Decentralized Control of Groups of Autonomous UAVs

Mark Bastourous

Normandie Université

THÈSE

Pour obtenir le grade de **Docteur de Normandie Université**

Spécialité **Informatique**

Préparée au sein des laboratoires **LITIS** (Laboratoire d'Informatique du Traitement de l'Information et des Systèmes - EA 4108) et **GREAH** (Groupe de Recherche en Electrotechnique et Automatique du Havre - EA 3220) de **l'Université le Havre Normandie**

Decentralized Control of Groups of Autonomous UAVs

Présentée et soutenue par **Mark Bastourous**

Thèse soutenue publiquement le XX-12-2021 devant le jury composé de

David Fofi	Professeur	Université De Bourgogne-Franche-Comté	Rapporteur
Simon G. Fabri	Professeur	University of Malta	Rapporteur
Samira Hayat	Senior Researcher	University of Klagenfurt, Lakeside Labs GmbH	Examinatrice
Julien Marzat	Chercheur HDR	ONERA	Examinateur
Frédéric Guinand	Professeur	Université le Havre Normandie	Directeur de thèse
François Guérin	MdC HDR	Université le Havre Normandie	Encadrant

litis

UNIVERSITÉ LE HAVRE NORMANDIE

GREAH

Remerciements

Many thanks.

Résumé

Au cours des dernières décennies, de multiples travaux ont montré que l'apport des robots pouvait être significatif pour des domaines très variés, tels que la logistique, la surveillance, les opérations de secours, l'agriculture de précision ou la médecine pour n'en citer que quelques uns. L'éventail des applications potentielles s'élargit encore lorsque le système est composé de plusieurs robots fonctionnant en groupe, un système communément appelé *Multi-Robot System (MRS)*.

Dans le cadre du déploiement d'un MRS de nombreux défis restent, encore aujourd'hui, à relever.

Parmi les principaux problèmes, le déplacement coordonné du groupe, avec la prise en compte de contraintes de robustesse est central. Très souvent, la gestion de la coordination entre les robots d'un groupe est déléguée à un ordinateur central qui, sur la base des deonnées transmises par les robots, détermine une planification du déplacement puis transmet à tous les robots du groupe un ensemble de commandes que ceux-ci doivent exécuter. Une telle approche est sensible aux perturbations et aux pannes. En effet, si un problème survient sur l'ordinateur central ou si la qualité des communications se dégrade, le système dans son ensemble devient inopérant comme la montré récemment un vol coordonné de manière centralisé en Chine qui a finalement abouti à la chute des machines au milieu du public[1]. De plus, dans la majorité des cas, la centralisation, du fait de la nécessité de connaître à chaque instant le nombre de machines en cours d'utilisation, rend plus difficile l'extensibilité du système par l'ajout ou le retrait de robots.

La littérature dans le domaine des MRS, riche de travaux s'intéressant aux problèmes de coordination dans les systèmes décentralisés et distribués, fait apparaître plusieurs types d'approches, depuis les schémas Leader-Followers (LF) jusqu'aux techniques de vol en essaim en passant par l'algorithmique distribuée. Un second problème, lié au premier, est souvent abordé, il s'agit de la manière dont les informations transitent entre les machines, avec des questions relatives à la quantité et au type de données qui doivent être échangés au sein du groupe, mais également relatives à la qualité des communications et des perturbations pouvant les affecter, comme la latence, la perte temporaire de connexion et les délais de transmission.

Ce travail de thèse a pour objectif d'apporter des solutions pour le vol en groupe dans le cadre d'approches décentralisées et distribuées. Les algorithmes présentés dans ce

[1] https://youtu.be/gohD23-0UBw vidéo consultée le 30 octobre 2021

manuscrit prennent en considération des conditions réalistes pour les communications, pour les référentiels considérés ainsi que pour la localisation des machines.

La première contribution présentée est un contrôleur de haut niveau pour le vol en formation dans un schéma de type Leader-Followers (LF). Le contrôleur présente quelques caractéristiques clefs qui peuvent être résumées par sa facilité de réglage et sa capacité à s'adapter à des communications intermittentes, en compensant les informations perdues par des données des capteurs embarqués, en provenance principalement de capteurs de vision.

