



**HAL**  
open science

# Alignment of Three Robots without Communication nor Localization\*

Frédéric Guinand, François Guerin, Mark Bastourous

► **To cite this version:**

Frédéric Guinand, François Guerin, Mark Bastourous. Alignment of Three Robots without Communication nor Localization\*. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), Oct 2019, Bari, Italy. pp.647-654, 10.1109/SMC.2019.8914442 . hal-02501345

**HAL Id: hal-02501345**

**<https://hal.science/hal-02501345>**

Submitted on 6 Mar 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Alignment of Three Robots without Communication nor Localization\*

Frédéric Guinand<sup>1,3</sup> and François Guérin<sup>2</sup> and Mark Bastourous<sup>1</sup>

**Abstract**—This paper presents a decentralized method for aligning three robots without communication and without explicit localization. Each robot is assumed to be equipped with an omnidirectional camera and is assumed to be able to detect the two other robots in the images produced by its own camera. Each robot decides of its own movements based on the continuous measures of angles between itself and the others. We present both the algorithm that performs such a task and its theoretical analysis. We prove the correctness of the algorithm when the robots are reduced to points, when collisions are not considered and when the initial formation is not an equilateral triangle. When collisions are considered, the probability for the algorithm to be successful is greater than  $1 - \frac{1}{36}$ . We then present some simulation for which a security distance  $d_{sec}$  has been introduced. During the execution of the algorithm by each robot, if the distance between any two robots is lower than this security distance we suppose that a collision is likely to occur and consider that the Algorithm has failed. Simulation results report that the larger  $d_{sec}$ , the bigger the percentage of failures. However runs are always successful when  $d_{sec} = 0$  which suggests that the theoretical bound is not tight and could be improved. In addition, simulations reveal that the main source of failures was not the one expected by the theoretical analysis.

## I. INTRODUCTION

Pattern formation by groups of robots has received an increased attention these last years. The availability of non expensive and easy-to-set up robotic platforms has contributed to the emergence of swarm robotics for which pattern formation and formation control are keys.

Regarding pattern formation issues, roboticists as well as researchers working in distributed computing have produced several models and results.

Sugihara and Suzuki were among the first to propose a distributed method for controlling multiple mobile robots [12]. In their work, the authors address the generic problem of the geometric pattern formation, including line formation. The robots are considered as identical and they are not supposed to communicate with each other. The method developed by the authors is not fully autonomous since the user, for the line formation, has to explicitly choose the two robots that will be the extremities. In addition it is assumed that each robot is able, thanks to a sensor, to determine the position of the other robots, in its own reference frame. The model used in this work was more formally defined in [13]. The

robots are assumed to be reducible to a point in the 2D space. Each robot is anonymous, has a memory, and its own coordinate system. In addition, after observing its environment the position of the other robots are known in its own coordinate system. The authors show in particular that oblivious robots can achieve the task of gathering altogether in a single point of the plan, a problem also addressed with mainly the same model in [3]. In the same line, in 2001 Paola Flocchini and her colleagues refined and constrained the model, but after the observation phase, a robot knows the position, in its own coordinate system, of the robots located in its surrounding. If some assumptions regarding the model are discussed in other papers, obliviousness, sense of direction, chirality, etc. [9], [5], the localization knowledge after the observation phase is never questioned.

Complementary to these theoretical approaches roboticists have proposed various solutions for pattern formation. Instead of considering that each robot is able to localize the other ones in its own coordinate system, they try to find methods for allowing the robots to mutually localize themselves. Note also that pattern formation is not necessarily the final goal of the process, usually they also try to maintain the geometric pattern during a mission-specific movement of the group.

One of the founding work is due to Balch and Arkin [4]. They propose a method for building different topologies: line, column, diamond and wedge. Every proposed approach requires the knowledge of the position (using GPS coordinates or dead reckoning) of some other robots. The methods consist for each robot to maintain its position with respect to a particular point that could be either, the central point of the formation or, the position of a leader or the position of a specific neighbor. Other approaches include the computation of mutual localization which is mainly obtained through information exchanges between robots. This mutual localization can also be achieved based on the use of bearing angles, as it was exposed in [10], but in that case the robots have to communicate with each other. In [7], based on some prior information about the robots height and other items in the environment, each robot performs a distance estimation and run some control law to form some specific pattern from any initial configuration. This is achieved without the need for the robots to communicate with each other. The harsh underwater conditions for both wireless communications and mutual localization have led Sousselier and his colleagues to propose a line formation algorithm based on local localization information using ultra-wave ping. Their swarm is supposed to work in a synchronous way.

Our contribution is to propose an algorithm for aligning

\* This work was supported by SolarFarm project no XXX

<sup>1</sup> Normandy Univ - UNIHAVRE - LITIS (France)  
frederic.guinand, mark.bastourous@univ-lehavre.fr

<sup>2</sup> Normandy Univ - UNIHAVRE - GREAH (France)  
francois.guerin@univ-lehavre.fr

<sup>3</sup> Institute of Computer Science, Faculty of Mathematics and Natural Sciences - Cardinal Stefan Wyszyński University in Warsaw (Poland)

three robots requiring neither communication nor localization. In our approach each robot measures the angle formed by itself with the two other robots and behaves according to both this angle and its derivative. So, our robots are not oblivious albeit they only need to store their last angle value. In addition our robots are not communicating with each other, they are anonymous, have no common coordinate system, may not have the same chirality and are supposed to move at a constant speed, not necessarily the same for each robot. However it is assumed that robots are able to perform simultaneously sensing, computing and moving.

