CRANFIELD UNIVERSITY

OLATZ DÍAZ PÉREZ

VISUAL CUE BASED VEHICLE TO VEHICLE COOPERATIVE POSITIONING

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Autonomous Vehicle Dynamics and Control

MSc
Academic Year: 2017–2018

Supervisor: Dr Juhyeon Hong and Dr Hyo-Sang Shin
August 2018

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING
Autonomous Vehicle Dynamics and Control

MSc

Academic Year: 2017–2018

OLATZ DÍAZ PÉREZ

Visual Cue Based Vehicle to Vehicle Cooperative Positioning

Supervisor: Dr Juhyeon Hong and Dr Hyo-Sang Shin
August 2018

This thesis is submitted in partial fulfilment of the requirements for the degree of MSc.

Abstract

Formation flight helps multi-agents cooperate visually and accomplish missions effectively. But in order to achieve a good formation shape, the agents forming the swarm must have good inter-communication among them. Currently, the main way of communication between vehicles is done by Radio Frequency, but due to its various drawbacks, the use of RF wants to be limited. Therefore, another way of communication is proposed: the visual cue based communication.

In this project, a set of autonomous vehicles will be forced to adopt a certain shape that is related to the received visual cue based marker, that is, a lead vehicle will show a marker to the followers and these will have to perform the shape that is related to the received marker. Therefore, first, a marker detection algorithm is developed which is able of detecting markers. Then, depending on the marker that has been identified, one or another formation will be executed using a potential function based approach. The approach has some visual constraints included in order to adapt it to the real scenario and only relative distances and angles between vehicles will be employed in the potential functions.

The whole thesis has been developed in a simulation environment in Matlab and Python. The results show that besides the visual constraints included, the agents are able to position themselves in a formation with equal inter-agent distance.

Keywords

UAV; Visual Cue based; Formation Flight; Potential Field; Collision Avoidance; Marker detection.

Contents

Abstract v

Contents vii

List of Figures ix

List of Tables xi

List of Abbreviations xiii

Acknowledgements xv

1 Introduction 1
- 1.1 Background and motivation . . . 1
- 1.2 Aims and objectives . . . 3
- 1.3 Project Definition . . . 6

2 Literature Review 7
- 2.1 Vision-based navigation . . . 7
- 2.2 Marker detection . . . 8
- 2.3 Formation flight . . . 10
- 2.4 Formation Control Techniques . . . 11
- 2.5 Relative distance between UAVs: Depth estimation . . . 13

3 Visual cue based positioning and formation flight system 15
- 3.1 Marker detection . . . 15
  - 3.1.1 ArUco markers and dictionaries . . . 16
  - 3.1.2 Marker creation . . . 18
  - 3.1.3 Marker detection . . . 20
- 3.2 Depth estimation . . . 24
- 3.3 Formation flight . . . 27
  - 3.3.1 Location of the UAVs in a 2D space . . . 28
  - 3.3.2 Definition of the formation shape . . . 28
  - 3.3.3 Potential function definition . . . 29
    - Definition of $\vec{P}_i^f$ for each UAV . . . 29
    - Calculation of relative distances . . . 30
    - Definition of the attractive and repulsive potential fields . . . 31

3.3.4 Visual based constraints 34
First constraint 34
Second constraint 36
Third constraint 36
Unwanted situations in formation generation 39
3.3.5 Method for escaping from a local minimum 40
3.3.6 Update of the positions 43
3.3.7 Equal inter-agent distance 45
3.4 Integration between the Marker Detection and the Formation shape 48

4 Results Analysis **51**
4.1 Marker detection 51
4.2 Formation shape 55
4.2.1 Formation shape with no constraints 55
4.2.2 Variation of the attractive and repulsive forces 59
4.2.3 Effect of the visual based constraints 63
4.2.4 Equal inter-agent distance 65

5 Conclusions **73**
5.1 Future Work 76

A Extra Data **79**

Bibliography **83**

List of Figures

1.1 Unmanned Aerial Vehicles in formation flight 2
1.2 Example of a marker and its associated formation shape 4

2.1 Examples of fiducial markers proposed in other words 8
2.2 Leader-follower approach for Formation Flight 11

3.1 ArUco marker's characteristics 17
3.2 Structure of the marker 17
3.3 Aruco markers used in the project 19
3.4 Markers candidates detection 21
3.5 Marker identification 23
3.6 Location of the 4 corners of the marker in the camera frame 24
3.7 Calculation of the horizontal distance of the two top corners in meters . . 25
3.8 Scenario representation of marker(object) detection 26
3.9 Initial positioning of a set of N UAVs 28
3.10 Definition of $\vec{P}_i^f$ for a certain $\vec{P}_i$ 30
3.11 Examples of $\vec{P}_i^f$ for each $\vec{P}_i$ 30
3.12 Example of the forces applied to agent located in $\vec{P}_i$ due to other agents . . 34
3.13 Normal distribution 35
3.14 Field Of View Comparison 37
3.15 FOV situation of two UAVs 38
3.16 Local minimum problem 39
3.17 Local minimum problem: forces 40
3.18 Escaping from a local minimum 41
3.19 Escaping from a local minimum 42
3.20 Re-arranging points $\vec{P}_i^f$ for equal number of $\vec{P}_i^f$ at each edge 47
3.21 Block diagram of the whole algorithm 49

4.1 Detection of the markers employed 52
4.2 Principles and applications of a LIDAR 53
4.3 Absolute error in Depth estimation 54
4.4 Standard deviation in Depth estimation 54
4.5 Circular formation shape with no constraints 56
4.6 Diamond-shape formation with no constraints 57
4.7 Circumference-shape formation with no constraints with an initial random distribution of UAVs 59

4.8 Variation of the total attractive force with time (UAVs starting from random positions) 60
4.9 Variation of the total attractive force with time (enlarged) 61
4.10 Variation of the total attractive force with time (Initial random distribution) 62
4.11 Variation of the total attractive force with time (Initial uniform distribution) 62
4.12 Circular formation shape with constraints 64
4.13 Diamond-shape formation with constraints 65
4.14 Equal inter-agent distance in a circular shape 67
4.15 Equal inter-agent distance in a diamond shape 68
4.16 Equal inter-agent distance in a diamond shape 70

A.1 Square shape formation 80
A.2 Triangular shape formation 81

List of Tables

3.1 Formation shape related to each ID 19
3.2 Camera parameters of the web-cam model employed 26
3.3 Remarks to be taken into account 26

4.1 Camera parameters of the web-cam model employed 51

List of Abbreviations

CNN	Convolutional Neural Networks
CORSE	Center Only Relative State Estimation
DSLR	Digital Single-Lens Reflex
EKF	Extended Kalman Filter
FOV	Field of View
GPS	Global Positioning System
IMUs	Inertial Measurement Unit
ISR	Intelligence, Surveillance, and Reconnaissance
LED	Light-Emitting Diode
LIDAR	Laser Imaging Detection and Ranging
RF	Radio Frequency
SARSE	Subtended Angle Relative State Estimation
SATM	School of Aerospace, Technology and Manufacturing
SEEA	School of Energy, Environment and Agrifoods
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
TCP	Transmission Control Protocol

Acknowledgements

First, I would like to thank my academic supervisors Dr Juhyeon Hong and Dr Hyo-Sang Shin who have given to me suggestions and best advice all along the thesis.

I am also very grateful to all my family, specially to my parents and my sister Itxaso, who have always been supporting me, not only during this year but also during my whole academic trajectory and personal life. My friends have also shown to me that no matter how far away we are from each other our friendship is so strong in order to feel that they are next to me.

Finally, I want to thank to all the new friends and people that have entered in my life during these last year. Thanks to all of them I feel that this year has probably been the best one of my life, with so many incredible experiences. To end, I will like to quote a small sentence which is my life philosophy and that has perfectly been applied during this year.

"Live every moment,
Laugh every day,
Love beyond words"

Thank you to all of you.

Chapter 1

Introduction

1.1 Background and motivation

In the last century, Unmanned Aerial Vehicles (UAVs) have gained a big importance due to the multiple functions they are able to perform as a result of the set of innovations achieved in this field of interest. In fact, in the last decades many researches have been carried on processing the information provided by a single UAV. However, due to the cost reduction of those vehicles, interconnection of multiple UAVs has been possible, obtaining an adaptive and autonomous acting system.

A system formed by multiple UAVs has a set of advantages. On the one hand, it is able to solve more tasks and also to reduce the time of executing those tasks while the quality of the collected data is increased. On the other hand, if the task is carried out in an unknown or difficult environment, a swarm of UAVs makes easier the navigation and the system will be more robust.

Currently, Unmanned Aerial Vehicles are often used in civil and military situations that are dangerous for humans operators. In order the swarm of UAVs overcome efficiently the mission it has been designed for, there are two main challenges that must be faced. On the one hand, the vehicles must achieve an adequate and efficient swarm formation, in fact, formation keeping helps multiple agents to cooperate visually and ac-

complish missions effectively. Moreover, formation flight includes fuel saving, improved efficiency in air traffic control and cooperative task allocation. On the other hand, and in order to accomplish the first challenge, the UAVs must have a good inter-communication.

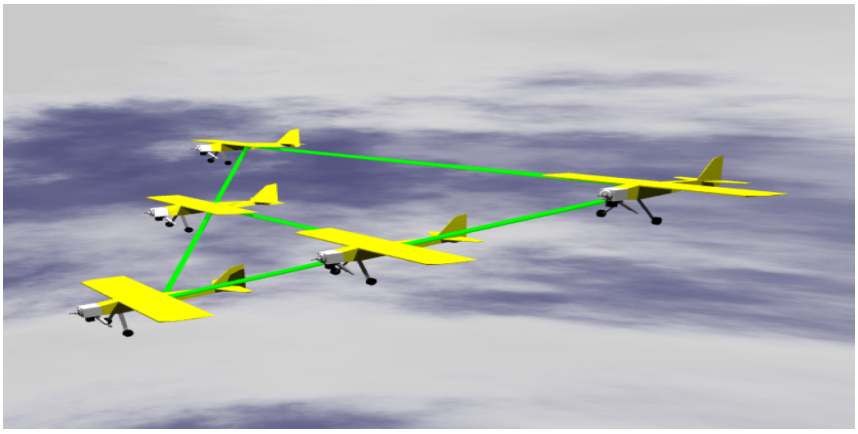


Figure 1.1: Unmanned Aerial Vehicles in formation flight [1]

Currently, much of the communication between vehicles is achieved by Radio Frequency (RF) communication, but this method has some drawbacks. First, from the military point of view, anything that is an active emitter (even though it is a small radar) can be picked up from hundreds of miles away by other Forces. Secondly, to use RF communication in a swarm, a large number of RF channels are required and the bandwidth has its limitations.

Therefore, when working with a big number of agents in a swarm, the RF becomes really complicated. So it is being looked for how the use of RF could be limited by proposing another way of communication: the visual cue based communication.

[1] Image obtained from http://web.me.iastate.edu/sbhattac/group/cyberphysicalsystems.htm

1.2 Aims and objectives

In general terms, the aim of the project is:

"*To design a squad of autonomous vehicles that can position themselves in order to achieve a shape formation between the vehicles.*"

As it has already been mentioned, for a swarm of UAVs to overcome a mission in an efficient manner, the agents forming the swarm must adopt and maintain a certain shape formation among them. In addition, depending on the type of mission the UAVs are carrying out, there will be an appropriate and most suitable formation that the vehicles must keep.

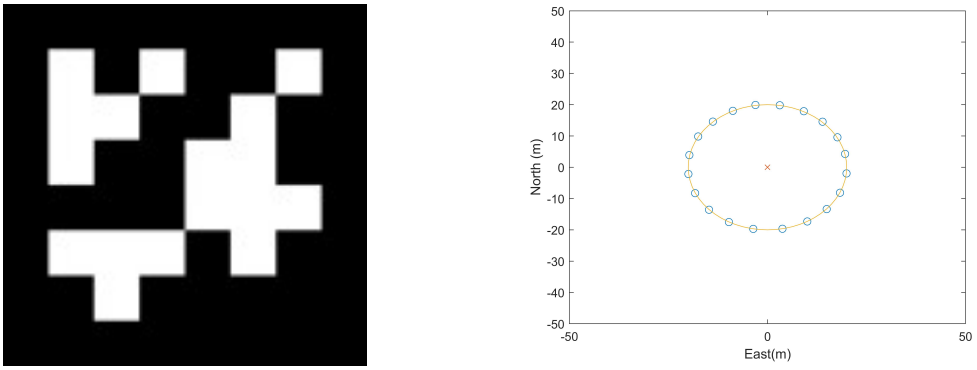
In this project, the vehicles will be forced to adopt a certain shape received by a vision cue based marker. It is going to be assumed that there will be a lead vehicle that is the one showing the marker to the others, in order the rest of them adopt the shape while the leader waits for them. Nevertheless, in this work it is not going to be analyzed the environment looking for the best formation shape for the current mission, that is, as everything is being carried out in a simulation scenario, it is being assumed that the most suitable formation shape is given my the author of the project.

Once the main aim is defined, a set of objectives that must be reached during the thesis are listed and explained below:

1. *"To develop of a marker detection algorithm (Image processing)"*. Each marker will be associated to a certain shape formation, therefore, the program must be able to detect and recognize the shown marker due to its pattern. For this step, the author will act as the lead vehicle, that is, it is going to be the one that decides which marker to show at each moment to the computer camera. It is developed in Python.

2. *"To develop a vehicle's shape formation and keeping algorithm"*. A potential field function is designed which depending on the input, that is, the marker detected, the potential field forces the agents to follow the desired formation shape. An example

is shown in Figure 1.2. It will be a combination of range based and angle based potential field function. In addition, equal inter-agent distance wants to be achieved. This algorithm has been developed in Matlab.



(a) (b)

Figure 1.2: Example of a marker and its associated formation shape

3. *"To guarantee the collision avoidance among vehicles"*. This objective is related to the previous one, as the potential field function has been designed to make sure that vehicles do not collide among them.

4. Integration between the marker detection and the shape formation algorithms. Some data is sent from Python to Matlab.

In addition, some remarks need to be done for the developing of the shape formation algorithm. As it is assumed that there is no RF communication between the vehicles, but only 'visual' communication, at a certain moment each vehicle will not know the other agents' absolute position, it is unknown data. Nevertheless, the relative distance and relative angle between vehicles is known if and only if that vehicle is inside the viewer's field of view. This is possible because from the image processing step, the relative distance parameter can be calculated. Therefore, the above mentioned constraint is considered in the project.

To end, and even though in this project there has not been time to implement the algorithm in a hardware demonstration, an additional code has been designed and validated

which calculates the relative distance from a marker to the camera lens. Therefore, in a hardware simulation, each agent would be equipped with a marker of a known size, and the other vehicles would be able to calculate their relative distance to it.

1.3 Project Definition

The project has been divided into the following sections:

1. **Literature review (Chapter 2):** In this chapter previous research works related to the topic are presented. It is the first and the most important step carried out due to the novelty the topic was for the student.

2. **Visual cue based positioning and formation flight system (Chapter 3):** Taking into account all the information gathered from the literature review, the methodology that has been decided to follow is presented.

3. **Results analysis (Chapter 4):** The results obtained from the simulation are presented in this chapter, together with the validation of the developed algorithms.

4. **Conclusions (Chapter 5):** Once the results are analyzed, the conclusions obtained from the thesis are presented in this section. Future work is also proposed in this chapter. The idea is that in future academic years, other students will follow with this research topic, so the direction that it would be preferable that they follow will be commented.

Chapter 2

Literature Review

2.1 Vision-based navigation

Currently most aerial navigation systems use multi-sensor fusion to estimate accurately the vehicle and the surrounding vehicles' states: position, velocity or acceleration. These sensors usually include GPS (Global Positioning System) and Inertial Measurement Units (IMUs), however, they can be considered complex and expensive navigation systems which are not the most appropiate alternative for small and disposable UAVs. Instead, simpler and less expensive navigation systems which mainly use data from a single camera have been developed.

From the data obtained from the camera, navigation information will be generated (which will always be relative information to other vehicles) and with it, guidance and control laws are designed as in (Johnson et al., 2007). In these study two approaches are proposed that rely on visual input for airborne navigation in leader-follower operations or formation flying: Center Only Relative State Estimation (CORSE) and Subtended Angle Relative State Estimation (SARSE). In these methods, a single camera is employed to obtain noisy measurements of a target's position in the image plane. Once the target's horizontal and vertical position is captured, an Extended Kalman Filter (EKF) is used to estimate the relative position and velocity of the aircraft relative to other vehicles or to a

target.

2.2 Marker detection

For many applications where a specific and accurate localization in the environment is required, camera pose estimation is a usual problem. This is very common in augmented reality applications or robotics as occurs in (Azuma, n.d.). These happens because in order to obtain the camera pose from camera images, it is necessary to know the correspondences between some known points located in the environment and their corresponding camera projections. Whereas there are many approaches where the image is analyzed looking for natural characteristics as in (Wagner et al., 2010) or in (Lowe, 1999), others are based in fiducial markers due to their facility to be detected which make possible higher velocity and precision in the detection process.

But first, it must be explained what it is exactly a fiducial marker. A fiducial marker system is formed by a group of markers and an algorithm that makes possible their detection in images or in a camera frame. In the following Figure 2.1 different types of markers are shown.

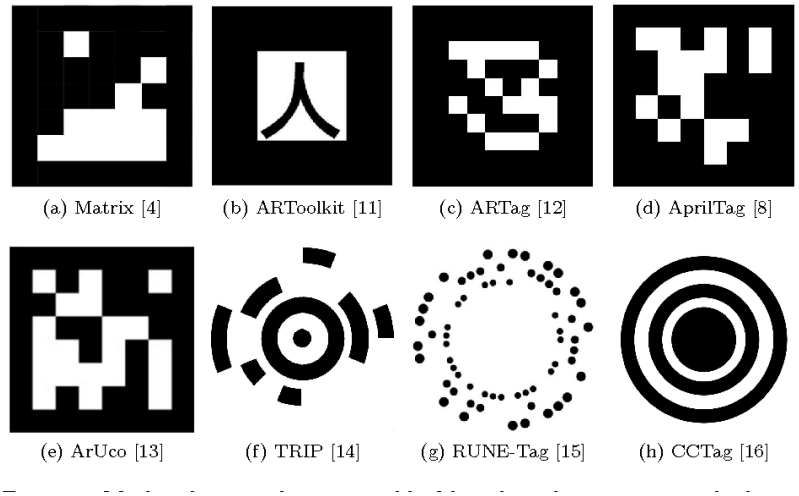


Figure 2.1: Examples of fiducial markers proposed in other words [1]

[1] Image obtained from (Benligiray et al., 2017)

Some proposals use points such as LEDs (Light-Emitting Diode) or retroreflective spheres as fiducial markers as in (Dorfmüller and Wirth, 1998) or (Ribo et al., 2001). Others employ circular markers where the identification is encoded in circular areas (Naimark and Foxlin, 2002), and there are other many different types of them. But among them, square markers have gained popularity due to the advantages they provide. On the one hand, from their four corners which are 'prominent points ', the camera pose can be easily obtained. On the other hand, the inner region is analyzed and used to identify the marker. In order to analyze the inner region, various techniques exist. The two main approaches are: using a binary code or using an arbitrary pattern as an image.

In the arbitrary pattern approach, the most popular system is probably ARToolKit (Kato and Billinghurst, 1999) and has been widely employed in many academic researches during last decades. This system consists on a set of markers composed by a wide black border and an inner image which is kept in a database of valid patterns. But this system has two main drawbacks. On the one hand, as a template matching technique is used for markers identification, many false positives are obtained, or even many confusions between different markers. On the other hand, a fixed global threshold is used for square detection and as a consequence, the sensitivity to lighting variation is quite high.

The binary pattern approach is the most popular technique among square fiducial markers. The first proposal was Matrix (Rekimoto, 1998) and after it other systems as ARTag (Fiala, 2010), ARToolKit Plus (Wagner and Schmalstieg, n.d.) or BinARyID (Flohr and Fischer, n.d.).

Finally, in ArUco (Garrido-Jurado et al., 2014), a square-based fiducial marker system with binary codes is proposed. There, apart from using a set of predefined markers, a method for generating markers dictionaries is proposed, where the size and the number of markers that form the dictionary can be configured by the user depending on the ones that the user requires. The advantages that this technique has comparing to the previous ones are that the algorithm produces markers with a criterion of maximizing inter-marker

distance and the number of bit transitions. Moreover, an error detection and correction method is proposed, which allows a greater number of erroneous bits error correction.

2.3 Formation flight

Formation flying is the discipline where two or more aircraft are flying under the command of a flight leader. It has been a research challenge that has been deeply investigated in the 21st century with aircraft and spacecraft systems.

