search”. In: *Acta Polytechnica* 55.3 (2015), pp. 162–168.
- [30] Robert Kuschnig et al. *Profiling ieee 802.11 performance on linux-based networked aerial robots*. na, 2012.

- [31] Ryan Leigh, Sushil J Louis, and Chris Miles. “Using a genetic algorithm to explore A*-like pathfinding algorithms”. In: *2007 IEEE Symposium on Computational Intelligence and Games*. IEEE. 2007, pp. 72–79.
- [32] Nesrine Mahdoui, Vincent Frémont, and Enrico Natalizio. “Cooperative exploration strategy for micro-aerial vehicles fleet”. In: *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE. 2017, pp. 180–185.
- [33] Nesrine Mahdoui, Vincent Frémont, and Enrico Natalizio. “Cooperative frontier-based exploration strategy for multi-robot system”. In: *2018 13th Annual Conference on System of Systems Engineering (SoSE)*. IEEE. 2018, pp. 203–210.
- [34] Vasily A Maistrenko, Leonov V Alexey, and Volkov A Daniil. “Experimental estimate of using the ant colony optimization algorithm to solve the routing problem in FANET”. In: *2016 International Siberian Conference on Control and Communications (SIBCON)*. IEEE. 2016, pp. 1–10.
- [35] Gajamohan Mohanarajah et al. “Cloud-based collaborative 3D mapping in real-time with low-cost robots”. In: *IEEE Transactions on Automation Science and Engineering* 12.2 (2015), pp. 423–431.
- [36] Simon Morgenthaler et al. “UAVNet: A mobile wireless mesh network using unmanned aerial vehicles”. In: *2012 IEEE Globecom Workshops*. IEEE. 2012, pp. 1603–1608.
- [37] Maitreyi Nanjanath and Maria Gini. “Dynamic task allocation for robots via auctions”. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006*. IEEE. 2006, pp. 2781–2786.
- [38] Martijn N Rooker and Andreas Birk. “Multi-robot exploration under the constraints of wireless networking”. In: *Control Engineering Practice* 15.4 (2007), pp. 435–445.
- [39] Jürgen Scherer et al. “An autonomous multi-UAV system for search and rescue”. In: *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*. 2015, pp. 33–38.
- [40] Patrik Schmuck and Margarita Chli. “Multi-uav collaborative monocular slam”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3863–3870.

- [41] Martin J Schuster et al. “Multi-robot 6D graph SLAM connecting decoupled local reference filters”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 5093–5100.
- [42] Weihua Sheng et al. “Distributed multi-robot coordination in area exploration”. In: *Robotics and Autonomous Systems* 54.12 (2006), pp. 945–955.
- [43] Reid Simmons et al. “Coordination for multi-robot exploration and mapping”. In: *Aaa/Iaaa*. 2000, pp. 852–858.
- [44] Agusti Solanas and Miguel Angel Garcia. “Coordinated multi-robot exploration through unsupervised clustering of unknown space”. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. Vol. 1. IEEE. 2004, pp. 717–721.
- [45] I Vidal et al. *A multi-service multi-mav communication framework for future secure societies*. Tech. rep. Technical report, University Carlos III of Madrid and IMDEA Networks Institute, 2015.
- [46] Sonia Waharte, Niki Trigoni, and Simon Julier. “Coordinated search with a swarm of UAVs”. In: *2009 6th ieee annual communications society conference on sensor, mesh and ad hoc communications and networks workshops*. IEEE. 2009, pp. 1–3.
- [47] Jerry Chun-Ping Wang, Brett Hagelstein, and Mehran Abolhasan. “Experimental evaluation of IEEE 802.11 s path selection protocols in a mesh testbed”. In: *2010 4th International Conference on Signal Processing and Communication Systems*. IEEE. 2010, pp. 1–3.
- [48] Qing Wang, Domenico Giustiniano, and Daniele Puccinelli. “Open-VLC: Software-defined visible light embedded networks”. In: *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*. 2014, pp. 15–20.
- [49] Barry Brian Weger and Maja J Matarić. “Broadcast of local eligibility for multi-target observation”. In: *Distributed Autonomous Robotic Systems 4*. Springer, 2000, pp. 347–356.
- [50] Ulf Witkowski et al. “Ad-hoc network communication infrastructure for multi-robot systems in disaster scenarios”. In: *Proceedings of IARP/EURON workshop on robotics for risky interventions and environmental surveillance (RISE 2008), Benicassim, Spain*. 2008.

- [51] Wencen Wu and Fumin Zhang. “Robust cooperative exploration with a switching strategy”. In: *IEEE Transactions on Robotics* 28.4 (2012), pp. 828–839.
- [52] Brian Yamauchi. “Frontier-based exploration using multiple robots”. In: *Proceedings of the second international conference on Autonomous agents*. 1998, pp. 47–53.
- [53] Zhi Yan et al. “Team size optimization for multi-robot exploration”. In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Springer. 2014, pp. 438–449.
- [54] Jing Yuan et al. “A cooperative approach for multi-robot area exploration”. In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2010, pp. 1390–1395.
- [55] Linhu Zhao, Yashuhiro Tsujimura, and Mitsuo Gen. “Genetic algorithm for robot selection and work station assignment problem”. In: *Computers & Industrial Engineering* 31.3-4 (1996), pp. 599–602.