Our motivation for proposing such a model is that on real robotic platforms multicore processors are now often present, allowing parallel activities to occur, whilst the achievement of constant and precise mutual localization is still difficult to obtain. Moreover, our final aim is to implement this method on our fleet of home-made robots.

It is also assumed that: there is neither wind nor any environmental problem/disturbance, the environment is free of obstacles and the ground is flat (essential for Algorithm 2).

The Alignment Algorithm is described in the next Section. We theoretically prove, in Section III that this algorithm is successful 97% of time. Simulations results are presented and analyzed in Section IV. Finally some directions for improving and extending the current method and some perspectives are discussed in the Conclusion.

## II. ALIGNMENT OF THREE ROBOTS WITHOUT COMMUNICATION NOR LOCALIZATION

### A. Introduction

Given three robots as in Figure 1 located on the same plane. As previously mentioned, each robot is assumed to be able to measure the angle it forms with the two other robots which corresponds to angle  $\gamma$  on the Figure.

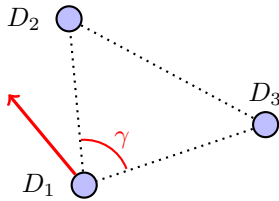


Fig. 1. Arbitrary position of the three robots. Each robot is able to measure its  $\gamma$  angle, the angle it forms with the two other robots.

The  $\gamma$  angle is the key element on which the alignment algorithm is built on.

### B. Alignment Algorithm

The algorithm presented in this section is independently executed by each robot and ends only when the alignment is obtained. It can be considered as a continuous process for reaching a predefined state, this state being a line. From the point of view of any robot, a line is obtained as soon as its

$\gamma$  angle is equal to 0 or to  $\pi$ . In this latter case, the robot is located between the two other ones.

Before reaching this desired state, at any moment the moving behavior of a robot is driven by both  $\gamma$  value and  $d\gamma/dt$ . This means that the movement of a robot may change if the value of  $\gamma$  is changing because of its own movements or because of the movements of the other robots (as illustrated by Figure 2):

- 1) if  $\gamma = 0$  or  $\gamma = \pi$ , the robot stops,
- 2) if  $\gamma \geq \pi/2$ , the robot moves into a direction corresponding to the bisector of this angle,
- 3) if  $\gamma \leq \pi/3$  the robot stops moving,
- 4) if  $\pi/3 < \gamma < \pi/2$ , the robot moves into the direction of one of the two other robots (randomly chosen),
- 5) a robot keeps on moving as long as  $\gamma$  increases, thus, as long as  $d\gamma/dt \geq 0$

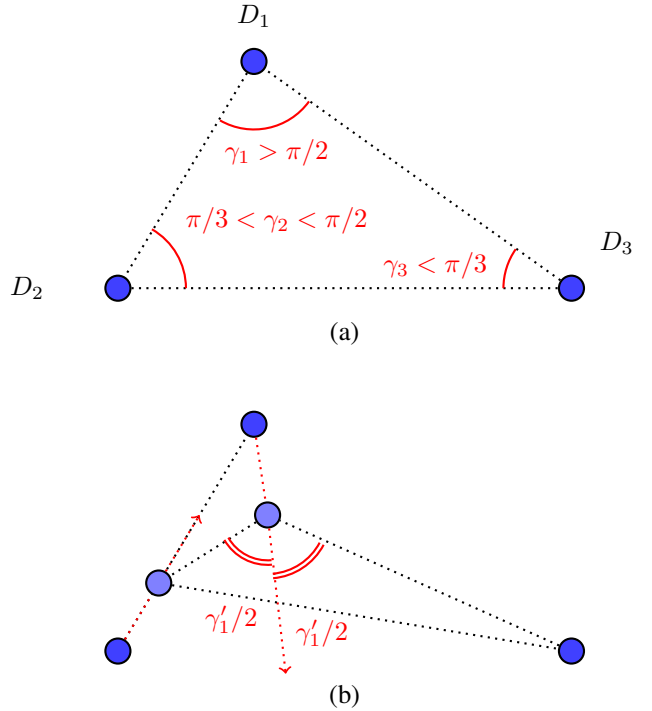


Fig. 2. Initially,  $D_3$  will not move since  $\gamma_3 < \pi/3$ ,  $D_1$  starts moving in the direction of the bisector of its  $\gamma$  value since  $\gamma_1 > \pi/2$  and  $D_2$  starts moving in the direction of one of the two other robots ( $D_1$  is randomly chosen in the present case), since  $\pi/3 < \gamma_2 < \pi/2$  (a). While  $d\gamma_1/dt \geq 0$  and  $d\gamma_2/dt \geq 0$ , both  $D_1$  and  $D_2$  keep on moving (b). Note that  $\gamma'$  denotes the new value of the  $\gamma$  angle.

## III. THEORETICAL ANALYSIS

In this section we propose a theoretical analysis of Algorithm 1. We always consider that robots are reduced to points but we distinguish two possibilities, when collisions are considered (at least two robots are located at the same position) or not.

In order to prove the correctness of the algorithm when collisions are not considered and its probability of success when collisions are considered, we start by proving some Lemmas.