Formation flight has many advantages for the Unmanned Aerial Vehicles that form the formation. These advantages are not only from an operational point of view, but also in terms of energy-saving. From the operational aspect, Military Forces have focus their attention on the idea of using a large number of 'cheap' UAVs acting like a swarm. This alternative reduces costs due to the fact that using a big number of small UAVs is less expensive than employing an advanced tactical aircraft. Moreover, as a swarm of vehicles is being employed, larger areas can be covered by the system when carrying out an ISR (Intelligence, Surveillance, and Reconnaissance) mission or the agents can be rapidly deployed from the operational basis where they are located. In addition, the system of UAVs in flight formation is robust to jamming and other threats.

Other beneficial aspect is the fuel consumption reduction because of aerodynamics effects. In nature, this effect can be noticed in birds formations. For instance, V-shape formation flying is energy-efficient, and its application into aircrafts has been investigated in many studies as in (Vachon et al., 2002), (Koo and Shahruz, 2001) and (Ning et al., 2011). Behind the wingtips, wing vortexes are generated and the trailing aircraft reuses the wake turbulence's energy in order to generate its corresponding dynamic forces, and as a consequence, the induced drag will be harshly reduced. Big reduction of the induced drag have been obtained, up to 40% in V-shaped flight formations. In addition, drag reduction will make possible to reduce fuel-consumption because less thrust will be needed for the required mission.

2.4 Formation Control Techniques

There are several techniques in order to achieve the desired formation shape among a given number of UAVs. Most of the approaches are based on the leader-follower principle and on the geometrical approach. This technique is based on first choosing one of the agents forming the swarm as the virtual leader or choosing a virtual point for the same purpose. Then, the states of the rest of the UAVs (position, velocity...) will be controlled relative to the virtual leader as in (Li and Liu, 2008) or in (Shi and Yang, 2013), and a guidance law is applied to the virtual leader.

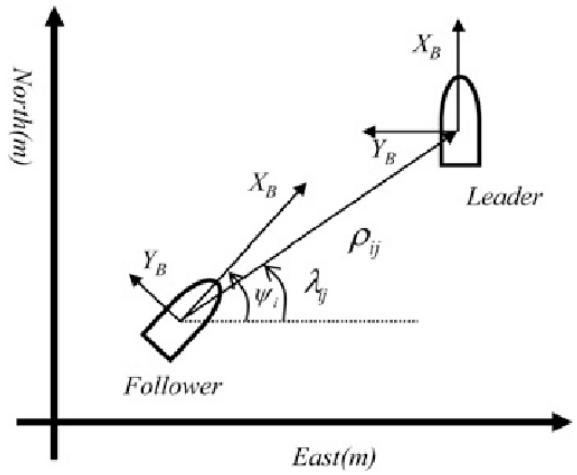


Figure 2.2: Leader-follower approach for Formation Flight [2]

Linear Model Predictive Control was used in (Iskandarani et al., 2014) to control a formation flight formed by three UAVs. In this study, the future outputs were predicted in order to optimize the control inputs. Researches from West Virginia University have also developed studies in this area and they have designed some control laws which are based on Non-Linear Dynamic Inversion. Then, they have done their validation testing the method in a formation of three fixed-wing UAVs as explained in (Campa et al., 2007) and (Gu et al., 2006). The proposed method, under some predefined assumptions, turns into a linear system a certain system of higher order, in order to apply and control the UAVs by a simpler linear technique.

[2]Image obtained from (Peng et al., 2011)

The major part of the researches carried out using leader-follower approach were performed using a small number of agents, which rarely exceed five agents, and a finite number of formation shapes. Nevertheless, the number of agents taking part in the formation can be increased if a cascade-type architecture is designed. This was studied in (No et al., 2011), and it consists on the idea that the leader-follower functions are transferred from one vehicle to another one, that is, among the followers one of them will be chosen as the leader for the followers behind him. However, this type of approach that is based on the leader-follower behaviour has a big drawback which is the weakness to a leader behaviour, because if for any reason the leader fails and there is no any plan to solve the issue, the whole swarm will be affected and this implies the mission's failure.

Another approach that has been quite used in guidance and navigation researches is to use potential fields. In the researches explained in (Paul et al., 2008) there is a combination between a virtual leader formation approach with an extended version of the potential field solution that was presented in previous studies as (Do, 2006). In this combined approach, the potential field that will determine the guidance command for the UAVs composing the formation is composed by four components. First, a virtual leader and virtual formation positions are set up in order to know where each UAV will have to position itself. The first potential field component is pointing towards the virtual position that must adopt each UAV in the formation. The second component is the one that regulates the distances between the platforms and the third one is for collision avoidance among the vehicles. To end, there is one component for obstacle avoidance. As it is taken for granted, the last two components are repulsive potential fields, in order the agents do not approximate to a certain point. The approach mentioned has the advantage that it is not required a discrete decision making technique as the ones that are normally used in Sense and Avoid methods, that is, when using a potential field approach, a continuous control and a natural motion of the UAVs is assured for obstacle and collision avoidance.

In (Dang and Horn, 2014*b*) and (Dang and Horn, 2014*a*), a potential field approach is

also used, where as in the previous studies, there is potential field formed by several components or contributions in order to reach a goal and for avoiding collisions. Nevertheless, no virtual leader is needed in this approach. Formation merging, control, splitting, collision avoidance with static obstacles and target tracking guidance ideas are explained in those studies where the vehicles do not perform a rigid pre-defined formation. Instead, the formation evolves due to the application of the vector field and the agents' positions to an equilibrium position of the vehicles with the encircling field and environment. Therefore, performing a structured formation is not so relevant in these studies as in others.

To end, researches carried out and explained in (Jung and Kim, 2014), show a potential-function based formation shape control scheme for swarm simulation. There, the function is designed to avoid collision among agents, approach a destination, and achieve a certain shape formation around the destination. Each UAV that forms the swarm, moves to the formation position which is generated from a reference point and the neighboring agents position. The results shown in that paper are for an elliptical, diamond-shaped and heart-shaped formations. Moreover, the local-minimum problem which is very usual in every potential function based approach is studied and a solution is given to it. The local minimum problem occurs in a situation where an agent is trapped between the attractive and repulsive potential fields, therefore, in this research, the vehicle escapes from a local minimum using a virtual escape point after recognizing the trapped situation.

2.5 Relative distance between UAVs: Depth estimation

From a camera frame, it can not be obtained the absolute position of other vehicles or objects in the space, but relative distance from the camera to an object or agent can be tracked. For been able to develop a potential field based approach to obtain a formation shape among the UAVs, each UAV must estimate where the others are located in order to be able to avoid it.

In nature, it is able to find some cases in which this problem is concerned. Some

flying insects, flies for instance, have been developing many types of eye designs due to the evolution and their ability of adapting to the environment. In other words, depending on the task that the insect has to perform, the insect's eyes will adopt one or another shape. In (Bitsakos and Fermüller, 2006), a study is performed analyzing the depth estimation of the eye of a dipteran fly where experiments show that for the compound eyes of dipteran flies distance can be estimated with accuracy if the viewed objects are relatively close to the eye.

Moreover, there are some other approaches to perform the depth estimation of an object. There is a method named STM for determining distances to objects and rapid autofocusing of camera system. The method described in (Subbarao and Surya, 1994) uses image defocus information for doing the estimation and is based on a new Spatial-Domain Convolution/Deconvolution Transform. Only two images that are taken with different camera parameters are required, being these parameters lens position, focal length and aperture diameter. Moreover, both images can be arbitrary blurred and there is no necessity of been a focused image. This is the main reason why STM is much fast in comparison with Depth-from-Focus methods such as (Tang et al., 2017), where lens position or focal length of best focus is searched.

In addition, there are many researches carried out with the purpose of replacing the above mentioned traditional methodologies with some new approaches based on CNN (Convolutional Neural Networks). In (Tang et al., 2017), a digital single-lens reflex (DSLR) camera is used together with a laser scanner to obtain high-resolution images and very accurate depth estimations. Furthermore, the experimental results carried out have shown the validity of the proposed technique.

Chapter 3

Visual cue based positioning and formation flight system

This chapter can be divided into three main sections. First, the procedure adopted for marker detection is going to be explained which is going to be implemented in Python. Then, a potential function-based approach for formation flight is developed and implemented in Matlab. To end, the integration between the marker detection in Python and the formation shape algorithm is carried out.

3.1 Marker detection

There are many types of markers and detection algorithms currently available. For this project, it has been decided to use ArUco Library instead of writing a whole image process recognition code. Many reasons can explain the decision taken, which will be explained now.

ArUco is an OpenSource library for detecting square fiducial markers in images and can be used in research and commercial projects for free. The library is written in C++, but it is possible to adapt it to other languages as Python, for instance. The main reason of using these type of fiducial markers for the project is the good results that can be expected

in their detection. On the one hand, from the time consuming point of view, developing a whole code from the beginning would have been very time consuming for the research student, without being this the main objective of the thesis. On the other hand, the main reason is that using a fiducial marker library and algorithm enables a very fast detection of the markers with reduced number of erroneous detections. This point is crucial at this research study, as a mistake in the detection process can have two different consequences. If no marker is detected, there will not be any input signal going into the swarm, and therefore, the agents will not perform any manoeuvre or formation. In addition, if a wrong marker was detected, the UAVs would perform an inappropriate formation shape for the mission they are carrying out, which would cause the swarm not to overcome the mission they are carrying out, or even could lead to catastrophic failure.

As explained in the literature review, each library has its own set of markers as Artoolkit +, Chilitags, AprilTags... and the one used in the project: ArUco dictionary. It is very important the design of the dictionary in order to avoid any confusions between markers as the more differences among them the better is the detection process. In ArUco there are many types of markers and each type belongs to a dictionary. Anyway, there are some common characteristics that every marker share.

3.1.1 ArUco markers and dictionaries

Each ArUco marker is characterized by some features as seen in figure 3.1:

- A vector that contains four points which are the four corners of the square fiducial marker
- An identification number (ID) that is unique for each marker of a certain dictionary
- Its size once printed in a paper (in meters)
- The translation and rotation that relates the center of the marker and the location of the camera

Moreover, in the following Figure 3.1 it can be regarded the coordinate system that is employed in the library.

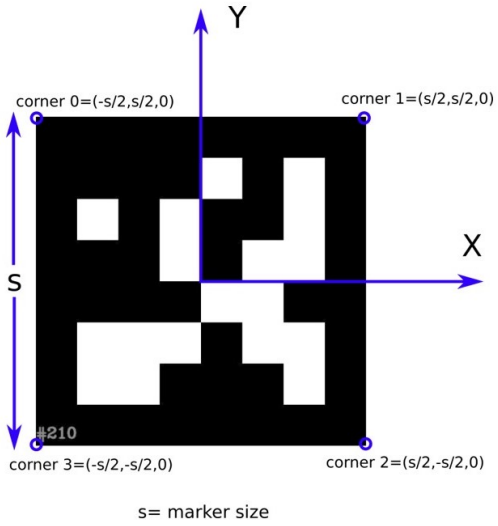


Figure 3.1: ArUco marker's characteristics

Every ArUco marker is composed by an external black border and a binary pattern inside of it. The inner pattern is unique for each marker and it is related to an ID.

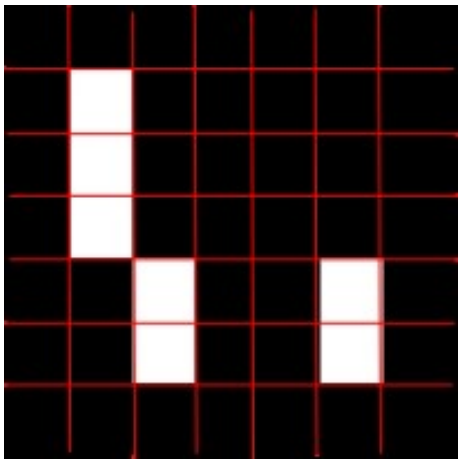


Figure 3.2: Structure of the marker

Therefore, the binary pattern is the one that identifies the corresponding marker.

Moreover, there are many dictionaries in this library. A dictionary of markers is formed by a set of markers that are going to be used for a particular application, in other

words, it is a list of binary codifications of the markers that form a certain dictionary. The main properties of the dictionary are:

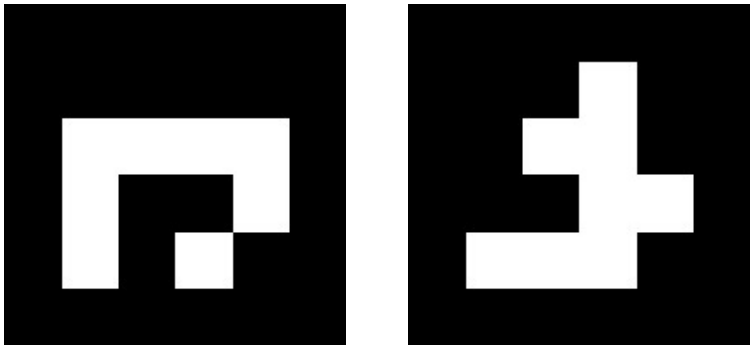
- The size of the dictionary, that is, the number of markers that are inside of it
- The marker size, that is, the number of bits that form each marker

In addition, ArUco library includes some predefined dictionaries with several dictionary sizes and marker sizes within a range. For example, the dictionary called DICT_6X6_250 is a dictionary composed by 250 different markers having all of them a size of 6x6 bits. Moreover, it must be explained that even though at a first instant it may have sense to think that the marker's identification number is obtained by converting the binary code inside the inner region to its corresponding decimal base number, this is a wrong idea. The reason for this is that when the number of bits that compound the marker is too high, the decimal base number obtained at the conversion would be very huge, and this is not a practical solution. Instead, the marker's ID is just the marker index inside the dictionary. For example, if the dictionary is just formed by 5 markers, the corresponding IDs will be: 0, 1, 2, 3 and 4.

In relation to the suitable number of bits for a dictionary, the more number of bits, the more number of words that will be in the dictionary and therefore, the probability of confusion will be smaller

3.1.2 Marker creation

Before starting with the detection process, the markers have to be printed and located in the environment. For this thesis, an already existing markers dictionary is used, in particular, the DICT_4X4_50. As explained before, this dictionary is composed by 50 different markers with a size of 4x4 bits. At the beginning, the idea of creating an own dictionary formed by markers designed by the user was considered. But when testing them, it was proved how the detection process was better with the predefined ones.



((a)) ID=11 ((b)) ID=15

((c)) ID=23 ((d)) ID=25

Figure 3.3: Aruco markers used in the project

The following markers have been chosen, where each of them will be related to a certain UAV formation.

In the following table, the desired formation related to each shape is defined.

ID	Formation shape
1	Square shape
15	Diamond shape
23	Circular shape
25	Triangular shape

Table 3.1: Formation shape related to each ID

3.1.3 Marker detection

In the real scenario it is assumed that there is a lead vehicle that shows a marker to the others in order these ones perform a certain formation around the lead vehicle which will be taken as the reference point for performing the shape.

Therefore, every vehicle forming the swarm must be equipped with a camera that enables it detecting the command of the leader, that is, the formation that the swarm must adopt to complete successfully the mission.

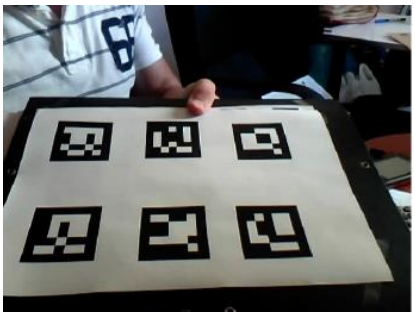
So, in Python, the following procedure has been developed to detect an image or a camera frame where an ArUco marker is visible. The objective will be to obtain:

- The position of the four corners of the marker, in the order explained in Figure 3.1
- The identification number of the detected marker

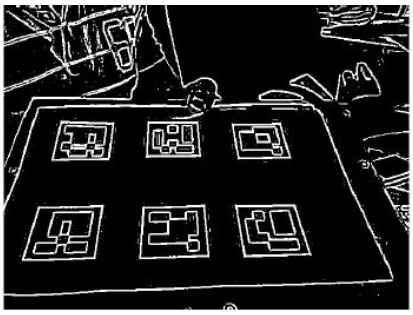
The marker detection process can be divided into two main steps:

1. **Marker candidates detection**: Detection of rectangles in the image.
    - First, the image segmentation is carried out in the original Figure 3.4 (a). It begins with applying an adaptive thresholding to segment the markers (Figure 3.4 (b)). The local adaptive thresholding approach is very robust to several lighting conditions, therefore, it is an appropriate alternative for the real scenario that is considered in the thesis. The purpose of this step is to extract the most prominent contours or borders in the gray-scale image.
    - Then, the contour extraction is performed in the already thresholded image. In ArUco algorithm, Suzuki and Abe algorithm is used, which is explained in (SUZUKI, n.d.). A set of image contours are obtained, but many of them will not be relevant for the detection process because they are not convex or they do not approximate to a square shape (Figure 3.4 (c)). So, in order to carry

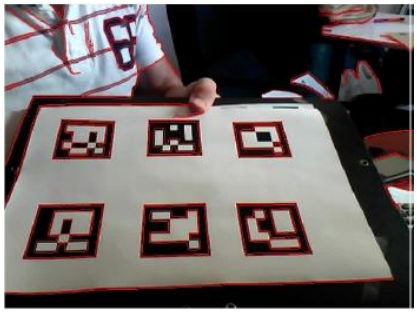
out a polygonal approximation to discard any other invalid shape, Douglas-Peucker algorithm developed in (H DOUGLAS and K PEUCKER, 1973) is applied. As ArUco markers are enclosed inside a rectangular contour, the aim will be to discard any other contour that does not approximate to a 4-vertex polygon. So, any other border with a smaller or bigger number of points is removed. In addition, other additional filters are applied, for instance, removing rectangles that are too close to each other. This step is also necessary because the adaptive threshold usually detects both the external and the internal border of the marker, and it is desired to remain with the most external one.



(a) Original image [1]



(b) Adaptive thresholding footnote[1]



(c) Contour detection footnote[1]

Figure 3.4: Markers candidates detection

[1]Image obtained from (Garrido-Jurado et al., 2014)

2. **Marker identification**: Analysis of the inner codification to know if the detected contours are markers or not.

    - A marker code extraction is performed to analyze the inner region of the detected contours. Firstly, the projection perspective is removed in order to obtain the frontal view of the detected rectangle area employing an homography matrix as in Figure 3.5 (b).
    - Then, the resulting image is thresholded. For that, Otsu's algorithm is used, which gives the optimal threshold value if the image distribution is bimodal.
    - Later, the binarized image is divided into different cells as in Figure 3.5 (b), where the regular grid form depends on the marker size and the border size. For example, if the marker is divided into a 5x5 grid, only the inner 4x4 grids will contain useful information to identify the marker, because the outside black border is equal for every marker.
    - At each cell, the amount of white and black pixels are counted, to check if it is a white or a black bit located at that cell. In other words, each element of the cell is analyzed and a value 0 or 1 is assigned to it depending on the major number of pixels of that colour. 0 corresponds to a black bit and 1 to a white one. (Figure 3.5 (c))
    - At this step, a rejection process is also carried out. Because if no presence of outer black border is found, that is, if all the bits in the border are not 0, the following step will not be executed.
    - To end, the inner bits are analyzed to check if the detected marker structure is part of a dictionary or not. This step is also complex as four different identifiers must be tested, one for each possible rotation of the marker in the environment, as the markers may be rotated. Nevertheless, it can be assumed that for out thesis application, the UAVs will not be upside down. So, when compering each of them with the data of a dictionary, if any one coincides, the

marker is given as valid and its identification number is obtained.

- Other error correction techniques are also applied when it is required, that is, when no match is obtained between the detected marker and the dictionary.

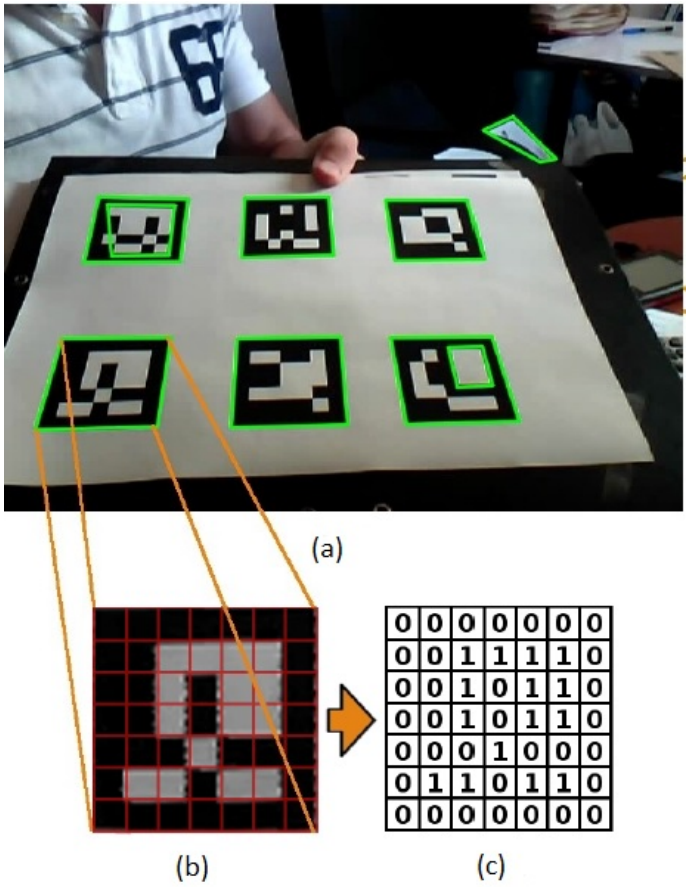


Figure 3.5: Marker identification [2]
(a) Polygonal approximation and removal of undesired contours
(b) Perspective transformation
(c) Bit assignment

Therefore, the marker detection procedure is already explained which is written in Python. Later, the performance of this algorithm is going to be shown.