La seconde contribution de ce travail concerne toujours le vol en formation selon un schéma LF mais en l'absence de communications entre les robots. Dans ce contexte, un composant prédictif est ajouté à un filtre de Kalman étendu (EKF). Une autre caractéristique de la méthode mise au point est l'absence de mesure directe de la distance, en lui substituant les fonctionnalités des moments d'images pour l'asservissement visuel, une technique connue sous le nom de IBVS (*Image Based Visual Servoing*). Cette méthode n'a pas fait l'objet de nombreux travaux pour les systèmes sosu-actionnés, tels que les drones multirotors (quadri, hexa, etc.). De plus cette technique n'a pas été étudiée dans le domaine MRS de la même manière que nous le proposons dans ce travail, par l'utilisation des moments des images. De bons résultats ont été obtenus à la fois en simulation et lors d'expériences réelles.

Enfin, pour la dernière contribution, nous nous intéressons au problème de l'alignement de trois machines sans communication entre elles et sans aucune mesure de distance, toujours dans un contexte décentralisé. A l'aide d'un algorithme utilisant seulement les angles entre les robots, nous montrons que la probabilité théorique de parvenir à l'alignement est de 1, mais diminue lorsqu'une distance de sécurité non nulle entre les robots est définie.

Abstract

Robotics have proven over the last decades to have positive impact across many domains, medical, logistics, surveying, search and rescue, agriculture, to name a few.. There are more prospect applications for robotics when the system has multiple of them functioning in a group, which is formally known as Multi-Robot System (MRS).

Within such frontiers of deploying MRS come challenges that are not yet resolved till our current days. One of the most important issues is to manage the fleet of robots in a coordinated motion robustly. Any MRS should be developed with algorithms that can guarantee naturally within the group of robots, an obstacle avoidance module. Usually for such coordinated motion to be planned, the data from the different robots within the fleet should be transmitted to a central processing unit, then planning algorithms are calculated and transmit back actuation commands to be executed by each individual robot. Such systems are prone to failure and perturbations. Indeed, if a problem occurs on the central processing unit or if the quality of communications degrades, the system as a whole might become uncontrollable as it happened recently in China during a show using a swarm based on a centralized coordination system.[2] Moreover, in most cases, centralization obstructs system scalability. Indeed, scalability means that at any moment, some robots can join or leave the group, a possibility difficult to consider when all information have to go through one central processing unit.

MRS literature have many propositions on how to address the coordination challenge. Among the most cited approaches we can distinguish, Leader-Followers (LF) schemes, distributed algorithms and swarming techniques. A second issue, linked to the first one, is often considered: the way information are transmitted between the robots, with concerns related to the quantity and type of data that should be exchanged within the group, but also related to the quality of the communications, latency and delay, and perturbations that might occur potentially leading to temporary loss of connection.

In this thesis, solutions are proposed within a distributed decentralized frameworks. Algorithms developed are taking into consideration realistic conditions regarding communications, coordinate frames, and global localization. A first contribution consist of designing a high level decentralized formation controller for MRS in a LF framework. This controller has some key features that can be summed up in being easy to tune, can handle intermittent communications by compensating the missing information utilizing

[2] https://youtu.be/gohD23-0UBw last visit Oct. 30 2021

robot's onboard sensors, significantly the visual feedback. Then a second proposition is concerned with forming LF MRS without inter-robots communications. In this proposition a predictive component is introduced within a classical Extended Kalman Filter (EKF). Another key feature of such system is the exclusion of direct distance measurement, utilizing the image moments Image Based Visual Servoing (IBVS). IBVS has not been widely explored in literature on underactuated systems such as a classical aerial multirotor (quad, hexa). Moreover such technique was not investigated before in MRS society in the same way of image moments. It showed good results in simulation and real experiments.