---

**Algorithm 1** Alignment Algorithm
 

---

```

1: procedure ALIGN
2:   state  $\leftarrow$  INIT
3:    $\vec{D} \leftarrow$  NULL
4:   while  $((\gamma \neq \pi) \&\&(\gamma \neq 0)) \&\&(state \neq STOP)$  do
5:     if  $(\gamma \geq \pi/2)$  then
6:       if  $(d\gamma/dt \geq 0)$  OR  $(state = INIT)$  then
7:         move in the direction of the bisector of  $\gamma$ 
8:         state  $\leftarrow$  MOVE
9:       else
10:        state  $\leftarrow$  STOP
11:     else
12:       if  $\gamma \leq \pi/3$  then
13:         state  $\leftarrow$  STOP
14:       else
15:         if  $(d\gamma/dt \geq 0)$  OR  $(state = INIT)$  then
16:           if  $\vec{D} \neq NULL$  then
17:              $T \leftarrow$  robot randomly chosen
18:              $\vec{D} \leftarrow$  direction towards  $T$ 
19:             state  $\leftarrow$  MOVE
20:             move according to  $\vec{D}$ 
21:           else
22:             state  $\leftarrow$  STOP

```

---

**Lemma 1:** If the triangle is equilateral, the algorithm fails.

*Proof:* Based on the algorithm, for moving a robot should have  $\gamma > \pi/3$  and in an equilateral triangle  $\gamma_1 = \gamma_2 = \gamma_3 = \pi/3$ , thus no robot is moving and thus they cannot align. ■

In the sequel we only consider non equilateral triangles.

**Lemma 2:** Alignment Algorithm allows only six different scenarii of movement for the robots.

*Proof:* We first remark that there exist only 5 different configurations for the angles. Without loss of generality we assume that  $\gamma_1 \geq \gamma_2 \geq \gamma_3$ .

All possible combinations are:

- 1)  $\gamma_3 = \gamma_2 = \gamma_1 = \pi/3$ , already considered by Lemma 1
- 2)  $\gamma_3 \leq \gamma_2 \leq \pi/3 < \pi/2 \leq \gamma_1$
- 3)  $\gamma_3 \leq \gamma_2 \leq \pi/3 < \gamma_1 < \pi/2$
- 4)  $\gamma_3 \leq \pi/3 < \gamma_2 < \pi/2 \leq \gamma_1$
- 5)  $\gamma_3 \leq \pi/3 < \gamma_2 \leq \gamma_1 < \pi/2$

Based on the algorithm, and due to triangle properties (there cannot be three angles strictly greater than  $\pi/3$ ) at most two robots may move at the same time, thus  $D_3$  is not moving since  $\gamma_3$  is the smallest  $\gamma$  angle.

For combinations 2 and 3 and according to the algorithm, only  $D_1$  moves which leads to Scenario 1 on Figure 3. For the case 4,  $D_1$  is moving in the direction of the bisector of its  $\gamma$  angle and there exist two possibilities for the movement of  $D_2$ . Either  $D_2$  moves in the direction of  $D_3$ , leading to Scenario 2 on Figure 3, or in the direction of  $D_1$  corresponding to Scenario 3. Finally, for the last case (5), there exist three possible combinations of movements: (i)  $D_1$  and  $D_2$  are moving in opposite directions: Scenario 4, (ii) only one of the two robots,  $D_1$  or  $D_2$ , is moving to  $D_3$  leading to Scenario

5, or (iii) both  $D_1$  and  $D_2$  are moving in the direction of  $D_3$ , corresponding to Scenario 6. ■

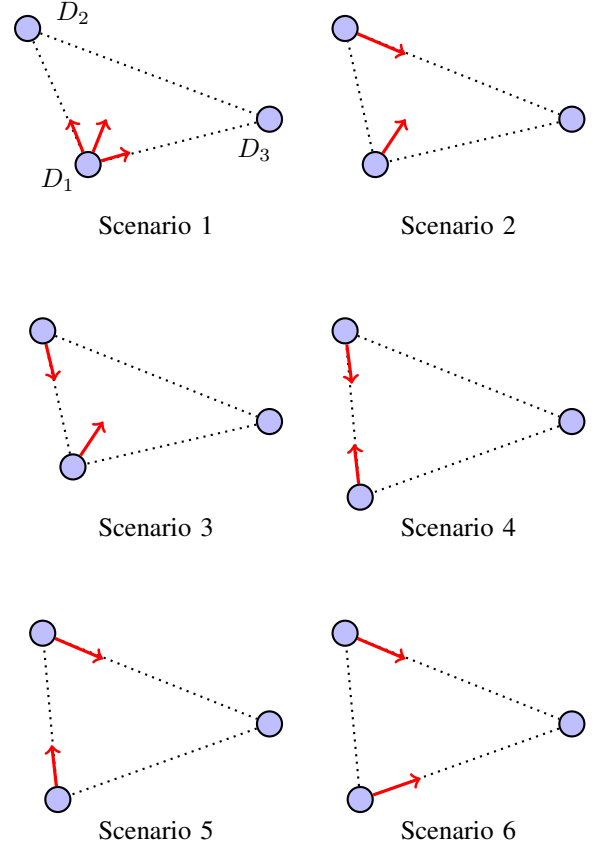


Fig. 3. The six different moving scenarii according to Algorithm 1.

**Lemma 3:** When  $\gamma_2 = \gamma_1$ , for scenarii 4 and 6 the algorithm reaches a degenerated alignment if collisions are not considered and it fails if not.