[2]Image obtained from (Garrido-Jurado et al., 2014)

3.2 Depth estimation

In the real scenario that is being considered, it is important for each vehicle to know their relative distance to the others, in order to be able to avoid them without collision. Nevertheless, later some constraints are going to be developed related to this idea because the camera that the UAVs have incorporated will have a maximum detection range. Therefore, the following assumptions are being made in order the UAVs are able to detect the relative distance from other agents to them.

- Each vehicle has incorporated an ArUco marker of a known size
- Every vehicle is flying at the same altitude, or at least with minimal height variations
- Every vehicle has a camera incorporated of known specifications and can detect ArUco markers and the positions of its corners

The following procedure is followed to calculate the depth from a marker to a camera frame:

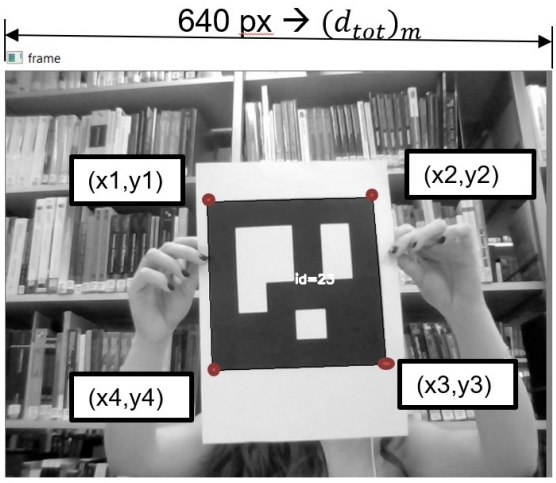


Figure 3.6: Location of the 4 corners of the marker in the camera frame

The camera has a known resolution of 640x480 pixels, and the positions of the 4 corners of the image (x_1,y_1),(x_2,y_2),(x_3,y_3) and (x_4,y_4) for a reference coordinate system that has its origin on the left top corner of the camera frame are known (see Figure 3.6).

First, the horizontal distance between the two top corners is obtained with Equation 3.1. For that, it is enough with knowing the positions in pixels of the two top corners (x_1,y_1) and (x_2,y_2).

$$d_x(\text{px}) = x_2(\text{px}) - x_1(\text{px}) \tag{3.1}$$

Also the inclination of the detected marker is calculated as follows:

$$\text{angle inclination}(\alpha) = \arctan\left(\frac{y_2(\text{px}) - y_1(\text{px})}{x_2(\text{px}) - x_1(\text{px})}\right) \tag{3.2}$$

Once the inclination and the horizontal distance in pixels are both known, the horizontal width of the marker in the camera frame in meters is obtained in Equation 3.3 applying some basic trigonometry as seen in Figure 3.7 where the marker's width l in meters is known:

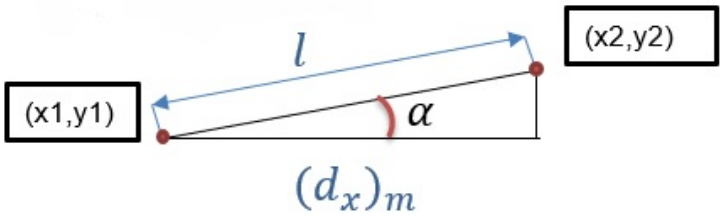


Figure 3.7: Calculation of the horizontal distance of the two top corners in meters

$$d_x(\text{m}) = l(\text{m}) \cdot \cos\alpha \tag{3.3}$$

For the next step some basic characteristics of the camera are considered. In order to perform the thesis, as the laptop web-cam has been employed for the simulations, its parameters are taken for the calculations. In the following Table 3.2 the required parameters are summarized:

Parameter	Value	Units	Description
–	640×480	pixel	Camera resolution
FOV	60	deg	Field of view

Table 3.2: Camera parameters of the web-cam model employed

In addition, the scenario assumed to perform the relative distance or depth calculation is described in 3.8:

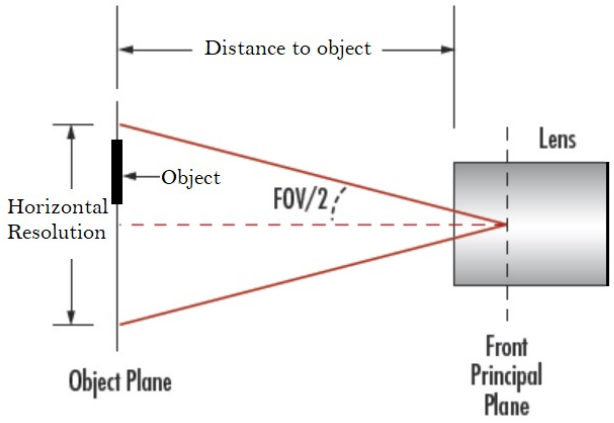


Figure 3.8: Scenario representation of marker(object) detection

So, after having performed the inclination correction and knowing the following data:

Parameter	Value	Units
Horizontal resolution	640	Pixels
Marker width	18	cm

Table 3.3: Remarks to be taken into account

First, a simple law of three is performed to convert the horizontal resolution from pixels to meters, by applying:

$$d_{tot}(\text{m}) = \frac{\text{horizontal resolution (px)} \cdot d_x(m)}{d_x(\text{px})} \tag{3.4}$$

Where d_{tot}(m) is the total horizontal distance in meters of the object plane. So, in order to finally estimate the relative distance from the camera frame to the known marker:

$$\text{distance to marker (m)} = \frac{d_x(\text{m})}{2 \cdot \tan(\frac{\text{FOV}}{2})} \tag{3.5}$$

3.3 Formation flight

In the last decades, much attention has been focused on the formation control of swarms of UAVs, being a swarm a big number of autonomous agents. Formation control may be performed in several environments such as air, water and ground, but for the current work they are taken as aerial vehicles. Nevertheless, the methodology and the corresponding results are carried out at a two-dimensional space, because it is assumed that the height is fixed for every UAV, that is, that all of them are flying at the same horizontal plane.

Formation control with a large number of agents involves deciding the position that each vehicle must adopt to maintain a designated formation but it is also crucial avoiding collisions with neighboring agents at any moment. So, in order to achieve the already mentioned goal, a potential function based approach is employed. The alternative proposed can be adapted to a set of different scenarios and missions, that is, depending on the mission the UAVs are carrying out, a suitable shape formation must be adopted by the agents. Therefore, the idea is to develop an algorithm which allows that with little modifications on the code, several shapes can be performed.

In addition, in this work some constraints will be applied to the potential function based approach in order to adapt it to the real scenario. These constraints will be:

- Limited field of view of the camera that each UAV has incorporated, that is, each agent will only know the relative distance to the one some of the other vehicles are

located (the ones that are inside its Field of View(FOV)).

- Limitations on the relative distance estimation. Depending on the marker size and the camera's focal length among others, marker detections from bigger or smaller distances will be possible to perform.
- Accuracy in the relative distance estimation.

3.3.1 Location of the UAVs in a 2D space

First, a set of N agents will be located in a uniform distribution in the plane. The number of agents can be adjusted by the user.

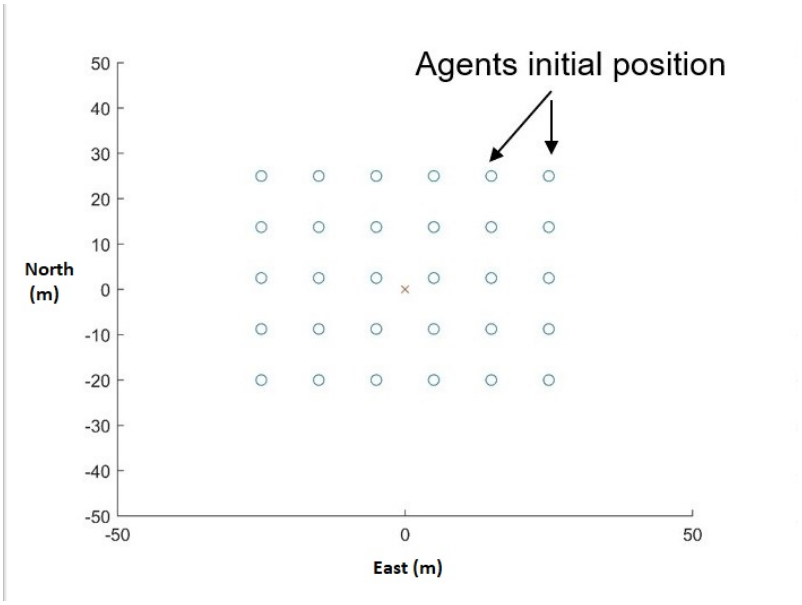


Figure 3.9: Initial positioning of a set of N UAVs

3.3.2 Definition of the formation shape

Then, several formation shapes are going the be defined by a formation line. For that, an absolute reference point is set. It is necessary to set up a reference point in order to be able to fix the formation at an exact location. The practical meaning of the reference point can be three:

- One could be that the leader is located there. This leader shows the marker to the rest of agents so that they perform a formation shape where the leader does not take part.

- Another theory could be that is is an enemy, and therefore, the swarm must enclose it.

- It could also be taken as an obstacle that has to be avoided and surrounded by the swarm.

For the project, the first hypothesis is going to be taken, that is, the leader will be located in the reference point. In addition, the agents will be forced to adopt positions over the formation line, therefore, on the formation line it must be enough space for all of them to be located above, ensuring an inter agent separation distance. Moreover, it is assumed that at every step, the followers know the position of the leader, that is, the formation shape will always be to a less distance from the formation line than the maximum depth estimation distance that agents can perform.

In this work four formation shapes have been performed: circular shape, diamond-shape, square shape and triangular shape.

3.3.3 Potential function definition

Definition of $\vec{P}_i^f$ for each UAV

The next step is to calculate the point on the formation line towards each agent has to move at each time step. Therefore, for each agent located at a certain position $\vec{P}_i$, its corresponding $\vec{P}_i^f$ is calculated, where $\vec{P}_i^f$ is a point on the formation line that guides agent $\vec{P}_i$ to the formation line. The point $\vec{P}_i^f$ is defined as the intersection point located on the formation line when $\vec{P}_i$ is connected with the reference point, as seen in Figure 3.10 and Figure 3.11.

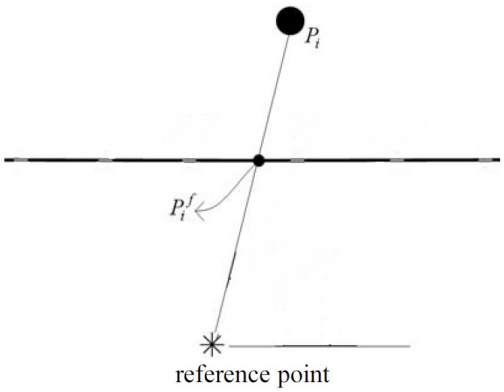


Figure 3.10: Definition of $\vec{P}_i^f$ for a certain $\vec{P}_i$

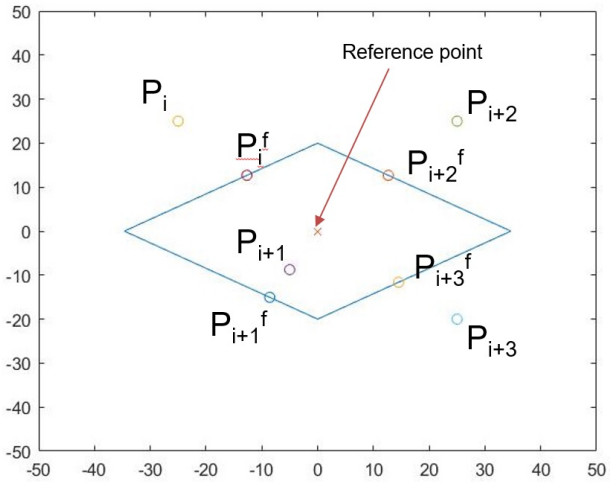


Figure 3.11: Examples of $\vec{P}_i^f$ for each $\vec{P}_i$

Calculation of relative distances

In this study it is assumed that each agent can obtain the relative distance from it to other vehicles, as well as its relative distance to the reference point. This assumption is much less restrictive than assuming that each agent knows the absolute position of the others and it is also more adaptable to the real scenario.

So, the current position of agent i is defined as $\vec{P}_i = [x_i, y_i]$ and of agent j as $\vec{P}_j = [x_j, y_j]$. Moreover, the intersection point located in the formation line when $\vec{P}_i$ is connec-

ted with the reference point, that is, $\vec{P}_i^f$, is $\vec{P}_i^f = [x_i^f, y_i^f]$.

Therefore, the distance vector between $\vec{P}_i$ and $\vec{P}_i^f$, $(\vec{\psi}_i^f)$, is calculated easily as well as the relative distance vector between agents i and j, $(\vec{\psi}_i^j)$, applying the following Equations 3.6 and 3.7:

$$\vec{\psi}_i^f = \vec{P}_i - \vec{P}_i^f \tag{3.6}$$

$$\vec{\psi}_i^j = \vec{P}_i - \vec{P}_j \tag{3.7}$$

Definition of the attractive and repulsive potential fields

In the next step, both the attractive and the repulsive potential fields are going to be set.

The potential field method is inspired in the concept of electrical charges. That is, if agents are seen as electrically-charged particles, every vehicle must have the same type of electrical charge in order the vehicles are sent away ones from the others when they get too close to each other. In addition, even though in this thesis no obstacle is being considered, obstacles must also have the same electrical charge that the UAVs not to collide with them. On the other hand, the destiny or goal point must have the opposite electrical charge so that the agents can approximate to it and replace the position.

So, each agent requires an attractive potential field U_i^a that forces it to move to the point $\vec{P}_i^f$ located in the formation shape. In addition, it needs a repulsive potential field U_i^r not to collide with other UAVs.

The attractive potential field of agent i has the following expression shown in Equation 3.8:

$$U_i^a\left(\vec{\psi}_i^f\right) = -c_a \cdot \left(1 - e^{\frac{||\vec{\psi}_i^f||^2}{la^2}}\right) \tag{3.8}$$

And the repulsive potential has the following form:

$$U_i^r(\vec{\psi}_i^f) = \sum_{j=1} \{c_r \cdot e^{-\frac{||\vec{\psi}_i^j||^2}{{l_r}^2}}\} \qquad (j \neq i) \tag{3.9}$$

Therefore, around each UAV, a potential field will be generated which will be the sum of the attractive and repulsive potential fields as it is shown in Equation 3.10:

$$U_i = U_i^a(\vec{\psi}_i^f) + U_i^r(\vec{\psi}_i^f) = -c_a \cdot \left(1 - e^{-\frac{||\vec{\psi}_i^f||^2}{{l_a}^2}}\right) + \sum_{j=1} \{c_r \cdot e^{-\frac{||\vec{\psi}_i^j||^2}{{l_r}^2}}\} \qquad (j \neq i) \tag{3.10}$$

Where c_a and c_r are constant values that regulate the strengths of their respective potential fields and they are adjusted in the algorithm in order to obtain a good performance of the UAVs when adopting a formation shape. In the following chapter it will be explained which values have been given to them. In addition, l_a and l_r are the correlation distances of the potentials as is explained in (Jung and Kim, 2014), and $||\vec{\psi}_i^f||$ and $||\vec{\psi}_i^j||$ are the modules of the vectors $\vec{\psi}_i^f$ and $\vec{\psi}_i^j$ respectively.

Therefore, the force vectors that will be applied to each agent of the swarm will be calculated by performing the negative of the gradient, hence, from the gradient definition, it is known that the force will point along the direction of steepest descent, which is also perpendicular to the lines of constant potential energy.

The corresponding force that is applied to each vehicle has the following form of Equation 3.20:

$$\vec{F}_i = -\nabla U_i = -\frac{2c_a||\vec{\psi}_i^f||}{{l_a}^2} e^{\frac{||\vec{\psi}_i^f||^2}{{l_a}^2}} \cdot \hat{\psi}_i^f + \sum_{j=1} \{\frac{2c_r||\vec{\psi}_i^j||}{{l_r}^2} e^{-\frac{||\vec{\psi}_i^j||^2}{{l_r}^2}} \cdot \hat{\psi}_i^j\} \qquad (j \neq i) \tag{3.11}$$

Where $\hat{\psi}_i^f$ and $\hat{\psi}_i^j$ are the unit vectors of $\vec{\psi}_i^f$ and $\vec{\psi}_i^j$ respectively.

The attractive $\vec{F}_i^a$ and repulsive $\vec{F}_i^r$ forces are broken down as follows in order to analyze them properly.

On the one hand, the attractive force is:

$$\vec{F}_i^{\,a} = -\frac{2c_a||\vec{\psi}_i^f||}{{l_a}^2} e^{\frac{||\vec{\psi}_i^j||^2}{{l_a}^2}} \cdot \hat{\psi}_i^f \tag{3.12}$$

The attractive force applied to each agent will be a vector pointing towards the agent's related $\vec{P}_i^f$ at that time step. Therefore, it is important to calculate the relative distance to that point $||\vec{\psi}_i^f||$, but also the direction cosines of the vector. Regarding the shape of the function, it can be said that it is a positive exponential function which depends on the distance vector between $\vec{P}_i$ and $\vec{P}_i^f$ (point located on the formation line). The attractive force will be directly proportional to $||\vec{\psi}_i^f||$ and the exponential is increased by the square of $||\vec{\psi}_i^f||$. The aim of having a positive exponential to the square of the distance is that when agents are really far away from their destiny point $\vec{P}_i^f$, the attractive force will be really strong. Whereas when they have reached their goal, $\vec{F}_i^a$ will be quite small. This is the reason why in a future step the attractive force variable will be used as the stopping criteria.

On the other hand, the repulsive force is:

$$\vec{F}_i^{\,r} = \sum_{j=1} \{\frac{2c_r||\vec{\psi}_i^j||}{{l_r}^2} e^{-\frac{||\vec{\psi}_i^j||^2}{{l_r}^2}} \cdot \hat{\psi}_i^j\} \qquad \left(j \neq i\right) \tag{3.13}$$

The repulsive force applied to agent i due to agent j will be a vector pointing to the opposite direction to the one agent j is located. In this case, the exponential function is used for avoiding collisions among agents. The aim of having a negative exponential function is that the repulsive force only affects those neighboring agents that are located within a certain range. In other words, when the distance between vehicles i and j is very big, as $||\vec{\psi}_i^j||$ is squared inside the negative exponential, the value of the exponential

will be nearly zero, without having effect that $||\vec{\psi}_i^j||$ is also multiplied before, that is, the strength of the exponential is much higher than the one of the multiplier $\frac{2c_r||\vec{\psi}_i^f||}{l_r^2}$.

In Figure 3.12, three agents are represented located in $\vec{P}_i$, $\vec{P}_{i+1}$ and $\vec{P}_{i+2}$. In addition, the attractive and repulsive forces that are applied to agent i are drawn.

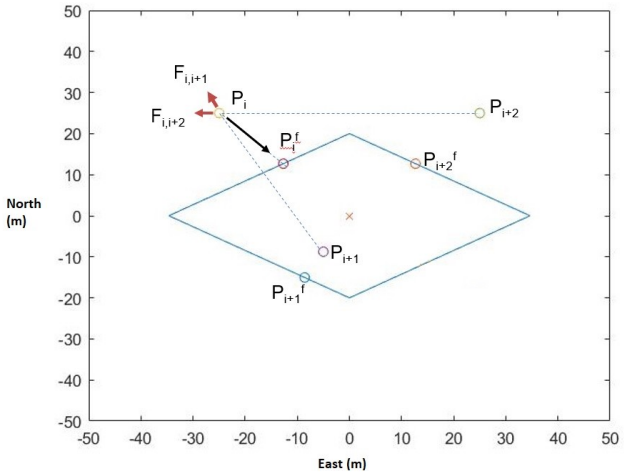


Figure 3.12: Example of the forces applied to agent located in $\vec{P}_i$ due to other agents

For stocking up every attractive and repulsive force that is applied to each agent, some matrices are built at each time step. The one of the attractive force will be a Nx1 matrix, but the repulsive matrix will have a dimension of NxN as every combination must be taken into account. Furthermore, once the module of the forces are calculated, they must be separated to calculate the force in each direction (North and East).

3.3.4 Visual based constraints

The next step will be to apply some constraints to the methodology that has been followed.