Finally, for the last contribution, we investigate the problem of the alignment of three robots without any communication between them and without any measure of distance, always in the context of a decentralized approach. Relying only on bearing measurements between robots, we show that the theoretical probability of success is 1, but decreases when a non-zero security distance between drones is defined.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Contents

1.1 Generalities

Robotics studies are attaining more importance due to the need of augmenting humans in order to solve many challenges, be more efficient, environment friendly and achieve tasks more safely. Robots have a long history in industry and their origin can be dated from the early 1960s (Unimate robot). They are mainly used for the so-called DDDD tasks standing for Dull, Dirty, Dangerous And Dear tasks. Since that time robots have spread in a large number of domains and sectors. In industry they are used for welding, painting, clamping, while in mining sector, robots help for mapping sites, inspecting mines without exposing humans to danger, or evaluating some volumes of minerals. In agriculture, using specific sensors (i.e: NDVI), robots are able to detect drought stress in some parts of the inspected fields or to measure health of crops. In construction sector, which is physically demanding, they are mainly used for maintenance and inspection tasks, especially when dangerous (large height or hazardous environment). In transport and logistics, at the extremities of the supply chain, big companies have great hope about autonomous container ships, and there are intense discussions about last mile delivery possibilities offered by drones in urban environments. But robots are also very present

nowadays in areas where they were, a priori, less expected such as leisure (show, races), ecology (endangered species tracking and monitoring), environmental monitoring, human health (medical cargo (organs, medics, testing kits) transportation, surgical) or everyday life (vacuum cleaner, lawn mower for instance).

Quantities of factors have contributed to this diffusion. They are to be found in the economy (lower price, birth of new manufacturers), technological improvements (better on-board computing, computing power increased tenfold, software frameworks availability). Also there are scientific advances which have solved many problems and which have made it possible to provide robots with better mobility and relatively higher autonomy. Finally another factor has played an important role in the diffusion of robots in recent years, is the ubiquitous use of drones.

1.2 Motivation

While many robots are deployed in the time when this study takes place, there are many more tasks that could be automated or delegated from humans to machines, specially in the context of multi robot systems. That draws the need to do more rigorous research in both fundamental and field robotics aspects on both the hardware and software.

Additionally to such aforementioned applications there are numerous advantages of using multiple robots in missions. For instance the use of Multi Robot Systems (MRS) bring flexibility, robustness, efficiency and redundancy, fault tolerance and scalability to the systems used in missions. Also capabilities that can never be met with only one robot as heterogeneous sensing, time reduction required for many tasks like search and rescue, inspections, surveillance, emergency delivery. Particularly research on multi aerial robots (drones) to collectively execute these tasks are gaining traction in academic literature.

In the domain of civil security for instance, when telecommunication networks fall down after a natural disaster (hurricanes, floods) and when repairing operations are not possible during hours or days because the area is not reachable by terrestrial vehicles, UAV swarms can, temporarily bring a solution for keeping the contact with concerned people.

In the domain of industry and more specifically for hazardous industrial sites, there is a crucial need of detect any abnormal situation or intrusion of non authorized people in a given zone. For large areas the deployment of many UAVs guarantees a short delay between the occurrence of any targeted event and the alert, improving the security of the site.

The MRS great capabilities, brought the motivation to solve the scientific and technical challenges to overcome the feasibility limitations for deploying such robotic systems. Challenges of paramount importance are coordination and communication issues.

In current scientific literature, coordination might be centralized or decentralized. In this thesis the focus will be toward aerial and ground mobile robots functioning in groups. The algorithms in literature could be categorized as centralized and distributed or decentralized. This thesis focuses on distributed/decentralized algorithms.

Distributed/Decentralized Systems

Distributed and decentralization of information flow among the group of robots grows attention day by day due to their great advantages over the centralized approaches. Some of the important distributed formation control properties are as follows:

- Scalability: The resources needed by a single autonomous agent do not grow with the increase of the total number of agents in the system.
- Fault tolerance: The system entities are resilient as there is not a single/central unit of control/failure.
- Concurrency: The process of (local) information is performed in a parallel and independent fashion in order to achieve a global goal.