*Proof:* When  $\gamma_2 = \gamma_1$ , the triangle is isosceles. For scenario 4, if  $D_1$  and  $D_2$  start simultaneously and are moving exactly at the same speed then, both  $\gamma_2$  and  $\gamma_1$  angles increases up to  $\pi/2$  and both robots will be positioned at the same point. If collision is not an issue, we can consider this final configuration as an alignment since the three robots belong to a same line. For scenario 6, both robots are moving toward  $D_3$  and, if  $D_1$  and  $D_2$  start simultaneously and move at the same speed, both  $\gamma_2$  and  $\gamma_1$  remain identical during the move. The movement stops when the three robots are positioned at the same point. Which may also be considered as a degenerated line. If collisions are considered, in both situations, as at least two of them are located at the same position, we can consider that they collide and thus that the algorithm fails. ■

**Lemma 4:** For Scenario 1, the execution of Algorithm 1 on every robot leads to the alignment of the three robots.

*Proof:* For Scenario 1,  $\gamma_1 > \pi/2$ , only  $D_1$  is moving and it moves in the direction of the bisector of its  $\gamma$  angle which entails a decrease in both  $\gamma_2$  and  $\gamma_3$  angles as illustrated by

Figure 4(a) and an increase of  $\gamma_1$ , thus  $d\gamma_1/dt > 0$ . After some time,  $\gamma_1$  reaches  $\pi$  and simultaneously  $\gamma_2 = 0$  and  $\gamma_3 = 0$ , thus every robot stops the execution of its own algorithm. Thus for this case the algorithm is successful since it converges to the alignment of the three robots. When  $\pi/3 < \gamma_1 < \pi/2$ , either  $D_1$  moves in the direction of  $D_2$  entailing a decrease of  $\gamma_3$ , or it moves in the direction of  $D_3$  entailing a decrease of  $\gamma_2$ . Without loss of generality let us suppose that  $D_1$  is moving in the direction of  $D_3$ . When  $D_1$  moves in the direction of  $D_3$ ,  $d\gamma_2/dt < 0$  and, as  $d\gamma_3/dt = 0$  then  $d\gamma_1/dt > 0$ . In addition,  $\gamma_3 < \pi/3$ , thus after some time  $\gamma_1 > \pi/2$  leading to the previous situation, as illustrated by Figure 4(b). As a consequence, for scenario 1, whatever the initial value of  $\gamma_1$ , given that  $\gamma_3 \leq \gamma_2 < \pi/3 < \gamma_1$ , the execution of the algorithm on every robot leads to the alignment of the three robots. ■

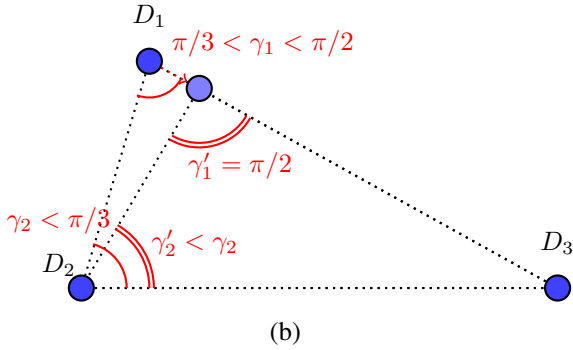
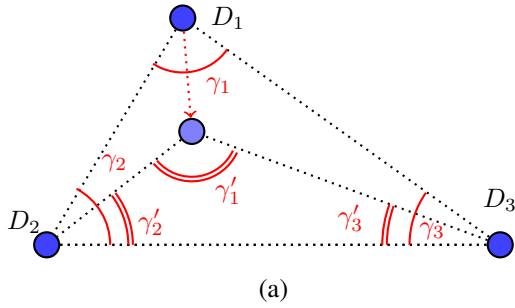


Fig. 4. Scenario 1. Case (a)  $\gamma_1 > \pi/2 > \pi/3 > \gamma_2 > \gamma_3$  and case (b)  $\pi/2 > \gamma_1 > \pi/3 > \gamma_2 > \gamma_3$

**Lemma 5:** When  $\gamma_2 < \gamma_1$ , Scenarii 3, 4, 5 and 6 lead to Scenario 1.

*Proof:* Given the assumptions that robots are starting to move at the same moment and that they are moving at the same speed, the evolution of Scenario 3 is illustrated by Figure 5. In this scenario,  $\gamma_1 > \pi/2$  thus  $D_1$  is moving in the direction of the bisector of his  $\gamma$  angle, while  $\pi/3 < \gamma_2 < \pi/2$  and  $D_2$  is moving in the direction of the initial position of  $D_1$ . As  $D_3$  is motionless, as soon as  $D_1$  and  $D_2$  are moving,  $d\gamma_1/dt > 0$  and  $d\gamma_2/dt < 0$ , leading  $D_2$  to change its state from MOVE to STOP. The resulting situation, only  $D_1$  is moving, corresponds to Scenario 1.

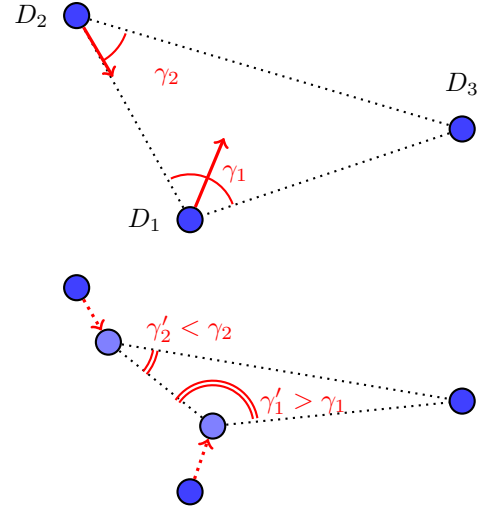


Fig. 5. Scenario 3. Initially  $\gamma_1 > \pi/2$  and  $\gamma_2 > \pi/3$ . When  $D_1$  and  $D_2$  are moving,  $\gamma_1$  increases while  $\gamma_2$  decreases, then according to the algorithm  $D_2$  stops, leading to a situation corresponding to Scenario 1.