First constraint

First, the errors that may be produced when an agent i is trying to detect the distance to the marker that agent j is carrying are being taken into account. In order to do so, an error

is going to be added to the real existing distance between two vehicles, in the way shown in Equation 3.14:

$$d_i^j = ||\vec{\psi}_i^j|| + \mathcal{N}(\mu, \sigma^2) \tag{3.14}$$

Thus, the error will be assumed that follows a general normal distribution where:

- μ is the mean or expectation of the distribution
- σ is the standard deviation
- σ^2 is the variance
- $||\vec{\psi}_i^j||$ is the real distance between agent i and j
- $\mathcal{N}$ is the normal distribution, also known as the Gaussian distribution, which is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean, as shown in Figure 3.13.

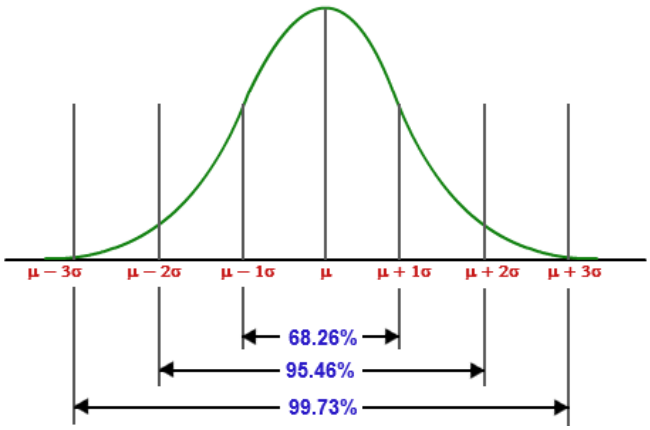


Figure 3.13: Normal distribution

For this particular case, as an error is going to be assumed in the distance measurement, which may be positive or negative, the mean of the distribution will be zero, whereas the variance will depend on the distance.

Second constraint

Then, some limitations on the relative distance estimation are taken into account, because depending on the marker size and the camera's focal length among others, marker detections from bigger or smaller distances will be possible to perform.

In the following chapter, some validations have been carried out to have an idea of the maximum distance to the one a marker could be detected with the laptop's camera-web. Taken that as a reference, and assuming that the UAVs will be flying in an open environment with good lighting conditions and with a better camera than the one of the student's laptop, markers could be approximately detected up to a distance of 25 meters.

Therefore, in the algorithm developed, if the distance between agent i located in $[x_i, y_i]$ and agent j positioned in $[x_j, y_j]$ is greater that 25 meters, the repulsive force between those two, that is, $\vec{F}_i^j$ and $\vec{F}_j^i$ will be zero.

$$if \quad \sqrt{(x_i - x_j)^2 + (x_j - y_j)^2} \geq 25 \quad then \quad \vec{F}_i^j = \vec{F}_j^i = 0 \tag{3.15}$$

Third constraint

To end, a last constraint is introduced at this step. This one is related to the field of view that each aerial vehicle has. The problem is that in the real scenario, agent i will not be able to see every other agent that integrates the swarm. Agent i will be moving in a direction, and therefore it can only see and detect those agents that are inside a field of view centered at its moving direction.

In Figure 3.14 a comparison between some cameras with different field of view is shown. Obviously, the greater the FOV, the better performance in the shape formation algorithm as the number of other agents detected by each agent will be higher. Therefore, it will be much safer because possible collisions with neighboring agents will be less.

[3]Image obtained from https://www.oemcameras.com/fov$_c$*omparison*

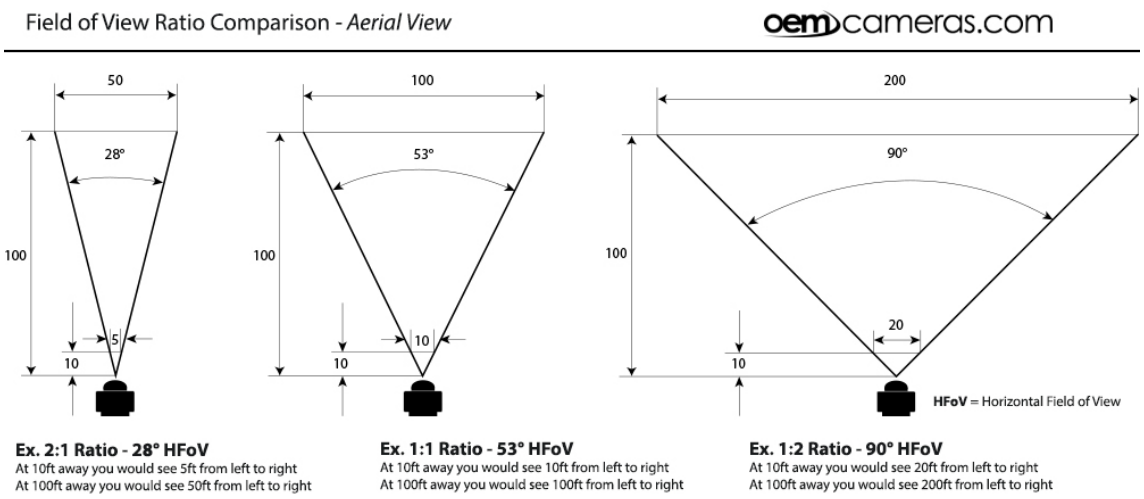


Figure 3.14: Field Of View Comparison [3]

To introduce the mentioned constraint in the algorithm, the following data is required in order to know which agents are inside the Field Of View of agent i:

- The moving direction vector of agent i at time step t_i is known and expressed as an unitary vector $\vec{v}_i = [v_{x_i}, v_{y_i}]$.

- For any other agent j, the distance vector from agent i to agent j is known and defined as $\vec{\psi}_j^i$ at time step t_{i+1}. At this point it is very important to be careful with the notation that has been used because even though vectors $\vec{\psi}_j^i$and $\vec{\psi}_i^j$ have the same magnitude, their direction cosines are not the same, they are rotated 180° respect to each other.

Therefore, to know if an agent j is inside the field of view of agent i, the following scalar multiplication is performed.

$$\vec{v}_i \cdot \vec{\psi}_j^i = ||\vec{v}_i|| \cdot ||\vec{\psi}_j^i|| \cdot \cos\alpha \tag{3.16}$$

Where $\alpha = \widehat{(\vec{v}_i, \vec{\psi}_i^j)}$

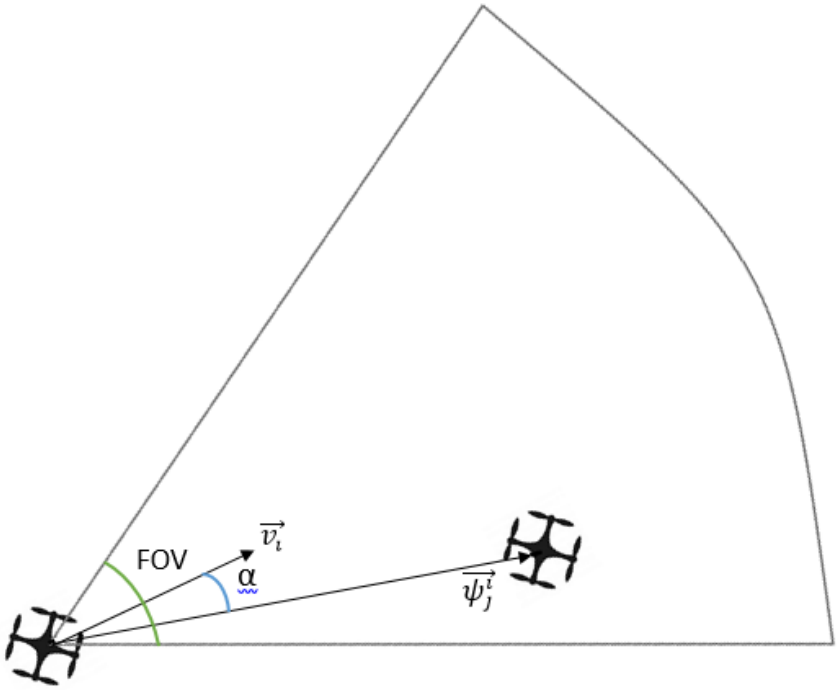


Figure 3.15: FOV situation of two UAVs

Then:

$$\cos\alpha = \frac{\vec{v_i} \cdot \vec{\psi}^i_j}{||v_i|| \cdot ||\psi^i_j||} \tag{3.17}$$

So if the camera employed has a field of view (FOV) equal to 60°, this means that the maximum α so that agent j is inside the field of view of agent i is 30°. Therefore, $\cos\alpha$ must be smaller than $\cos 30°$, that is, $\cos\alpha \leq \cos 30°$.

Consequently, as it has been done for the constraints explained before, in the case any agent j is not in the FOV of agent i, the repulsive force that is applied to agent i due to agent j, that is, $\vec{F}_{ri}^{\,j}$ will be zero, but this does not mean that $\vec{F}_{rj}^{\,i}$ will also be zero, because it will depend on the moving direction each of them is having.

Unwanted situations in formation generation

When a swarm formation wants to be achieved, sometimes agents may be placed in unwanted situations. In the following Figure 3.16, a very habitual problem that occurs when working with artificial potential fields is shown.

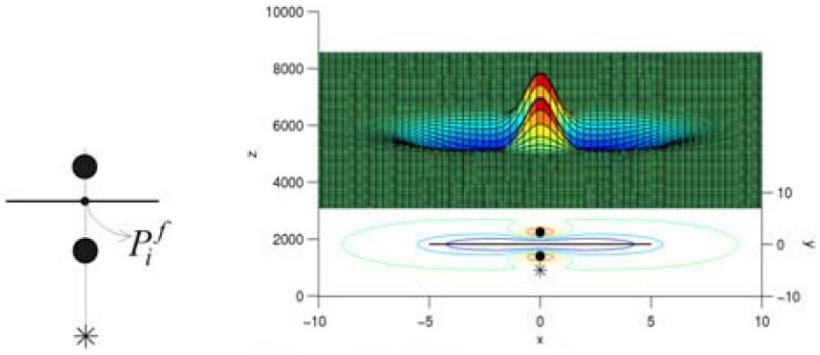


Figure 3.16: Local minimum problem [4]

In Figure 3.16 an unwanted situation is shown together with its corresponding potential distribution. In the figure, the two black dots represent two agents that are approximating the formation line. The $*$ is the reference point and the line is the formation line that must be occupied by the agents. When two agents are co-linear with the reference point, it means that both of them will have the same $\vec{P}_i^f$, and they will go approximating to that point one agent from the inside of the formation line and the other one from the outside. The problem only arises when both agents are near the formation line, because as in each time step $\vec{P}_i^f$ changes due to the combination of the attractive and repulsive potential fields, the fact that in a certain moment two agent have the same $\vec{P}_i^f$, does not mean that in the future time step they will also have it.

Agents in Figure 3.16 are stopped on a line orthogonal to the formation line due to their repulsive fields for collision avoidance.

[4] Image obtained from (Garrido-Jurado et al., 2014)

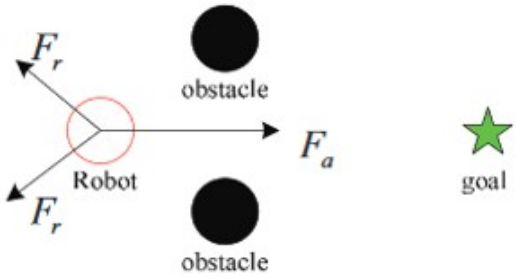


Figure 3.17: Local minimum problem: forces

The reason why they are stopped is explained in Figure 3.17 with another example. The local minimum is produced when the vector sum of all the forces acting in an agent are zero. Therefore, the agent will not have an input acceleration that makes it move to its goal and it is trapped.

3.3.5 Method for escaping from a local minimum

If no method is applied in order that the trapped agents escape from the point where they have been trapped, the UAVs that form the swarm will not be able to build the correct formation.

In general, agents will have different $\vec{P}_i^f$, but when unwanted situations occur as the one previously explained, and agents stop between attractive and repulsive potential fields, the following virtual escape method is proposed.

First, the local minimum has to be identified. The local minimum occurs when the following conditions are satisfied:

$$||\vec{\psi}_i^f|| < D_m \quad and \quad ||\vec{F}_i|| < F_m \tag{3.18}$$

Where D_m is the minimum distance between $\vec{P}_i$ and $\vec{P}_i^f$ and F_m is a minimum force of $\vec{F}_i$.

On the one hand, the first condition is used to check if an agent is arriving to its related $\vec{P}_i^f$ or not, because $||\vec{\psi}_i^f||$ is the relative distance between $\vec{P}_i$ and $\vec{P}_i^f$. As it has been

mentioned before, $\vec{P}_i^f$ keeps varying at each time step, so the local minimum problem only may appears when that distance is close to zero. On the other hand, the second condition is employed to check if the resultant of the attractive force from $\vec{P}_i^f$ and the repulsive forces for collision avoidance with neighboring agents is zero. Therefore, both D_m and F_m are chosen as constants that are nearly zero.

In the following Figure 3.18 an example of an agent escaping from a local minimum is shown.

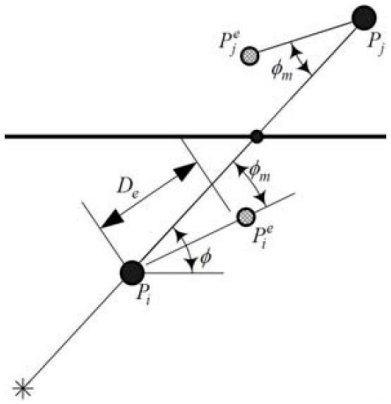


Figure 3.18: Escaping from a local minimum [5]

The agent located in $\vec{P}_i$ is trapped in a local minimum. This happens because two agents are co-linear with the reference point when they are approximating their corresponding $\vec{P}_i^f$ (which is the same for both). Then, so that the agents escape from this unwanted situation, two escape points, $\vec{P}_i^e$ and $\vec{P}_j^e$, are set, one for each agent that is located in $\vec{P}_i$ and $\vec{P}_j$ respectively. These new points are calculated as follows:

$$\vec{P}_i^e = \vec{P}_i + D_t \cdot \begin{bmatrix} \cos(\phi - \phi_m) \\ \sin(\phi - \phi_m) \end{bmatrix} \tag{3.19}$$

Where:

- D_t is set as a constant distance that is used for escaping from the trapped situation.

[5] Image obtained from (Garrido-Jurado et al., 2014)

- ϕ is the angle between an horizontal line and the line that connects the reference point with the agent's current position.

- ϕ_m is an arbitrary angle, set by the user, in order to escape from the local minimum.

ϕ_m is set as smaller that ϕ so that the agent does not diver too much from its destiny point. This angle will ensure that the agent that was trapped in the local minimum travels away from it. When the local minimum is checked using Equation 3.18, if these conditions are not satisfied, then, the agent located in $\vec{P}_i$ moves toward $\vec{P}_i^f$ on the formation line. Nevertheless, if the conditions are verified, the agent will move toward an escape point $\vec{P}_i^e$. At the end, both agents should be able to reach the formation line as in Figure 3.19.

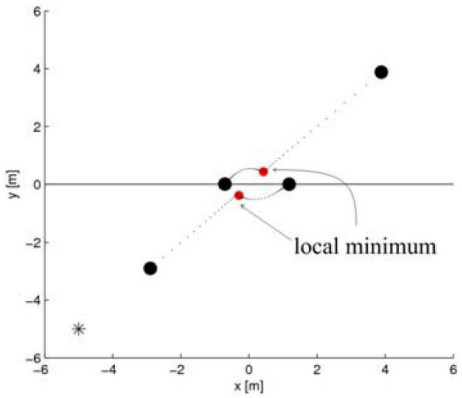


Figure 3.19: Escaping from a local minimum

To escape from the local minimum, the attractive force must be replaced by a new module and direction. Therefore, the total force applied to agent i so that it escapes from the trapped situation is:

$$\vec{F}_i = -\frac{2c_a||\vec{\psi}_i^e||}{{l_a}^2} e^{\frac{||\vec{\psi}_i^e||^2}{{l_a}^2}} \cdot \hat{\psi}_i^e + \sum_{j=1} \{\frac{2c_r||\vec{\psi}_i^j||}{{l_r}^2} e^{-\frac{||\vec{\psi}_i^j||^2}{{l_r}^2}} \cdot \hat{\psi}_i^j\} \qquad (j \neq i) \qquad (3.20)$$

Where $\vec{\psi}_i^e$ is the direction vector between agent i and $\vec{P}_i^e$, and is defined as in Equation

3.21.

$$\vec{\psi}_i^e = \vec{P}_i - \vec{P}_i^e \tag{3.21}$$

3.3.6 Update of the positions

In this section, once the forces that are applied to each agent are known, the control law for agent located in $\vec{P}_i$ is designed. Agent $\vec{P}_i$ has a mass and its dynamics can be described as follows:

$$\ddot{\vec{P}}_i = \vec{u}_i \qquad i = 1,2,3...N \tag{3.22}$$

Where u_i is the control input applied to each agent that is part of the swarm and $\vec{u}_i \in R^2$, that is, it is a two dimensional acceleration vector and N is the total number of agents.

Therefore, combining the attractive and repulsive forces calculated for each agent, the following control law is designed:

$$\vec{u}_i = a \cdot \frac{\vec{F}_i^a + \vec{F}_i^r}{||(\vec{F}_i^a + \vec{F}_i^r)||} \tag{3.23}$$

The objective is that with the potential fields approach, at each time step, the moving direction that each UAV must adopt is defined. Then, a constant a is added as a multiplier in order to obtain a constant acceleration.

From Newton's second law:

$$\vec{F} = m \cdot \frac{d^2\vec{r}}{dt^2} \tag{3.24}$$

$$\frac{d^2\vec{r}}{dt^2} = \frac{\vec{F}}{m} \tag{3.25}$$

Integrating in both sides, first the velocity is obtained:

$$\frac{d\vec{r}}{dt} = \frac{\vec{F}}{m} \cdot t + C_1 \tag{3.26}$$

Integrating again, positions are got:

$$\vec{r} = \frac{1}{2} \cdot \frac{\vec{F}}{m} \cdot t^2 + C_1 \cdot t + C_2 \tag{3.27}$$

In the project, at each time step of time dt, it is assumed that the agents start from zero velocity. This assumption is done so that the agents do not increment their velocity at each time step due to the applied input acceleration. In addition, constant acceleration is going to be set, that is, the module will always be the same, but the vector will keep varying at each time step for every UAV as appears in Equation 3.23.

Therefore, if at each step agents are supposed to start from a null velocity, then, from Equation 3.26 it is obtained that $C_1 = 0$.

Then, as the initial position of each agent at a certain time step is the final position at the previous time step:

$$\vec{r} = \vec{r_0} + \frac{1}{2} \cdot \frac{\vec{F}}{m} \cdot t^2 \tag{3.28}$$

Where, $\{r, r_0, F\} \in R^2$ and r_0 is the final position at the previous time step.

Using the notation employed in previous sections, and as $\frac{\vec{F}}{m}$ is an acceleration, then current position of each UAV will be updated using the following Equation 3.29:

$$\vec{P_i} = \vec{P}_{0i} + \frac{1}{2} \cdot \vec{u}_i \cdot t^2 \tag{3.29}$$

Where $\vec{P_i}$ is the position of agent i after having moved from its previous location $\vec{P}_{0i}$.

To end with the control law, some comments must be done. In the control law that has

been defined, it has been assumed that at each time step agents go from a zero velocity to a final one, where they are stopped until the next time step. This performance does not have any sense in a real environment because drones should move from the starting point to the goal with an approximately constant speed. Nevertheless, as in the potential fields approach forces are calculated, that is, an acceleration vector, in the simulation model that variable must be inserted. But it is not desired that agents keep increasing their velocity due to that acceleration. Therefore, in order to avoid the excessive growth of the velocity, it is simulated that agents go from a zero to a maximum speed at each time step which would represent a constant mean speed in the real scenario.

3.3.7 Equal inter-agent distance

On a formation shape, it will be much better if all the agents could maintain an equal distance among them when they are located on the formation line. The importance of this condition depends on the mission the swarm is carrying out. Therefore, in this project some ways of achieving equal inter-agent distance have been thought.