The evolution of Scenario 4 is described in Figure 6. Initially  $\pi/3 < \gamma_2 < \gamma_1 < \pi/2$ .  $D_1$  and  $D_2$  are moving in the opposite direction, but as  $D_3$  is not moving, both  $d\gamma_1/dt > 0$  and  $d\gamma_2/dt > 0$ . After some time,  $\gamma_1$ , which is greater than  $\gamma_2$  reaches  $\pi/2$ . At that point, the direction of  $D_1$  changes to the bisector of  $\gamma_1$ , a situation corresponding to Scenario 3 which evolution leads to Scenario 1 as it was previously proved.

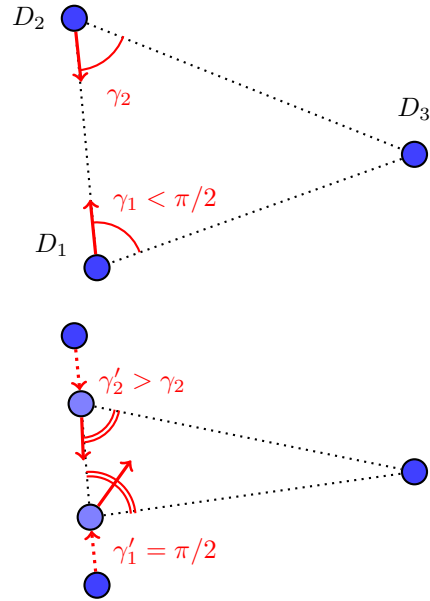


Fig. 6. Scenario 4. Initially  $\gamma_1 > \pi/3$  and  $\gamma_2 > \pi/3$  and during the movement  $d\gamma_1/dt > 0$  and  $d\gamma_2/dt > 0$ . After some time,  $\gamma_1 > \pi/2$  leading to a situation corresponding to Scenario 3.

The evolution of Scenario 5 is illustrated by Figure 7. The robot moving in the direction of the motionless robot ( $D_3$ ) has its  $\gamma$  value increasing. On the contrary, the other robot has its

$\gamma$  value decreasing entailing a change in its state from MOVE to STOP. The situation corresponds then to Scenario 1, where only one robot is moving.

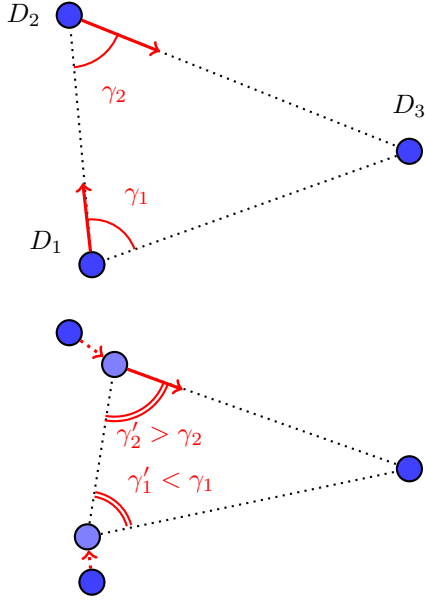


Fig. 7. Scenario 5. Initially  $\pi/3 < \gamma_1 < \pi/2$  and  $\pi/3 < \gamma_2 < \pi/2$ . As soon as  $D_1$  and  $D_2$  are moving,  $\gamma_2$  increases and  $\gamma_1$  decreases, leading to Scenario 1, a situation where only one robot is moving.

The evolution of Scenario 6 is illustrated by Figure 8. Both robots are moving in the direction of the motionless robot ( $D_3$ ). As  $\gamma_1 > \gamma_2$ , the distance between  $D_1$  and  $D_3$  is lower than the distance between  $D_2$  and  $D_3$ , then, as soon as they are moving,  $\gamma_1$  increases while  $\gamma_2$  decreases leading to a change in the state of  $D_2$  from MOVE to STOP. From that moment, only one robot is moving,  $D_1$ , a situation corresponding to Scenario 1.

Thus, under the assumption that  $\gamma_1 > \gamma_2$ , Scenarii 3, 4, 5 and 6 lead to Scenario 1 which leads to the alignment of the three robots. ■

The last Scenario which has not been considered yet is Scenario 2. When both  $D_1$  and  $D_2$  are moving, as illustrated on Figure 9, it may happen that both  $d\gamma_1/dt > 0$  and  $d\gamma_2/dt > 0$ , leading, potentially, to a situation where the two robots are positioned at the same point. In that case and if collisions are not considered, the situation is comparable to that described in Lemma 3 and the algorithm is successful. However, if collisions are considered, as two robots may be positioned at the same point, the algorithm may fail. A very detailed and careful analysis would probably help finding subcases for which this Scenario might be successful even for the collision case, however we want to find a lower bound for the probability of Algorithm 1 to be successful in that case, we thus consider that, for this scenario, the algorithm fails to align the three robots when collisions are taken into consideration.

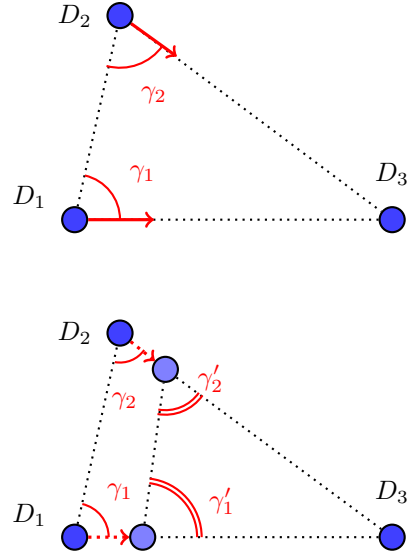


Fig. 8. Scenario 6. Initially  $\pi/3 < \gamma_2 < \gamma_1 < \pi/2$ . As soon as  $D_1$  and  $D_2$  are moving,  $\gamma_2$  decreases and  $\gamma_1$  increases, leading to Scenario 1, a situation where only one robot is moving.

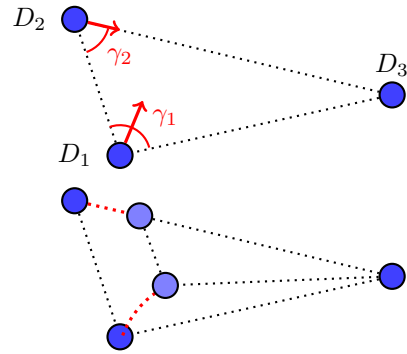


Fig. 9. Scenario 2. Initially  $\gamma_1 > \pi/2$  and  $\gamma_2 > \pi/3$ . When  $D_1$  and  $D_2$  are moving, both  $\gamma_1$  and  $\gamma_2$  may increase and the evolution may lead to a collision.

**Lemma 6:** The probability that Scenario 2 occurs is less than  $\frac{1}{36}$

*Proof:* Given three non aligned points, there exist only one circle that contains these three points. Without loss of generality we can consider that: the circle is the unitary circle, the point with the lowest  $\gamma$  value is  $D_3$  the point with the largest  $\gamma$  is  $D_1$  and that  $D_3$  is located at  $(1,0)$ . First note that as  $\gamma_1 > \pi/2$ , the three points have to be located on the same half-circle and that  $D_1$  is located between  $D_2$  and  $D_3$ . In addition, as  $\gamma_3 < \pi/3 < \gamma_2 < \pi/2 < \gamma_1$  then  $\gamma_3 < \pi/6$ . One such situation is illustrated on Figure 10.

We note also that the position of  $D_2$  is constrained by its  $\gamma$  angle that should be greater than  $\pi/3$ . Thus  $D_2$  has to be positioned on the unitary circle between  $2\pi/3$  and  $4\pi/3$  as illustrated on Figure 11, otherwise  $D_1$  cannot be positioned on the unitary circle between  $D_2$  and  $D_3$  without violating the constraint  $\gamma_2 > \pi/3$ .

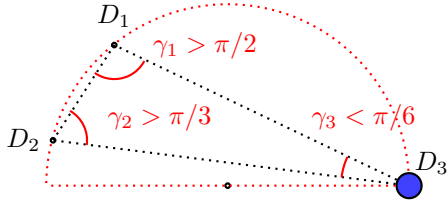


Fig. 10. Illustration of a situation where all the constraints are met:  $\gamma_3 < \pi/6$ ,  $\pi/3 < \gamma_2 < \pi/2$  and  $\pi/2 \leq \gamma_1$ .

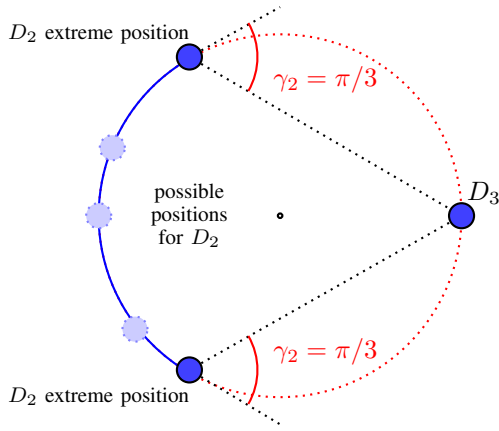


Fig. 11. According to the constraint  $\pi/3 < \gamma_2 < \pi/2$ , the possible positions for  $D_2$  are within the blue sector.

Then, whatever the position of  $D_2$  within that sector, the maximum sector in which  $D_1$  can be positioned is  $\pi/3$  as illustrated by Figure 12. This sector is maximum when  $D_2$  is located in  $(-1, 0)$  and reduces as soon as the position of  $D_2$  gets closer to its extreme possible positions.

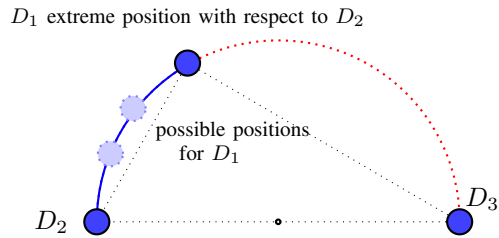


Fig. 12. In order to respect the constraints:  $\pi/3 < \gamma_2 < \pi/2 \leq \gamma_1$ , the possible positions of  $D_1$  are within the blue sector.

As a consequence, the probability of scenario 2 is the probability of having a triangle respecting the constraints on the angles:  $\gamma_3 < \pi/6$ ,  $\pi/3 < \gamma_2 < \pi/2$  and  $\gamma_1 \geq \pi/2$ , times the probability that  $D_2$  chooses the direction of  $D_3$  for moving. The first probability is equal to the probability of  $D_2$  to be positioned in the  $2\pi/3$  sector, times the probability of  $D_1$  to be positioned in the  $\pi/3$  sector starting at  $D_2$ . Thus  $\text{Proba}(\text{scenario2}) = \frac{2\pi/3}{2\pi} \times \frac{\pi/3}{2\pi} \times \frac{1}{2} = \frac{1}{36}$ . ■

We are now considering the theoretical probability for the algorithm to fail according to the initial distribution of robots

in the 3D space. First note that the execution of Algorithm 2 on every robot leads to a configuration for which all the robots are located at the same altitude, so on the same plane. In the sequel we thus restrict our analysis to a 2D space.