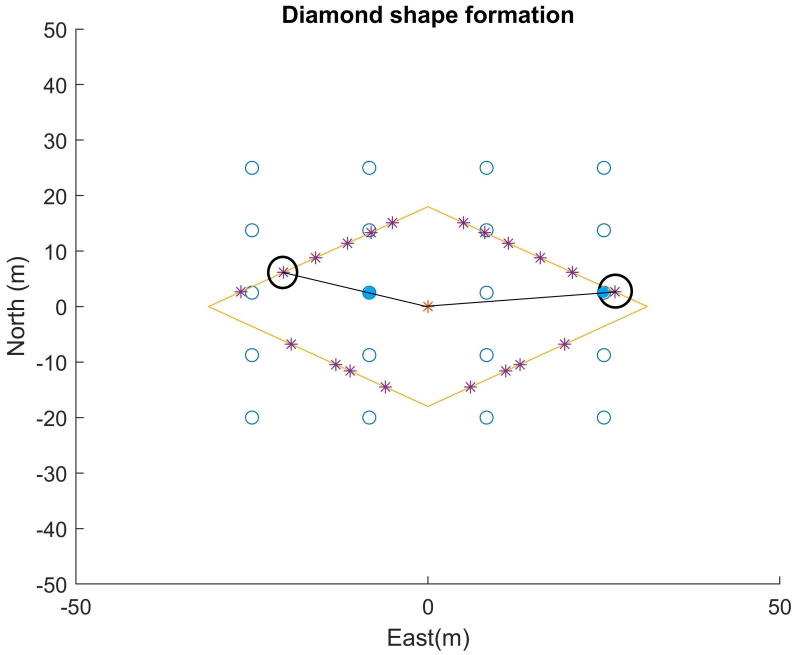
The first alternative is varying the repulsive potential field. The repulsive force can be regulated varying c_r, that is, the strength of the potential. If this parameter is increased, agents will be driven away from each other. Therefore, once every agent is on the formation line, the potential field generated around each agent will make that in zones with higher agent density, UAVs move away from each other, adopting an equilibrium position with a more uniform separation among them. This procedure can be more or less time consuming depending on the initial distribution of the agents, that is, if at the beginning they are very close to each other and the majority of the vehicles end in the same edge of a certain shape (because their corresponding points $\vec{P}_i^f$ are located in the same edge), more time will be required in order to achieve an equal inter-agent distance.

Therefore, an adaptation has been done to the previous method in order to 'send' the same number of UAVs to each edge when they start from a uniform distribution but not

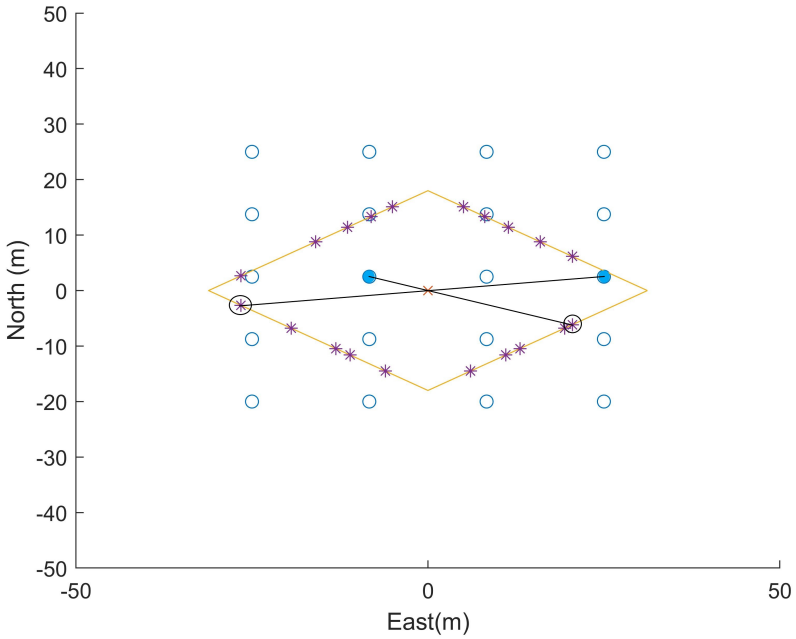
centered in the origin. Let's assume that a diamond shape wants to be obtained, which has 4 edges ($N_{edges} = 4$), with a set of N=20 agents. The idea is that when for each agent located in $\vec{P}_i$, its respective $\vec{P}_i^f$ is calculated, if at a certain edge there are already more than N/N_{edges} intersection points located, the other agents that would have their $\vec{P}_i^f$ located at that edge will have it instead in the 'opposite' edge. To understand the methodology adopted it cannot be forgot that $\vec{P}_i^f$ is defined as the intersection point located on the formation line when $\vec{P}_i$ is connected with the reference point (located in [0,0]). Therefore, it can be seen in the following Figure 3.20 how the points $\vec{P}_i^f$ of two agents are 'moved' to the opposite edges so that at each edge there are the same number of intersection points located. That is, in Figure 3.20 (a) there were more agents' $\vec{P}_i^f$ points located in the upper edges, and in Figure 3.20 (b), the same number of $\vec{P}_i^f$ points are at each edge.

When this methodology is followed, as $\vec{P}_i^f$ is the point on the formation line that guides agent $\vec{P}_i$, then those agents will have to go through the reference point where the leader is located. Therefore, an auxiliary leader avoidance term will be introduced in the potential field approach so that when the distance between the agents and the leader is smaller than 4 meters, their movement direction will be perpendicular to the distance vector that connects the leader and the agent. Following this technique, the agents will go around the leader without collision risk.

To end, a last alternative is proposed which has not been implemented in the simulation but the idea would like to be commented. The technique is based on the idea that once the agents are on the formation line, an angle based potential field is designed that makes that the relative angle between every vehicle is the same. For example, for the circular formation shape, if a swarm of 36 agents wants to be set up in a formation, the angle based potential field has to be able to 'push' the UAVs to positions separated among each of them by an angle of 10°.



(a) Time=0s



(b) Time=0s

Figure 3.20: Re-arranging points $\vec{P}_i^f$ for equal number of $\vec{P}_i^f$ at each edge

3.4 Integration between the Marker Detection and the Formation shape

Once the marker detection algorithm is designed in Python and the shape formation algorithm in Matlab, the integration between both programs has to be done.

In this case, a UDP (User Datagram Protocol) socket has been used. User Datagram Protocol is an alternative communications protocol to Transmission Control Protocol (TCP) used mainly to establish low-latency and loss-tolerating connections between applications on the internet.

In the simulation, the laptop's camera is launched with Python. When a marker is detected, the corresponding identification number of the marker is saved in a variable and transmitted to Matlab via UDP. In order to do so, first the ID is converted to a string in Python. Then, the string is sent using a UDP protocol. In Matlab, the variable is received and decoded from a string back to a number. As soon as the variable is received as a number, the formation shape algorithm begins, where agents adopt the shape that is related to the received identification number. In addition, as soon as the agents are located on the formation line with equal inter-agent distance, if a new marker is shown to the camera, the agents will adopt a new shape starting from the positions where they have ended in the previous formation.

In the next figure 3.21, a flow chart of the whole process is shown.

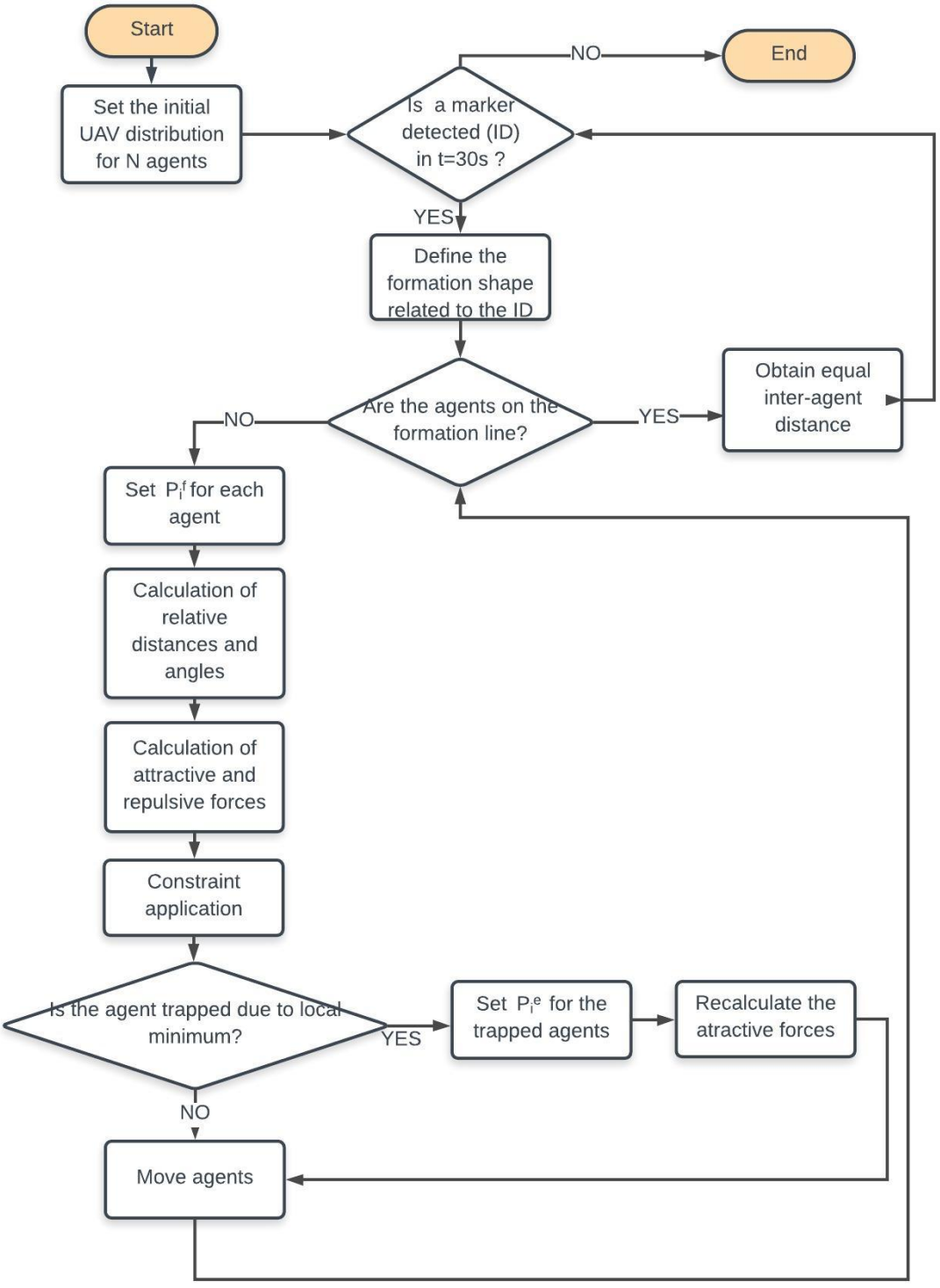


Figure 3.21: Block diagram of the whole algorithm

Chapter 4

Results Analysis

In this chapter, the procedure followed to test the algorithm designed by the student is going to be explained. The aim will be to validate its performance in order to analyze the results in the next chapter

4.1 Marker detection

At a first instance the marker detection algorithm is going to be analyzed. For doing so, the four markers that are used for defining the shape formations are printed in a paper with a marker width size of 18cm.

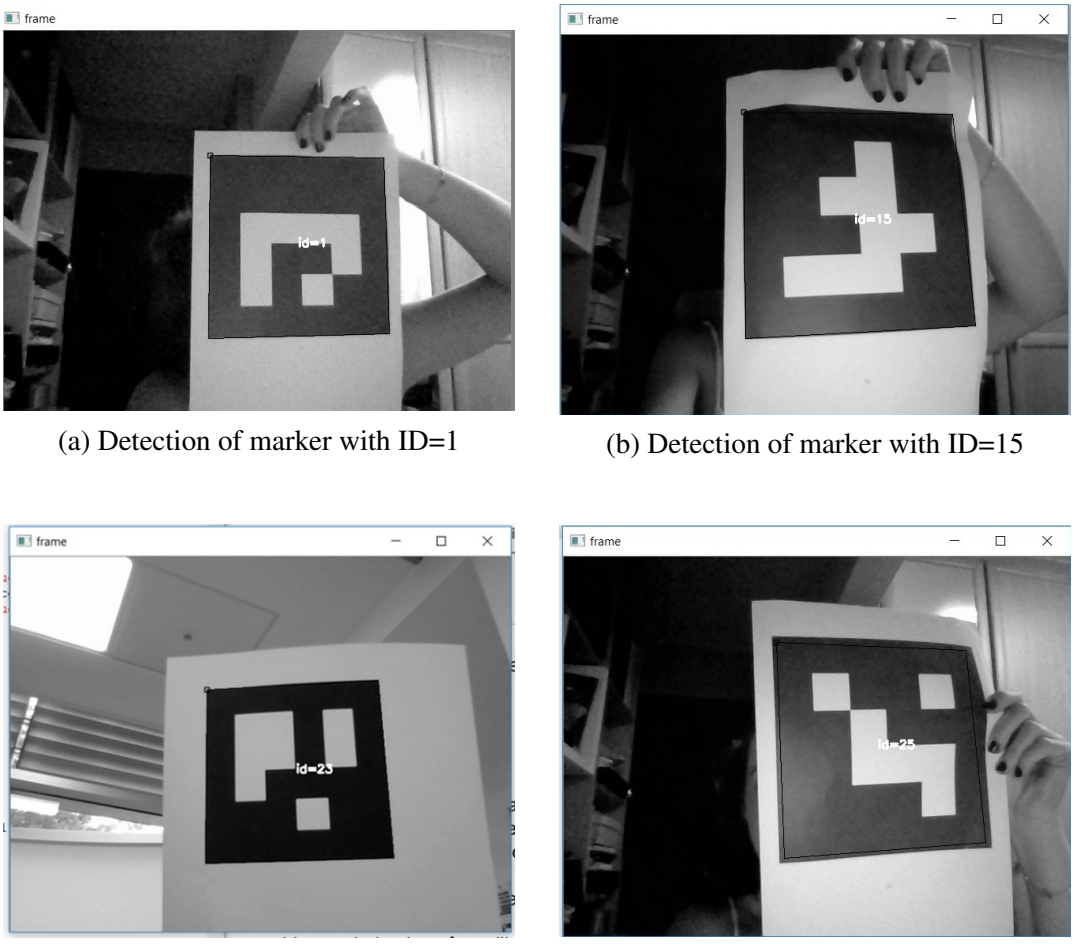
As the whole simulation procedure is going to be run in the student's laptop, the parameters of the laptop's camera web, which is a 'HP TrueVision HD', are defined [1]:

Parameter	Value	Units	Description
–	640×480	pixel	Camera resolution
FOV	60	deg	Field of view

Table 4.1: Camera parameters of the web-cam model employed

[1] Parameters obtained from the laptop HP spectre

The validation of the Marker detection algorithm has been done together with the depth measurement algorithm, as both techniques need the marker detection in order to work. In the next figure the detection proof of the markers with identification numbers equal to 1, 15, 23 and 25 is shown.



(a) Detection of marker with ID=1

(b) Detection of marker with ID=15

(c) Detection of marker with ID=23

(d) Detection of marker with ID=25

Figure 4.1: Detection of the markers employed

For the validation, a LIDAR (Laser Imaging Detection and Ranging) has been used. This is a surveying method that measures the distance to a target by illuminating the target with pulsed laser light and measuring the reflected pulses with a sensor.

In the next figure the operation mode of this type of sensor is represented together with some of its applications:

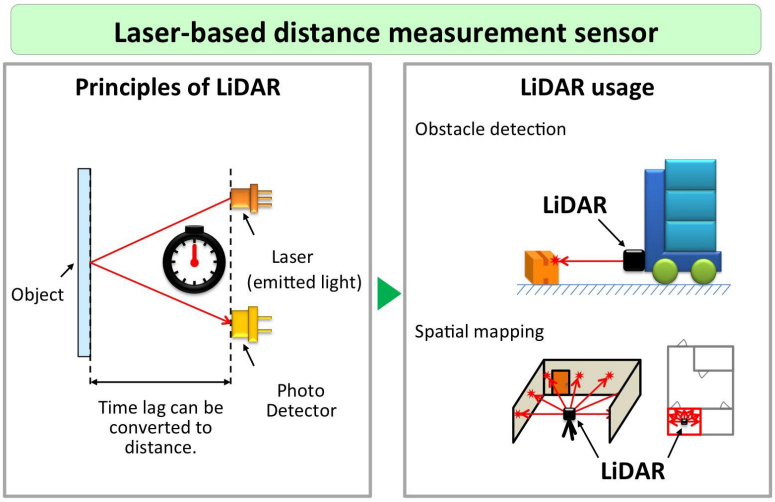


Figure 4.2: Principles and applications of a LIDAR [2]

Compared with a good but 'clasical' tape, a LIDAR has many advantages on many aspects: speed (time spent to take the measurement), accuracy, safety, versatility and functionality.

For the test, two laptops have being placed next to each other. In one of them the marker detection was performed, so the Lidar must reflect in the screen of this computer. In the other one, the measures taken by the Lidar are shown. These validation were performed in the flying laboratory of Building 83 of Cranfield University. This laboratory has a depth of around 10 meters and it must be added that it did not have good lighting conditions.

The procedure followed has been the following one. A person was holding the Lidar together with the marker, so that both of them were located to the same distance from the computer. Then, many distance measurements were performed and the algorithm was tested for various distance separation up to 9 meters. Then, absolute errors have been calculated for each distance separation and the average of the results obtained are presented in the following Figure 4.3.

[2]Image obtained from https://news.panasonic.com/global/stories/2017/51251.html

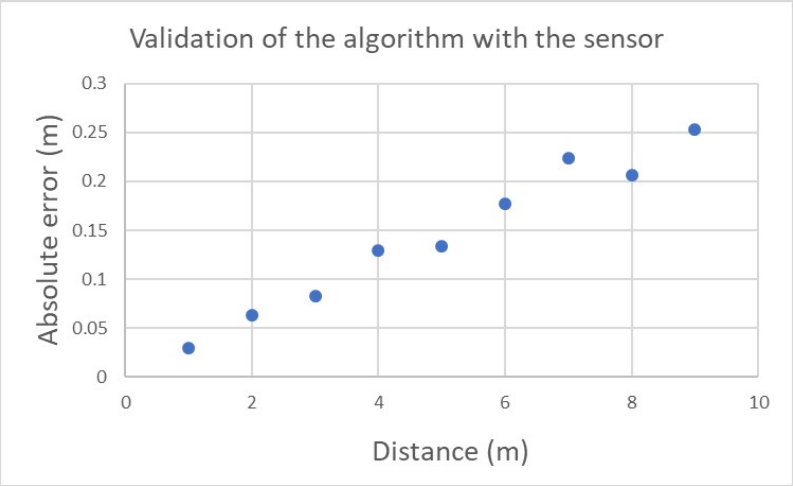


Figure 4.3: Absolute error in Depth estimation

Then, as it was explained in Section 3.3.4, where the visual based constraints were presented, the error that may be produced when an agent i is trying to detect the distance to the marker that agent j is carrying is calculated using Equation 3.14. Therefore, from the experimental data, the variance (σ^2) and the standard deviation (σ) are calculated for each distance separation. In Figure 4.4 the variation of the standard deviation (σ) is shown.

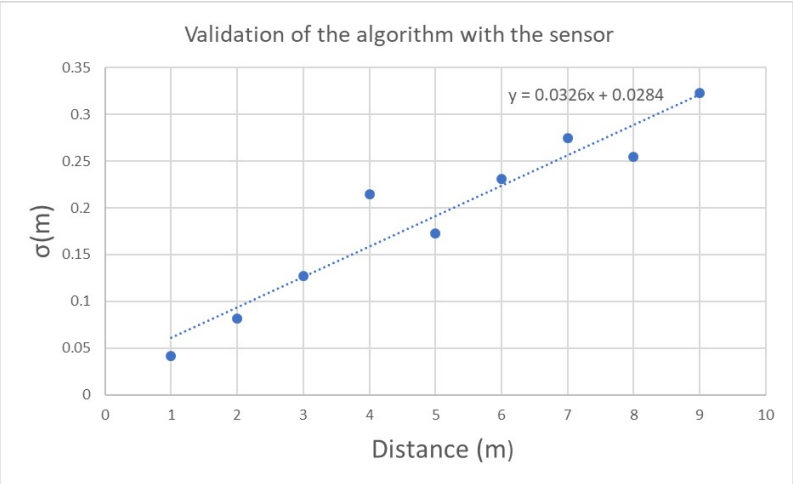


Figure 4.4: Standard deviation in Depth estimation

The values obtained follow approximately a straight line, that is, the points can be

approximated by a first order polynomial with equation $\sigma = 0.0326 \cdot distance + 0.0284$

Regarding Figures 4.3 and 4.4, it can be verified that both absolute error and standard deviation increase with distance. This can be attributed to two main reasons. On the one hand, as it has already been explained, the experiments were carried out in a laboratory without very good lighting conditions. Therefore, the more the marker was separated from the computer, the darker was the zone of the camera frame that contained the marker, hence, the precision of the depth estimation is worst with big distances. On the other hand, the most important reason is attributed to the resolution of the web-cam. In fact, the marker will be contained in a small number of pixels, and a small movement performed by the person or object who is holding the marker involves a variation on the pixel positions of the four corners of the marker. These effect affects more negatively when the marker is inside a smaller number of pixels, that is, with bigger distances.

Nevertheless, if the algorithm was tested in better lighting conditions, better results would be obtained. Furthermore, the marker was still detected up to the 10 meters distance but intermittently at the last measurements, it was not achieved a continuous detection.

4.2 Formation shape

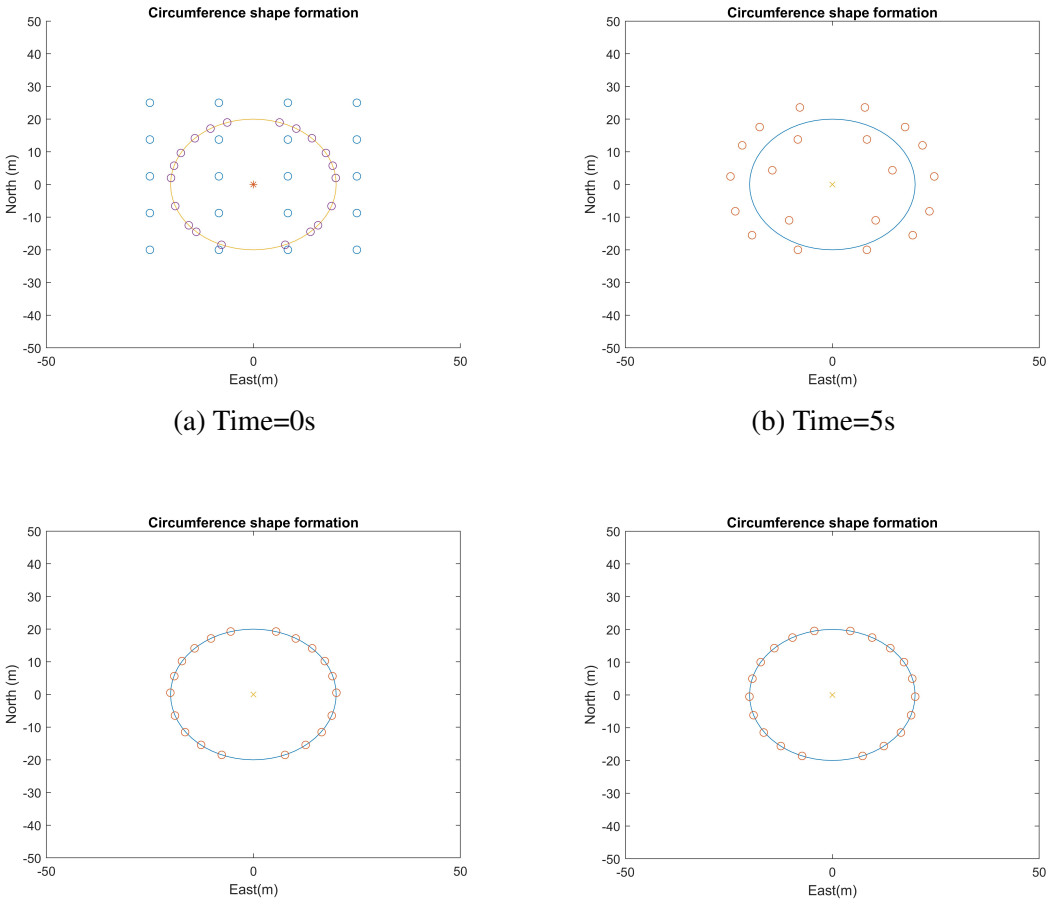
Four different shape formations have been achieved in this project: circular shape, diamond shape, triangular shape and square shape. At a first instance, the evolution of the four formations are shown without any type of visual based constraints, that is, the three restrictions will not be taken into account.

4.2.1 Formation shape with no constraints

At this step it is not going be an objective to obtain the same inter-agent distance, therefore, the following values of c_a, c_r which are the strengths of the attractive and repulsive fields are set:

$c_a = 300$

$c_r = 1050$



(a) Time=0s

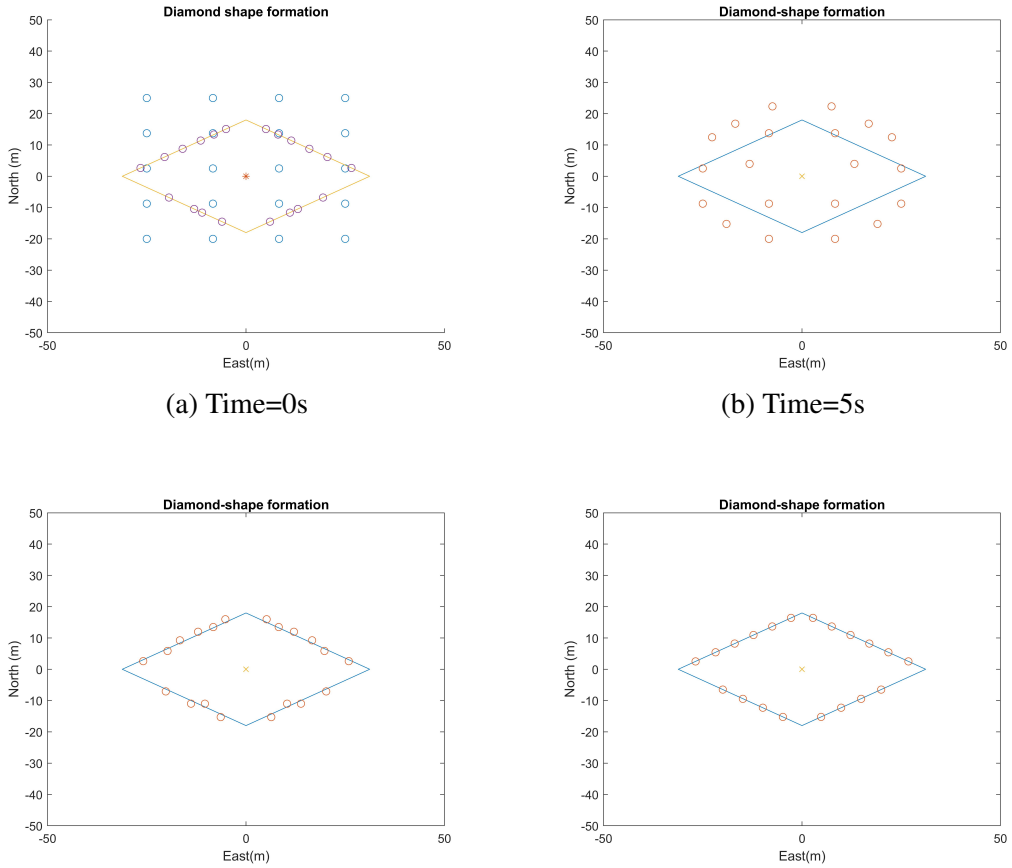
(b) Time=5s

(c) Time=10s

(d) Time=15s

Figure 4.5: Circular formation shape with no constraints

In the two formation shapes shown in Figures 4.5 and 4.6, where in both the same initial conditions are set, a set of 20 UAVs that are uniformly distributed in the plane will go approximating the desired formation shape. For the circular shape, it can be proven how in a smaller time period, the agents will be already located on the formation line. This occurs mainly because of the fact that the circular formation line is to a less distance from the initial position of the UAVs than in the diamond-shape formation line, where some agents will have to travel longer distances to arrive to their goal point.



(a) Time=0s

(b) Time=5s

(c) Time=10s

(d) Time=15s

Figure 4.6: Diamond-shape formation with no constraints

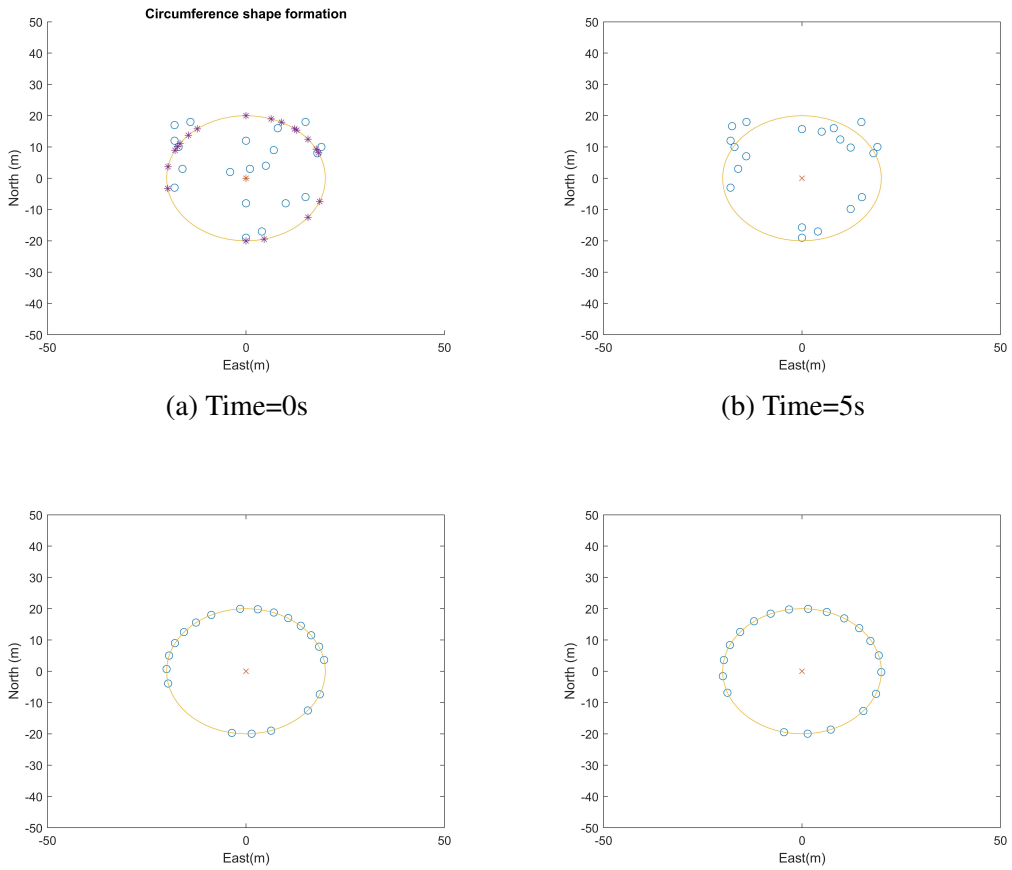
In addition, it can be regarded how $\vec{P}_i^f$, that is, the point on the formation line that guides agent $\vec{P}_i$ to the formation line and which is defined as the intersection point located on the formation line when $\vec{P}_i$ is connected with the reference point (leader vehicle) keeps varying at each time-step. This occurs because at each time step the agents are not moved directly towards their corresponding $\vec{P}_i^f$ because the combination between the attractive and repulsive forces will make that the direction vector of every agent changes at each time step.

Nevertheless, the strengths of the attractive and repulsive potential fields are used to regulate the direction vector that each agent has at a certain time step. If c_a is increased, the direction vector of every agent will be more directed to its destiny point, that is, to a

point above the formation line. When agents start from positions distributed uniformly in the plane, as in Figures 4.5 and 4.6, the separation between agents is enough so that their respective $\vec{P}_i^f$ are not located too near to each other, as shown in Figure 4.5 (a), where the purple circles on the formation line represent $\vec{P}_i^f$ and they are spaced all around the circumference. Therefore, the path that each agent follows will be almost free from collisions, hence, the repulsive forces will not be so important during the journey. In this case, both constants c_a and c_r can have similar values without affecting too much to the results. But the problem arises when agents start from positions located too close to each other. When this occurs, there may be many $\vec{P}_i^f$ with small distance separation among them, and therefore if c_r is not high enough, some agents may collide between each other during their approach to the formation line.

In order to regard the importance of constants c_a and c_r, the formation shapes will be achieved but the set of 20 UAVs will start from some random positions located inside a certain area.

In the simulation captured in Figure 4.7 the values of the constants of the strengths of the attractive and repulsive fields remain $c_a = 300$ and $c_r = 1050$. As a random initial distribution of the UAVs has been set, there is an area where the agent density is much higher comparing with other areas. In addition, in Figure 4.7 (a) together with the initial positions of the UAVs, the 20 points P_i^f are shown, which are very close to each other. If the repulsive potential field was not strong enough to reject neighboring agents that have risk of collision, the simulation would not be correct, as some UAVs would collide with each other and therefore they may break. In order to avoid the problem mentioned, the constant of the repulsive force has been set much higher than the one of the attractive force.



(a) Time=0s

(b) Time=5s

(c) Time=10s

(d) Time=15s

Figure 4.7: Circumference-shape formation with no constraints with an initial random distribution of UAVs

4.2.2 Variation of the attractive and repulsive forces

Now, how do the attractive and repulsive forces calculated with Equation 3.12 and Equation 3.13 change as the formation shape is being achieved is analyzed. For that, the total attractive force F^a_{total} and the total repulsive force F^r_{total} are shown, that is:

$$F^a_{total} = \sum_{i=1}^{N} ||\vec{F}^a_i|| \tag{4.1}$$

$$F^r_{total} = \sum_{i=1}^{N} ||\vec{F}^r_i|| \tag{4.2}$$

First, the attractive force is analyzed in the following Figure 4.8.

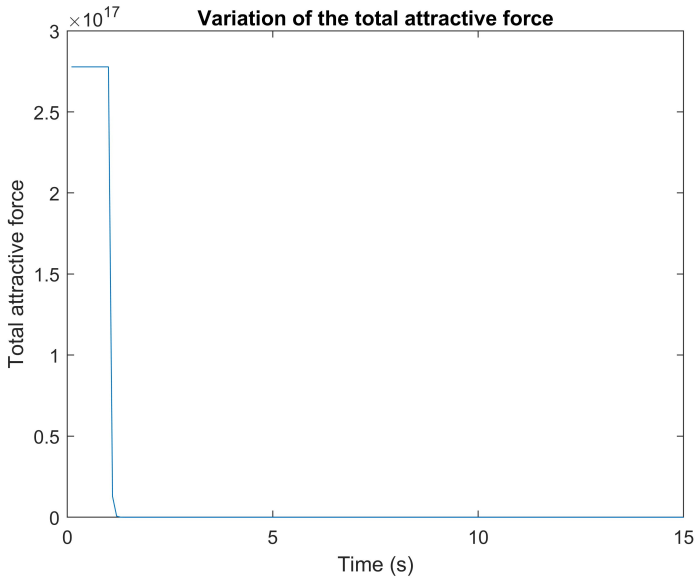


Figure 4.8: Variation of the total attractive force with time (UAVs starting from random positions)

It must be remembered the attractive force expression presented in Equation 3.12. It is a positive exponential function that depends on the distance vector between $\vec{P}_i$ and its respective $\vec{P}_i^f$, that is, the point located on the formation line. The force is directly proportional to $||\vec{\psi}_i^f||$, which is the distance just mentioned, and the exponential is increased by the square of $||\vec{\psi}_i^f||$.

The aim of having a positive exponential where the distance is squared is that when agents are really far away from their destiny point $\vec{P}_i^f$, the attractive force will be really strong so that the agents move in that direction, whereas when they have reached their goal, $\vec{F}_i^a$ will be very small. These features can be regarded in Figure 4.8. At the beginning of the simulation agents are still far from the formation line, but when they start to get closer to it, the total attractive force decreases rapidly.

Once the agents are on the formation line the attractive force should be very small as $||\vec{\psi}_i^f||$ will be almost zero. In order to see the behaviour of the total attractive force after the agents are in the formation line, the previous figure has been enlarged.

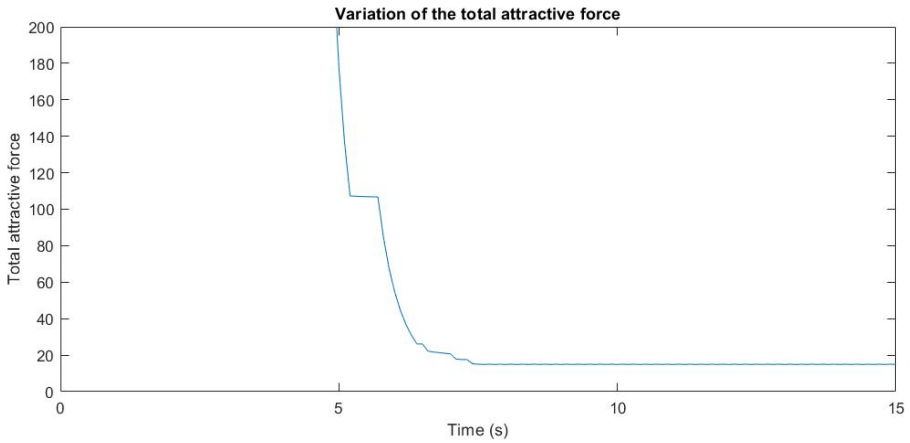


Figure 4.9: Variation of the total attractive force with time (enlarged)

In Figure 4.9 it is regarded how the total attractive force arrives to a certain value in which it is stabilized. For this reason, the total attractive force has been used as the stopping criteria in the simulation, that is, when the value becomes almost unchanging it can be assumed that the agents have arrived to the formation line.

Then, the repulsive force is analyzed. The shape that the total repulsive force adopts depends a lot on the current positions of the UAVs, therefore, depending on their initial distribution one or another shape will be achieved. In Figure 4.10, the evolution of the total repulsive force for achieving a circular shape formation with agents starting from random positions is shown.

The total repulsive force will depend on how much distance exists among the agents that form the swarm. Due to the expression of the repulsive force presented in Equation 3.13, the smaller the distance between 2 agents, the bigger the repulsive force that affects them. This has sense because the repulsive force must only be strong when agents are next to each other.

If Figures 4.11 (a) and (b) are compared, little difference can be regarded between the two formation shapes. Therefore, it can be concluded that the form that the total repulsive force adopts will mostly depend on the initial location of the UAVs. Nevertheless, some common behaviour can be regarded in all of them. When approaching the formation,

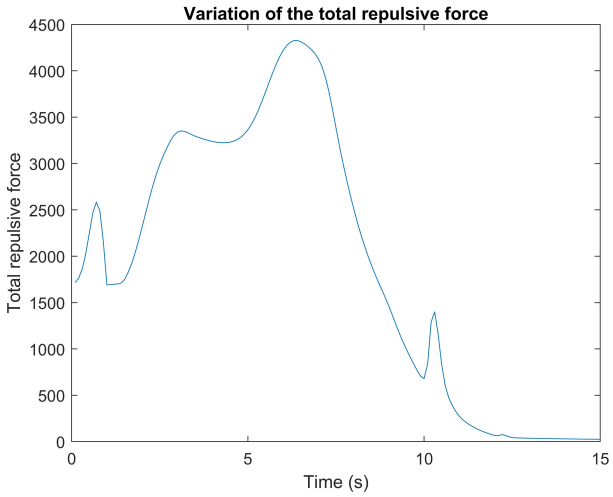
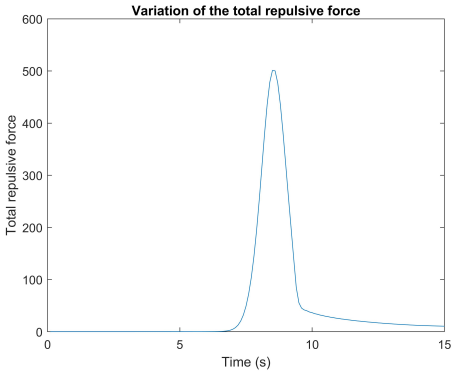
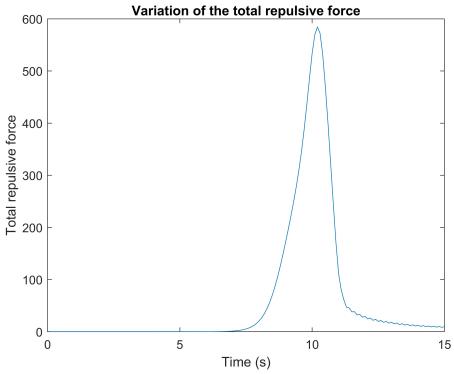


Figure 4.10: Variation of the total attractive force with time (Initial random distribution)



(a) Variation of the total repulsive force (Circular shape)



(b) Variation of the total repulsive force (Diamond shape)

Figure 4.11: Variation of the total attractive force with time (Initial uniform distribution)

the repulsive force will be increased, that is, there is a relative maximum. These relative maximum will make possible that agents maintain a certain inter-agent distance among them once they are on the formation line. Furthermore, due to the increase in the repulsive force, as it will be explained later, it will be possible to achieve a uniform distribution of the vehicles on the formation shape, that is, same inter-agent distance will be possible to reach.