---

#### Algorithm 2 Aligning robots at the same altitude

---

```

1: procedure 3DTo2D()
2:   state  $\leftarrow$  GO_DOWN
3:   while state  $\neq$  STOP do
4:     if (robot detection) then
5:       state  $\leftarrow$  STOP
6:     else
7:       if ground is reached then
8:         state  $\leftarrow$  GO_UP

```

---

**Theorem 1:** When collisions are not considered, Alignment Algorithm aligns the three robots, reduced as points, iff the initial configuration is not an equilateral triangle.

*Proof:* Straightforward considering all the Lemmas. ■

**Theorem 2:** When collisions are considered, the probability for Alignment Algorithm to align three randomly distributed robots in a 3D space is greater than 0.97.

*Proof:* According to Lemmas 5 and 4, Algorithm 1 fails to align the three robots when (i) the triangle is equilateral (Lemma 1), (ii)  $\pi/3 < \gamma_2 = \gamma_1 < \pi/2$  (Lemma 3) and (iii) when  $\pi/3 < \gamma_2 < \pi/2 \leq \gamma_1$  that corresponds to Scenario 2.

A triangle is made of three points and the choice of the two first points has no impact on the properties of the triangle, the third point only has to be considered.

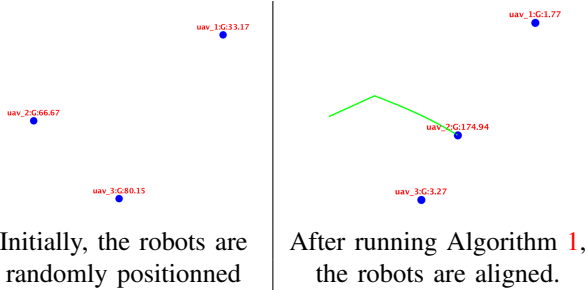
If we consider a plane (2D space), there are only two possible positions for the last point to form an equilateral triangle, thus the probability that three random points form an equilateral triangle is 0. So the probability that the algorithm fails because of case (i) is 0.

The second case corresponds to a subset of isosceles triangles. Given two points A and B in a 2D space, there are many different possibilities for the last point to form an isosceles triangle. Any point positioned on the bisector of the segment [AB] leads to an isosceles triangle as well as any point positioned on the circles which radius is equal to the length of the segment and the center is either A or B. However, both the bisector and the two circles are one-dimensional geometric objects. In comparison with the plane which is a two-dimensional object, the probability that the point C belong either to the line or to one of the two circles is again 0. So the probability that the algorithm fails because of case (ii) is 0 again. Remains the last case (iii) (Scenario 2 in Figure 3). According to Lemma 6, the probability to this scenario to occur is less than  $\frac{1}{36}$ . As a consequence, the probability that Algorithm 1 successfully aligns three randomly positioned robots in a 3D space is greater than or equal to  $1 - \frac{1}{36} \approx 0.97$ . ■

## IV. SIMULATIONS

### A. Simulation

For the simulations the environment is a square space of size  $L \times L$ . The three robots are randomly positioned within that space. The first simulations that were performed considered the theoretical framework for which a robot is a point and collisions are not considered. Without surprise Algorithm 1 was always successful. However, as we plan to use this algorithm for real experiments on our fleet of UAVs (built upon F450 DJI frames), we introduce what we call a *security distance*  $d_{\text{sec}}$ . During the simulation if the distance between any two robots is lower than  $d_{\text{sec}}$  the algorithm is supposed unsuccessful. Note that this security distance is equivalent to consider that robots are no more reduced to points but have a non-zero size and that we do not allow two robots to be too close to each other. Initially, the three robots are randomly positioned in that space but a minimum distance  $d_{\text{sec}} \times 2$  should be verified for the instance to be valid. The robots are supposed to be aligned as soon as either  $\gamma < \pi/30$  or  $\pi - \gamma < \pi/30$ . The simulator displays the initial position of robots and the path they follow during the execution of the algorithm as illustrated below.



The simulation results presented were obtained for almost 1 million runs. We have measured the frequency of each scenario and the percentage of unsuccessful runs.

### B. Results

For the runs, we have considered different environment sizes and different values for the security distances:  $L \in \{300, 500, 1000\}$  and  $d_{\text{sec}} \in \{5, 10, 15\}$ .

The results presented in Table I are stemmed from the aggregation of all the runs for all combinations of parameters (with the notable exception of the case for which  $d_{\text{sec}} = 0$ ).

Scenario	1	2	3	4	5	6
Percentage of occurrences	75.78	3.6	3.59	4.25	8.5	4.28
Percentage of failures	0	6	0	24	0	0

TABLE I

FOR EACH SCENARIO: PERCENTAGE OF OCCURRENCES AND PERCENTAGE OF UNSUCCESSFUL RUNS.

The measures reported on the second line is the percentage of occurrences of each scenario according to the random initial

positions of robots. Scenario 1 is far more frequent than the other scenarii and it is the only scenario for which only one robot is moving.