4.2.3 Effect of the visual based constraints

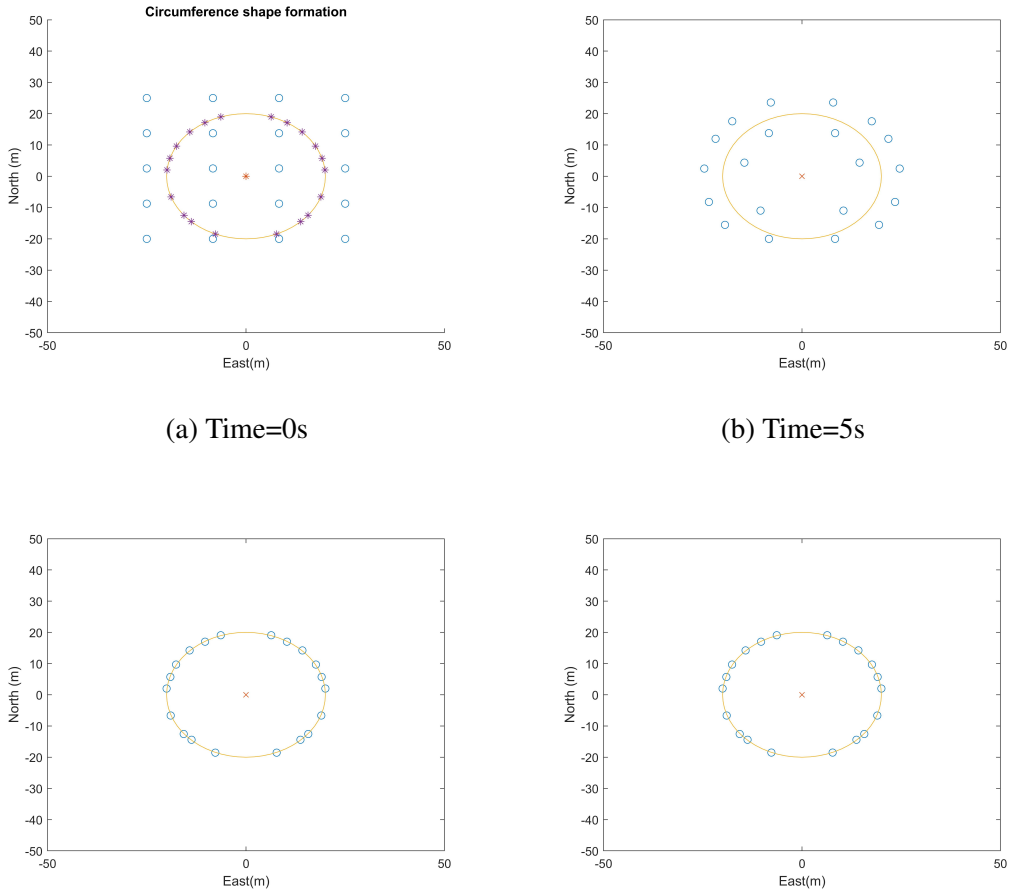
From this moment on, the effect that the three visual based constraints have on the simulation is analyzed. The three constrains that were already explained in the previous chapter can be summarized as:

- Error when an agent detects its relative distance to another agent
- Maximum detectable relative distance
- Field of View limitation

Among the three constraints just mentioned, the one that limits more the performance of the shape formation algorithm is the Field of View limitation, as there is a big difference between the fact that an agent can detect any other agent that is in a any point inside the 360° that surrounds it or just the agents that are inside a limited Field of View of 60°, for example.

Therefore, first, with fixed values of $c_a = 300$ and $c_r = 1050$, the differences respect the previous case when no constraints were considered are analyzed. In addition, a maximum detectable relative distance of 25 meters have been set, and a FOV of 60°.

The results obtained for the circular and diamond shape are shown in the next Figures 4.12 and 4.13.



(a) Time=0s

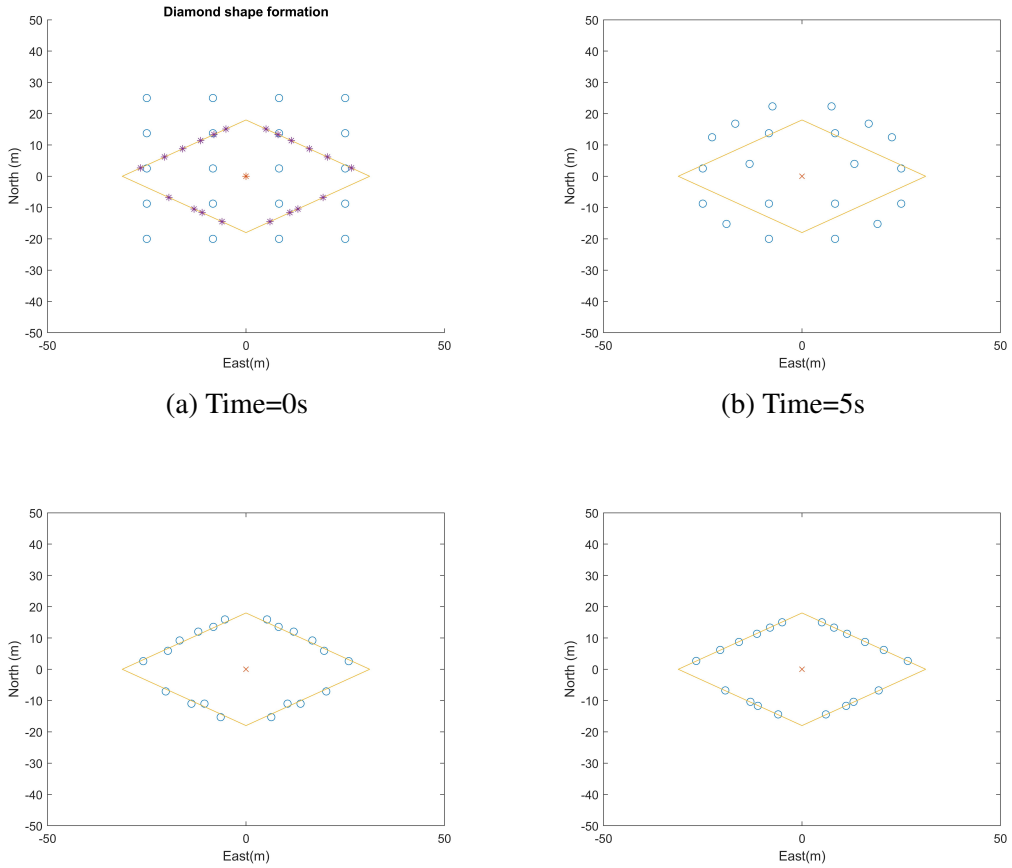
(b) Time=5s

(c) Time=10s

(d) Time=15s

Figure 4.12: Circular formation shape with constraints

It has been proven how, even though the three constraints have been introduced, the objective of achieving a shape formation among the agents is accomplished. Nevertheless, due to the fact that some agents do not see each other, the total repulsive force will be much less than in the previous analysis. That is, some agents may not have repulsive force interacting with each other. As a consequence, when agents are already located on the formation line, they will be closer to each other than in the simulation process where no constraints were taken into account.



(a) Time=0s

(b) Time=5s

(c) Time=10s

(d) Time=15s

Figure 4.13: Diamond-shape formation with constraints

4.2.4 Equal inter-agent distance

In order to achieve an equal inter-agent distance, two methodologies are adopted. Both of them have something in common, which is that the strength of the repulsive force c_r must be extremely high comparing it with the strength of the attractive field c_a. For the simulation, after having tried several combinations, the following values have been set:

$c_a = 300$

$c_r = 105000$

The procedure explained in the methodology is been carried out. First, as shown in the previous cases, every agent will be on the formation line. But as the initial distribution

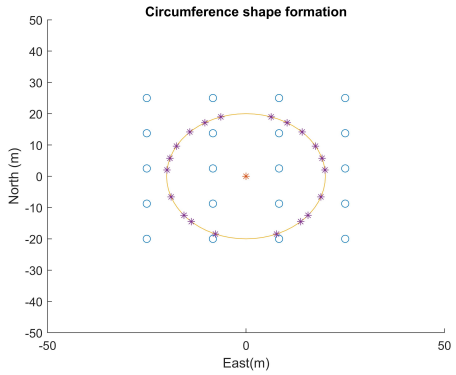
of the agents is not symmetrical respect the vertical and horizontal lines that cross the reference point, once the UAVs are forming a certain shape, the positions that the vehicles adopt in the shape (circumference, diamond, square or triangle) will not be either symmetrical.

But once all the agents are located on the formation line, even though the agents do not see each other, it is known in which direction must the repulsive force be directed to. Therefore, due the repulsive force, zones in which the agent density is higher, that is, where many UAVs are located, will have their agent density reduced. When the equilibrium point is achieved, the inter-agent distance will be equal, that is, all the agents will be located to the same distance from each other.

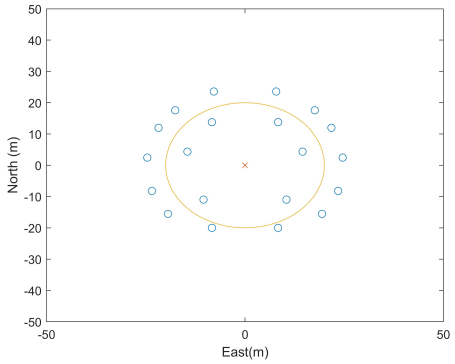
This procedure is more simple in a formation shape with no corners or vertex, as a circumference or an ellipse, because the trajectory that the agents follow when they are moving above the formation line is 'softer'. This can be regarded in Figure 4.14.

However, in any other formation shape that has corners, such as the diamond shape, the triangular shape or the square shape, it will be more difficult to achieve the equal inter-agent distance. The reason for this is that if once the agents are located on the formation line more agents are on one edge than in another, there will be some agents that must 'cross 'a corner in order to arrive to an edge that previously had lower agent density.

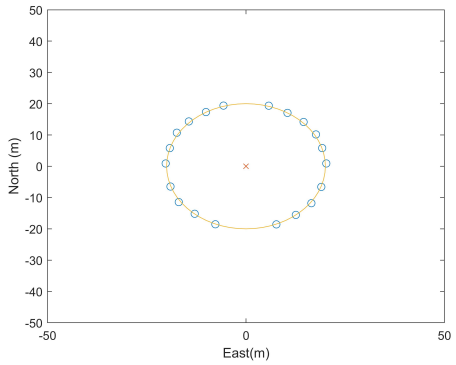
This procedure can be easily regarded in any of the other shapes, as shown in the sub-figures inside Figure 4.15.



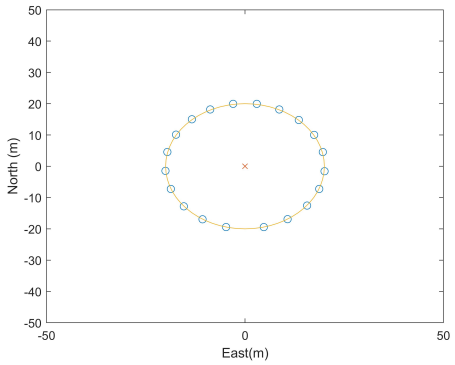
(a) Time=0s



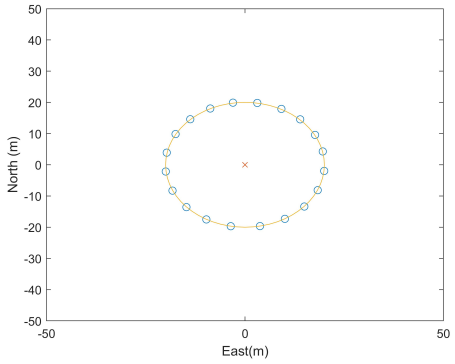
(b) Time=5s



(c) Time=10s

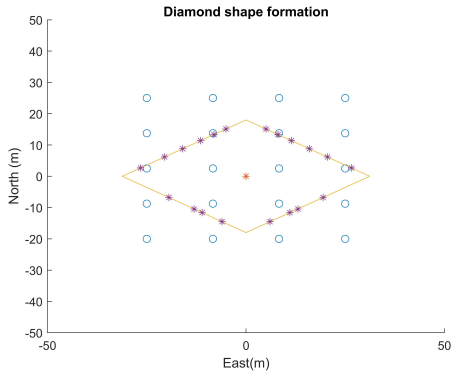


(d) Time=15s

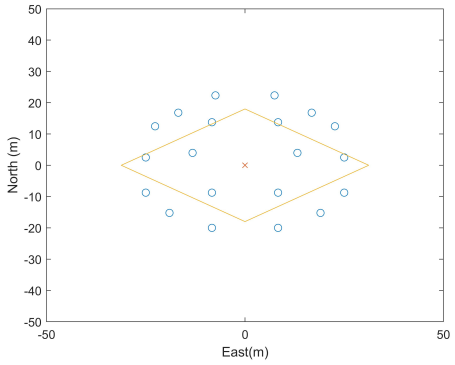


(d) Time=20s

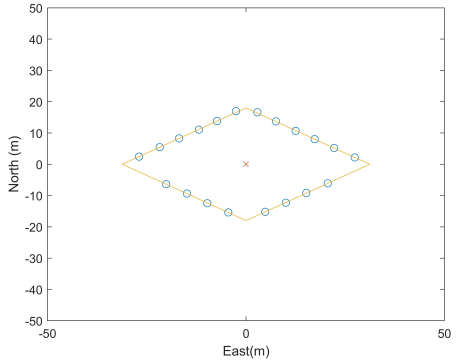
Figure 4.14: Equal inter-agent distance in a circular shape



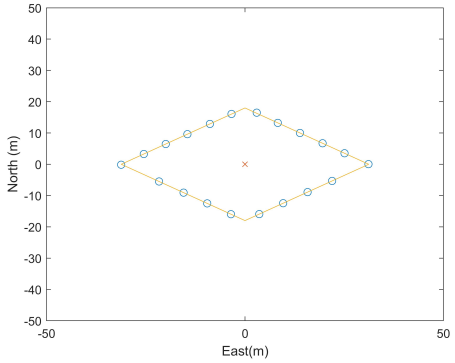
(a) Time=0s



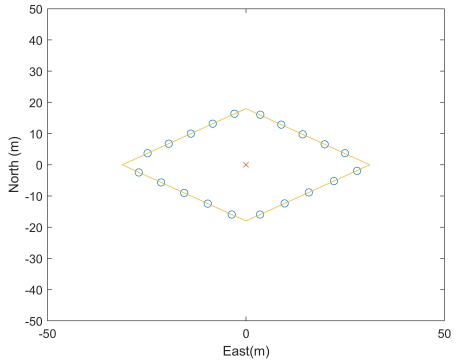
(b) Time=5s



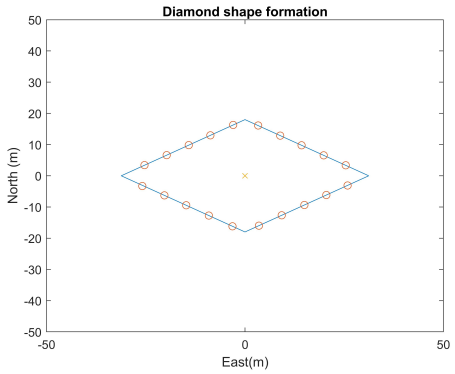
(d) Time=15s



(d) Time=25s



(d) Time=30s



(d) Time=35s

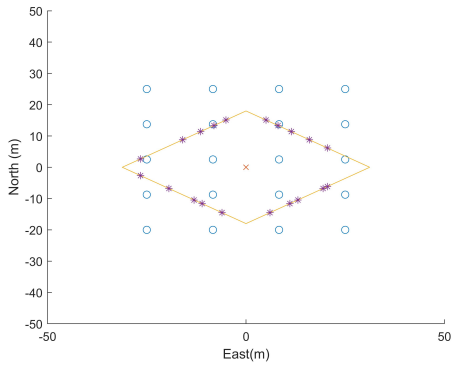
Figure 4.15: Equal inter-agent distance in a diamond shape

In both formation shapes the same inter-agent distance is achieved. Nevertheless, as explained before, more time will be required so that the agents in the diamond shape formation have the same inter-agent distance as a consequence of having to 'cross' a corner. In fact, at the beginning, 6 agents will approach each of the upper edges, and only 4 agents will arrive to the bottom sides. This occurs because at the beginning the UAV distribution is displaced upwards, hence, more $\vec{P}_i^f$ will be located in the upper sides. Therefore, two agents (one in each upper side) will have to move to the bottom edges so that the agent density and the inter-agent distance are equal in every edge of the diamond shape.

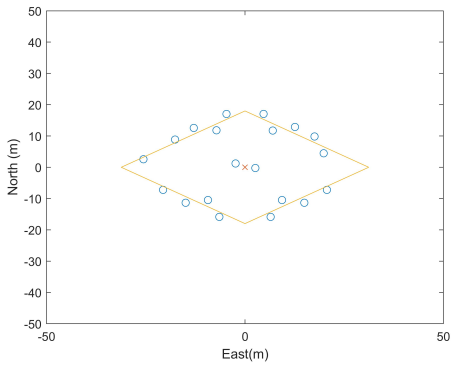
An alternative to the previous methodology in order to obtain the same inter-agent distance in those shapes with corners is that the same number of agents are sent to each edge of the shape formation. In the example just mentioned, as the diamond shape has 4 corners, then, if the swarm is formed by 20 UAVs, 5 vehicles must be directed to each edge. The path that each agent follows is shown in Figure 4.16.

At the first step it will be required to rearrange the UAVs' respective $\vec{P}_i^f$s so that the same number of $\vec{P}_i^f$ points are located in each edge of the diamond. Then, for the particular case shown in Figure 4.16 below, the two candidates that have been selected to go just to the opposite edge must travel a longer distance without colliding with the lead vehicle. This is a very important point because due to the attractive force every agent will be directed to its respective $\vec{P}_i^f$, and in this case if the agents followed a straight line from their current position to their goal position they will just crash into the leader.

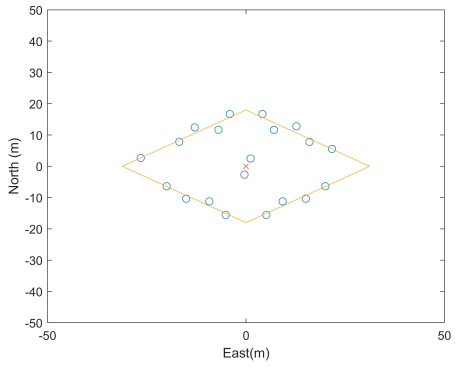
To avoid the above mentioned problem, when agents are close to the leader, a repulsive force with respect to the leader will appear so that the agents go around the lead vehicle. At this point it is important that their 'goal edge' is fixed, because if it is not, when a certain agent goes around the reference point, the location of its $\vec{P}_i^f$ will go varying from one edge to another due to the definition of the point. In fact, it must be remembered that the point $\vec{P}_i^f$ is defined as the intersection point located on the formation line when $\vec{P}_i$ is



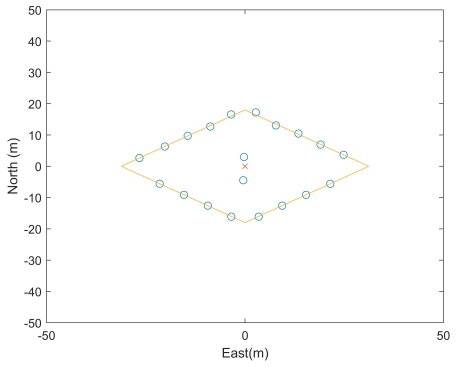
(a) Time=1s



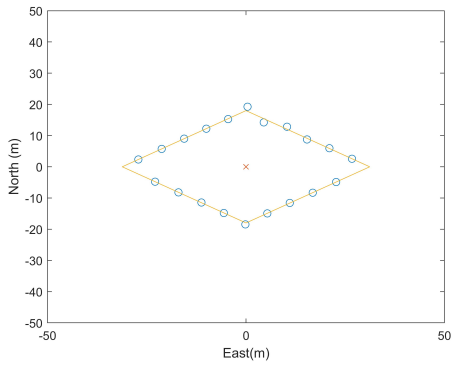
(b) Time=15s



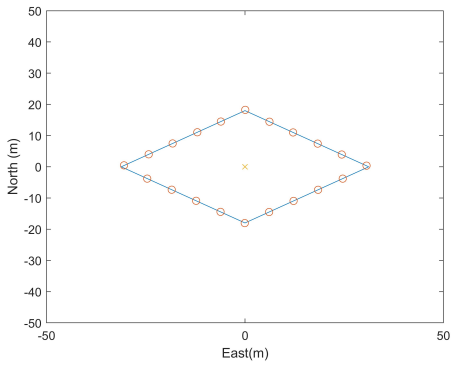
(d) Time=20



(d) Time=25s



(d) Time=35s



(d) Time=43s

Figure 4.16: Equal inter-agent distance in a diamond shape

connected with the reference point. Therefore, it is crucial that the 'goal edge' is fixed from the beginning in order to avoid any confusions so that the agent starts going around the lead vehicle but with no fixed goal direction.

Furthermore, it has been proven how for the same initial conditions, agents will need some more time in order to be on the formation line with the same-inter agent distance than in the previous approach when some agents had to go through a corner. For that reason, if the time required to achieve a formation shape is taken as a variable to prioritise one or ahother method, the first alternative will be chosen as the best option to face the problem of having the same inter-agent distance. Nevertheless, another possible way of achieving the desired goal is commented in the next chapter that may reduce even more the required time. Hence it will be interesting to study some other approach in future researches.

In addition, the formation process of square and triangular shapes are shown in Appendix A.

Chapter 5

Conclusions

Currently, the main way of communication between vehicles is done by Radio Frequency, but due to its various drawbacks, the use of RF wants to be limited. Therefore, another way of communication is proposed: the visual cue based communication.

In this project, a set of autonomous vehicles has been coordinated in order to achieve certain formation shapes among them where just visual cue based information could be used in order to execute the commands to accomplish the formation. The whole thesis has been developed in a simulation environment where the following steps were followed.

First, a marker detection algorithm has been developed. Depending on the ArUco marker shown to the camera, the vehicles will have to adopt a certain shape, which may be a circumference, a diamond shape, a triangle or a square. Once the marker is detected and identified, a potential field based formation shape algorithm makes the vehicles follow a certain path from their initial positions to their final locations on the formation line. The potential field approach has included some constraints in order to adapt it to the real scenario where only information obtained from a visual camera can be used, that is, relative distances and angles. In addition, as in the real scenario it is assumed that each vehicle is carrying a marker of a known size, an algorithm so that any other vehicle can know its relative distance to other vehicles has been developed.

The results obtained can be summarized in the following way:

1. Marker detection algorithm and depth estimation: The performance of both algorithms has been analyzed at the same time as both require a marker to be detected. The performance analysis has been done using a LIDAR. The results show that 18x18cm size square markers are perfectly recognized up to a distance separation of 9 meters using the 'HP TrueVision HD' camera of the laptop which has a camera resolution of 640x480 pixels. Nevertheless, even though the validation has been done in a laboratory with not very good lighting conditions, the marker is recognized for 10 meters distance (which was the maximum depth of the laboratory) but without a continuous marker detection.

In relation to the depth estimation, the absolute error between the estimation done by the algorithm and the Lidar has been calculated. When regarding the results it can be concluded that the absolute error is increased when the separation distance to the laptop is increased. This is attributed to two main reasons.

- On the one hand, when the relative distance was higher, the marker was located in the 'darkest' zone or with worst lighting conditions of the laboratory, so the marker detection process was more ambiguous.

- On the other hand, when distance is increased, due to the resolution of the camera, the marker will be contained in a smaller number of pixels, and a small movements performed by the person or object who is carrying the marker involves a variation on the pixel positions of the four corners of the marker which are used to calculate the depth estimation. This inaccuracy has a bigger impact on the results when the marker is contained in a small number of pixels.

2. Formation shape between vehicles: Depending on the detected marker, the vehicles are forced to adopt a certain formation shape among them. In the project, four formation shapes have been achieved, which can be divided into two main groups: shapes with no corners (a circumference for instance) and shapes with corners (the rest of them).

The main differences arise between both groups. The following conclusions have been obtained:

- When it was not considered the goal of achieving equal inter-agent distance, there were not big differences among the two groups. That is, the four formation shapes were achieved in approximately the same time period. In fact, the required time depended mostly on the initial UAV distribution, that is, how far the agents were located from the formation line.

- In the project three visual based constraints have been included: the error when an agent detects its relative distance to another agent, the maximum detectable relative distance, and the FOV limitation. When these constraints are added, the formation shape is also achieved by the UAVs, but as the repulsive force is less due to the constraints, the agents will have a higher risk of collision and they could also end closer to each other in some zones once they are on the formation line. Nevertheless, this last problem is later solved when achieving the same distance between vehicles.

- Regarding the attractive and repulsive forces that are used to design the control law, it has been proven how the total attractive force is exponentially decreased with time as agents are approaching the formation line. Because of this, in the simulation the total attractive force is employed for the stopping criteria in order to know if the agents are already on the formation line or not.

- Equal inter-agent distance was achieved. The best way of ensuring it is increasing the strength of the repulsive force so that once every agents is on the formation line as it is known the direction of the repulsive force among agents, an equilibrium point will be reached. It is based on the idea that in those areas where the agent density is higher, the vehicles will be spread so that the agent density in other zones is increased. When the visual based constraints are introduced, the time required to achieve the final uniform formation will be higher for the reason mentioned above. In addition, when the formation shape has some corners, the time needed will also

be higher. This happens because if the number of agents on each edge of the formation shape is not equal, one or some agents will have to 'cross' a corner so that the number of agents in each edge is the same. It has been proven how in 'rounded' shapes the 'crossing' is softer; thus, faster.

5.1 Future Work

To end, some other steps that could be done in order to continue with this research topic are proposed:

- **Implementation in real scenarios**: Due to the limited available time that has been to develop the project, it has not been possible to test the algorithm in a real environment. Therefore, a future step should be to implement the algorithm in a dynamic platform with real vehicles. In fact, when testing the algorithm in a real environment other factors may affect the correct performance of the formation.

- **Adaptation to different scenarios**: In this project is has been assumed that depending on the type of mission the vehicles are carrying out, a certain marker will be 'shown' by the leader to the rest of vehicles. But in order to make it more real, some different scenarios could be designed, with some obstacles... hence, depending on the current environment, the program should be able to decide which is the most suitable formation shape to follow which will be the one that the leader indicates to the rest of vehicles.

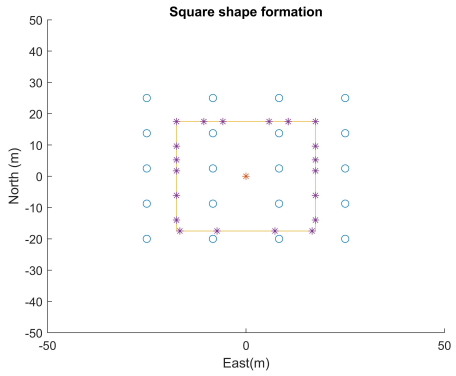
- **Equal inter-agent distance:** In this project it has been achieved the objective of obtaining the same inter-agent distance in two ways. In the approach where the same number of agents were guided to each edge of a formation shape it has not been included any algorithm which decides which agent should be the one that goes to the opposite edge. Therefore, it should be a good improvement if a variable such as the distance to the opposite edge was taken into account, so that the agent

that minimizes that distance is the one who changes its respective point $\vec{P}_i^f$ to the opposite edge.

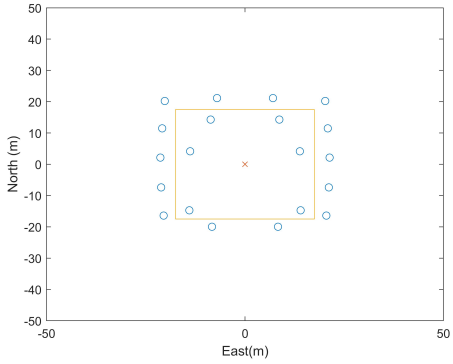
Appendix A

Extra Data

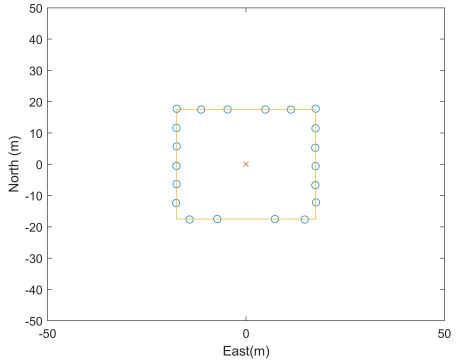
In this chapter, the square and triangular shape formations that have also been achieved are shown.



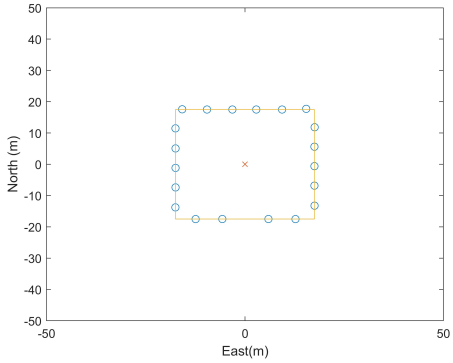
(a) Time=0s



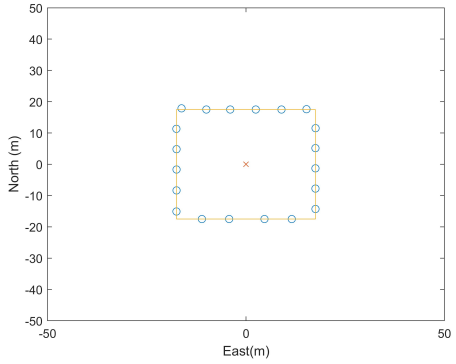
(b) Time=5s



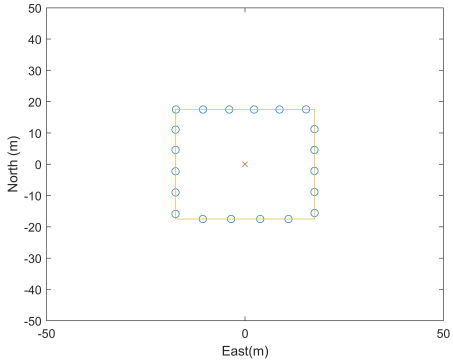
(d) Time=10s



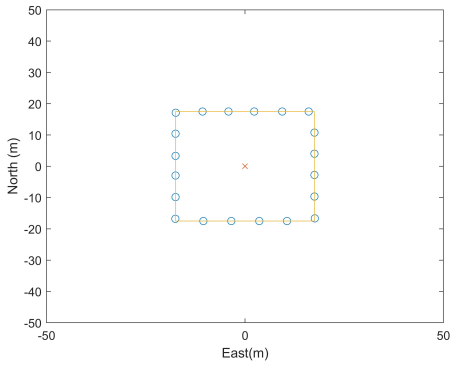
(d) Time=15s



(d) Time=20s

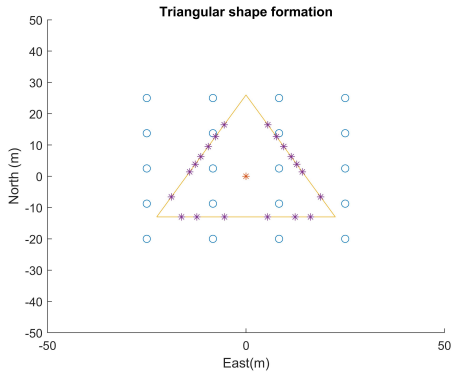


(d) Time=25s

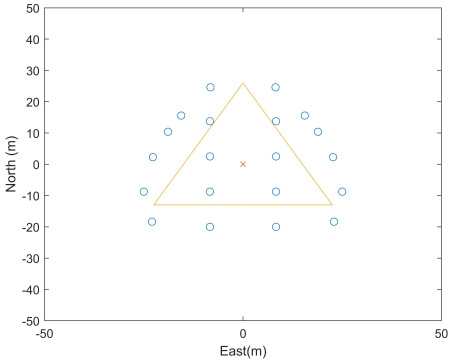


(d) Time=30s

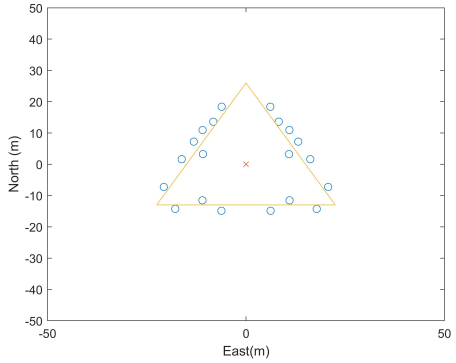
Figure A.1: Square shape formation



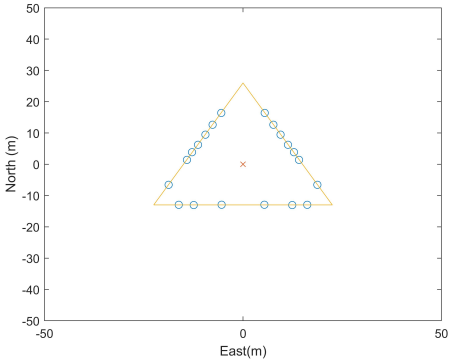
(a) Time=0s



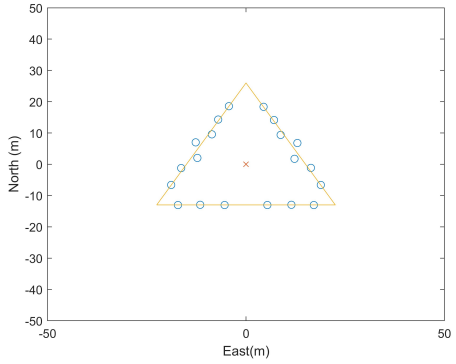
(b) Time=5s



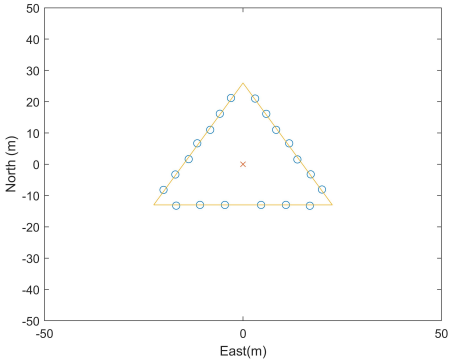
(d) Time=10s



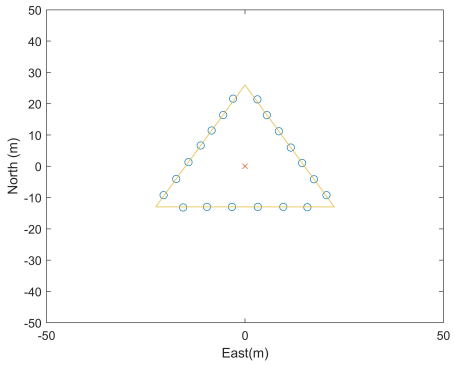
(d) Time=15s



(d) Time=20s



(d) Time=25s



(d) Time=30s

Figure A.2: Triangular shape formation

Bibliography

Azuma, R. T. (n.d.), 'A Survey of Augmented Reality', p. 48.

Benligiray, B., Topal, C. and Akinlar, C. (2017), 'Stag: A stable fiducial marker system', *CoRR* **abs/1707.06292**.

Bitsakos, K. and Fermüller, C. (2006), 'Depth estimation using the compound eye of dipteran flies', *Biological Cybernetics* **95**(5), 487–501.
URL: *http://link.springer.com/10.1007/s00422-006-0097-1*

Campa, G., Gu, Y., Seanor, B., Napolitano, M. R., Pollini, L. and Fravolini, M. L. (2007), 'Design and flight-testing of non-linear formation control laws', *Control Engineering Practice* **15**(9), 1077–1092.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S096706610700024X*

Dang, A. D. and Horn, J. (2014*a*), 'Intelligent Swarm-Finding in Formation Control of Multi-Robots to Track a Moving Target', **8**(4), 7.

Dang, A. D. and Horn, J. (2014*b*), Path planning for a formation of autonomous robots in an unknown environment using artificial force fields, IEEE, pp. 773–778.
URL: *http://ieeexplore.ieee.org/document/6982512/*

Do, K. (2006), Formation control of mobile agents using local potential functions, IEEE, p. 6 pp.
URL: *http://ieeexplore.ieee.org/document/1656537/*

Dorfmüller, K. and Wirth, H. (1998), Real-Time Hand and Head Tracking for Virtual Environments Using Infrared Beacons, *in* G. Goos, J. Hartmanis, J. van Leeuwen, N. Magnenat-Thalmann and D. Thalmann, eds, 'Modelling and Motion Capture Techniques for Virtual Environments', Vol. 1537, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 113–127.
URL: *http://link.springer.com/10.1007/3-540-49384-0*$_9$

Fiala, M. (2010), 'Designing Highly Reliable Fiducial Markers', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32**(7), 1317–1324.
URL: *http://ieeexplore.ieee.org/document/5184844/*

Flohr, D. and Fischer, J. (n.d.), 'A Lightweight ID-Based Extension for Marker Tracking Systems', p. 7.

Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F. and Marín-Jiménez, M. (2014), 'Automatic generation and detection of highly reliable fiducial markers under occlusion', *Pattern Recognition* **47**(6), 2280–2292.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S0031320314000235*

Gu, Y., Seanor, B., Campa, G., Napolitano, M., Rowe, L., Gururajan, S. and Wan, S. (2006), 'Design and Flight Testing Evaluation of Formation Control Laws', *IEEE Transactions on Control Systems Technology* **14**(6), 1105–1112.
URL: *http://ieeexplore.ieee.org/document/1709935/*

H DOUGLAS, D. and K PEUCKER, T. (1973), 'Algorithms for the reduction of the number of points required to represent a digitized line or its caricature', **10**, 112–122.

Iskandarani, M., Givigi, S. N., Fusina, G. and Beaulieu, A. (2014), Unmanned Aerial Vehicle formation flying using Linear Model Predictive Control, IEEE, pp. 18–23.
URL: *http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6819230*

Johnson, E. N., Calise, A. J., Watanabe, Y., Ha, J. and Neidhoefer, J. C. (2007), 'Real-Time Vision-Based Relative Aircraft Navigation', *Journal of Aerospace Computing,*

Information, and Communication **4**(4), 707–738.
URL: *http://arc.aiaa.org/doi/10.2514/1.23410*

Jung, H. and Kim, D. H. (2014), 'Potential-function-based shape formation in swarm simulation', *International Journal of Control, Automation and Systems* **12**(2), 442–449.
URL: *http://link.springer.com/10.1007/s12555-013-0133-6*

Kato, H. and Billinghurst, M. (1999), Marker tracking and HMD calibration for a video-based augmented reality conferencing system, IEEE Comput. Soc, pp. 85–94.
URL: *http://ieeexplore.ieee.org/document/803809/*

Koo, T. and Shahruz, S. (2001), Formation of a group of unmanned aerial vehicles (UAVs), IEEE, pp. 69–74 vol.1.
URL: *http://ieeexplore.ieee.org/document/945515/*

Li, N. H. M. and Liu, H. H. T. (2008), Formation UAV flight control using virtual structure and motion synchronization, IEEE, pp. 1782–1787.
URL: *http://ieeexplore.ieee.org/document/4586750/*

Lowe, D. (1999), Object recognition from local scale-invariant features, IEEE, pp. 1150–1157 vol.2.
URL: *http://ieeexplore.ieee.org/document/790410/*

Naimark, L. and Foxlin, E. (2002), Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker, IEEE Comput. Soc, pp. 27–36.
URL: *http://ieeexplore.ieee.org/document/1115065/*

Ning, A., Flanzer, T. C. and Kroo, I. M. (2011), 'Aerodynamic Performance of Extended Formation Flight', *Journal of Aircraft* **48**(3), 855–865.
URL: *http://arc.aiaa.org/doi/10.2514/1.C031046*

No, T. S., Kim, Y., Tahk, M.-J. and Jeon, G.-E. (2011), 'Cascade-type guidance law design for multiple-UAV formation keeping', *Aerospace Science and Technology* **15**(6), 431–439.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S127096381000115X*

Paul, T., Krogstad, T. R. and Gravdahl, J. T. (2008), 'Modelling of UAV formation flight using 3d potential field', *Simulation Modelling Practice and Theory* **16**(9), 1453–1462.
URL: *http://linkinghub.elsevier.com/retrieve/pii/S1569190X08001469*

Peng, Z., Wang, D. and XJ, H. (2011), 'Robust adaptive formation control of underactuated autonomous surface vehicles with uncertain dynamics', **5**, 1378–1387.

Rekimoto, J. (1998), Matrix: A realtime object identification and registration method for augmented reality, *in* 'APCHI'.

Ribo, M., Pinz, A. and Fuhrmann, A. (2001), A new optical tracking system for virtual and augmented reality applications, Vol. 3, IEEE, pp. 1932–1936.
URL: *http://ieeexplore.ieee.org/document/929537/*

Shi, X. and Yang, J. (2013), Design of formation flight control based on virtual body frame of the team formation, IEEE, pp. 796–800.
URL: *http://ieeexplore.ieee.org/document/6561030/*

Subbarao, M. and Surya, G. (1994), 'Depth from defocus: A spatial domain approach', *International Journal of Computer Vision* **13**(3), 271–294.
URL: *http://link.springer.com/10.1007/BF02028349*

SUZUKI, S. (n.d.), 'Topological Structural Analysis of Digitized Binary Images by Border Following', p. 15.

Tang, H., Cohen, S., Price, B., Schiller, S. and Kutulakos, K. N. (2017), Depth from Defocus in the Wild, IEEE, pp. 4773–4781.
URL: *http://ieeexplore.ieee.org/document/8099990/*

Vachon, M. J., Ray, R., Walsh, K. and Ennix, K. (2002), F/A-18 Aircraft Performance Benefits Measured During the Autonomous Formation Flight Project, American Institute of Aeronautics and Astronautics.
URL: *http://arc.aiaa.org/doi/10.2514/6.2002-4491*

Wagner, D., Reitmayr, G., Mulloni, A., Drummond, T. and Schmalstieg, D. (2010), 'Real-Time Detection and Tracking for Augmented Reality on Mobile Phones', *IEEE Transactions on Visualization and Computer Graphics* **16**(3), 355–368.
URL: *http://ieeexplore.ieee.org/document/5226627/*

Wagner, D. and Schmalstieg, D. (n.d.), 'ARToolKitPlus for Pose Tracking on Mobile Devices', p. 8.