The third line of Table I reports the percentage of unsuccessful runs for a given scenario while the values of failures for every combination of parameters are presented in Table II. These results call for remarks. We first note that there is a noticeable difference between what was expected from the theoretical analysis and the results obtained from the simulation. Indeed, the analysis shows that Algorithm 1 should mainly fail for Scenario 2. From the simulation it appears that only 6% of the runs corresponding to Scenario 2 are unsuccessful, which represents a little bit more than  $6\% \times 3.6\% \approx 0.2\%$  of the total number of unsuccessful runs. The main source of failures comes from Scenario 4, for which two robots are moving in their respective directions. Theoretically, since the probability of having an isosceles triangle is 0, a collision should not occur and the contribution of Scenario 4 to the total number of failures should be 0. However, in the simulation, because we want to use this algorithm for real experiments, we introduced a security distance that has to be respected between any two robots. And in the case of Scenario 4, if the difference of the  $\gamma$  angles of the two moving robots is small, after some steps the security distance is no more respected, thus a collision risk is supposed to be high, and the run is declared unsuccessful. This explain the 24% of unsuccessful runs for Scenario 4. In more detail we have 33.8% of unsuccessful runs for Scenario 4 when  $d_{\text{sec}} = 15$ , 24.5% for  $d_{\text{sec}} = 10$  and 19.2% for  $d_{\text{sec}} = 5$ .

In Table II the percentages of successful runs for Algorithm 1 for each combination of the parameters are reported. As one hundred percent of unsuccessful runs are due to a collision risk, it is not surprising that the larger the environment and the lower the security distance, the better the results obtained by the algorithm.

$d_{\text{sec}}/L$	300	500	1000
0	100	100	100
5	99.0	99.4	99.7
10	97.9	98.8	99.4
15	96.9	98.3	99.2

TABLE II

PERCENTAGE OF SUCCESS FOR ALGORITHM 1 ACCORDING TO THE DIFFERENT COMBINATION OF THE PARAMETERS. EACH CELL OF THE TABLE IS THE RESULT OF 100000 RUNS

These results have to be compared to the theoretical value of 97% computed in the previous Section when robots are reduced to points and show that the approach is relevant even for non-zero size robots. It is noticeable that for small environments and large security distances the percentage of unsuccessful runs can be a little bit greater than the theoretical value, but, as for the results of Table I, the main source of failures is Scenario 4 and comes from the introduction of a security distance between any two robots that has to be respected during the simulation.



## V. CONCLUSION AND PERSPECTIVE

In this work we have presented a novel simple decentralized communication and localization-free algorithm for the alignment of three robots. Assuming that each robot is a point and is equipped with a 360 degrees camera, we suppose that it can simultaneously, measure the angle it forms with the two other robots, computes the direction of its movement and moves. According to its  $\gamma$  angle and to its derivative (positive or null/negative), each robot, independently of the others, decides of its behavior: move or stop. The theoretical analysis of the algorithm shows that this algorithm is successful as soon as collisions are not taken into account. If not, the probability of success is greater than 0.97.

For the simulations a security distance was introduced, which is equivalent to consider that robots have a non-zero size. The motivation for introducing such a value is that we plan to implement this algorithm on our fleet of UAVs for performing real experiments. During any run if the distance between any two robots is lower than that security distance the run is considered unsuccessful. Despite this new constraint, the success of the algorithm, based on the simulation results, is between 96.9% and 99.7%. The larger the environment and the lower the security distance, the better the results obtained by Algorithm 1.

The perspectives of this work are manifold. Current simulations were run using GraphStream, a dynamic graph library [6]. We are currently implementing the algorithm on Gazebo coupled with ROS [1], [2]. Such a simulator is better able to take into account more realistic constraints and is a step towards its implementation on a real platform.

The second direction is precisely the implementation of the algorithm on our fleet of UAVs (built upon F450 DJI frames).



The main drawback of the method is that it cannot be easily extended to a number of robots greater than three, since it relies mainly on the properties of triangles for making the decision. Different possibilities for addressing this limitation are envisioned. The first currently investigated consists for each robot to choose, based on some particular angle properties, two other robots in its neighborhood, and then to follow Algorithm 1, until the alignment. The method is iteratively applied, on another couple of chosen robots, until the alignment of all the robots is reached. The second would be to assume the availability of an additional piece of information: the relative distance between robots, an information of the type: further, closer, similar distance.

## REFERENCES

- [1] Gazebo: <http://gazebosim.org>, last visit: 04/2019.
- [2] Ros: <http://www.ros.org>, last visit: 04/2019.

- [3] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, Oct 1999.
- [4] T. Balch and R. C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.
- [5] Shantanu Das, Paola Flocchini, Nicola Santoro, and Masafumi Yamashita. Forming sequences of geometric patterns with oblivious mobile robots. *Distributed Computing*, 28(2):131–145, Apr 2015.
- [6] Antoine Dutot, Frédéric Guinand, Damien Olivier, and Yoann Pigné. Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. In Aziz Alaoui and Cyrille Bertelle, editors, *Proceedings of Emergent Properties in Natural and Artificial Complex Systems. Satellite Conference within the 4th European Conference on Complex Systems (ECCS'2007)*, October 4-5. Dresden, Germany, pages 63–72, 2007.
- [7] B. Fidan, V. Gazi, S. Zhai, N. Cen, and E. Karataş. Single-view distance-estimation-based formation control of robotic swarms. *IEEE Transactions on Industrial Electronics*, 60(12):5781–5791, Dec 2013.
- [8] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Gathering of asynchronous oblivious robots with limited visibility. In *STACS 2001*, pages 247–258. Springer Berlin Heidelberg, 2001.
- [9] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theoretical Computer Science*, 407(1-3):412–447, November 2008.
- [10] Antonio Franchi, Giuseppe Oriolo, and Paolo Stegagno. Mutual localization in a multi-robot system with anonymous relative position measures. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, October 2009.
- [11] Thomas Sousselier, Johann Dreo, and Marc Sevaux. Line formation algorithm in a swarm of reactive robots constrained by underwater environment. *Expert Systems with Applications*, 42(12):5117–5127, July 2015.
- [12] Kazuo Sugihara and Ichiro Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotic Systems*, 13(3):127–139, 1996.
- [13] Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, January 1999.