Communication Issues

In literature as well most of the algorithms consider perfect communication between the robots within the group. For systems operating in large size areas, with sometimes harsh and changing environmental conditions, reliability of communications cannot always be guaranteed. Usually the full robot states are shared in between them, the task is shared as well. These information can be passed either in to a central point or as mentioned above in between the robots in a distributed/decentralized fashion. Some research study the issue of reducing the data being shared within the robots in the group.

The assumption of the presence of perfect communication is unrealistic. It is an issue that motivated this current study, leading to the development and test of algorithms that can be in charge of constructing robot formation, and navigating such formation in presence of such communication imperfections or even without communication utilizing the information that can be perceived solely by onboard sensors of the robots within the group.

When deploying MRS, the untackled question even in the recent literature, is the consequences of the group of robots losing their communications. This thesis proposes a line of research on acting as a recovery mode to retain formation shape using visual feedback inspired from how birds flock, form and reform their group with focus on possible solutions for coping with all these issues.

1.3 Thesis Contribution

The main focus in this thesis is toward the challenges that occur in coordinating the MRS in the presence of intermittent communication, or at the time of the complete communications dropout. While there are some studies in literature that rely on vision and other sensing modalities, they are not widely available and still quite challenging to execute. That was one of the motivations developed in this thesis, to tackle such challenges and propose solutions that can lead to multi robots navigate as intended. Also an important point was regarding absolute positioning challenge in MRS. Usual solutions rely on global positioning systems either indoor such as widely used motion tracking systems, or outdoor such as Global Navigation Satellite System (GNSS) i.e: GPS, GLONASS, Galileo, Beidou. That leads to most of the solutions being tested in constrained unrealistic environmental setup. Other scenarios where the GNSS availability is of concern like forests, signal jammed arena, etc, are also considered by this work.

In this thesis some solutions are proposed that can handle the MRS coordination relying on on-board sensing. That adds resilience to the system in unstructured situations when communications and absolute positioning are disturbed or unreliable.

Within that context, the main focus was a proposition of a control law for the a group of UAVs in order to make them flying in a coordinated manner. The first contribution is a high level controller for formation flight in the context of a leader-follower (LF) scheme. The presented controller presents some key features as its capability to self-adapt to intermittent communications between the robots of the group. This self-adaptation is obtained by compensating lost information by data stemmed from on-board sensors and mainly from visual feedback. With respect to existing literature, studies that were trying to solve this type of challenge with similar approaches, that first contribution was focusing on solving the problem with high level controller that can be tuned much more easier than the available algorithms.

The second contribution of this thesis is still concerned by formation flight following a LF scheme. But, the presented method works with neither communication between the robots, nor direct distance measurement. In such study the use of a well known visual control feedback framework was utilized. The exclusion of direct distance measurement, utilizing the image moments features for Image Based Visual Servoing (IBVS). It adapted for the need of the particular kinematics of the robots. A classical extended Kalman filter (EKF) is used in that context, particularly for its predictive component to compensate for the non communicated information about the leader robot.

For the last contribution, we were interested in the problem of the alignment of only three machines. There were two main constraints: the machines were unable to communicate with each-other and they were unable to measure distance between them, directly or even indirectly. For solving the problem, we designed a decentralized algorithm using only bearing angles between robots. We show, by a theoretical proof, that the probability to reach the alignment is close to 1, but decreases when a non zero security distance between the robots is imposed.

1.4 Structure

This thesis is distributed as follows, chapter 2 is discussing the State Of The Art (SOTA) in the MRS and swarm robotics with focus on decentralized methods. Each particular method of the thesis are having their own particular SOTA.

Chapter 3 represents the first contributed approach toward the decentralized algorithm proposed to control a formation of multi aerial robots. Chapter 4 is proposing the second approach where, the visual feedback is formulated in a different way allowing the formation to not rely on distance or displacement measurements. While chapter 5 is showing a higher level approach where there is no leader in the MRS, and only bearing measurement is used to form a line out of three robots. All the methods available in chapter 3, 4, and 5 have the theoretical formulation and simulation results. Chapters 4, 5 are corroborated with practical experimental results.

Chapter 6 is providing a general conclusion and perspectives of this thesis.

CHAPTER 2

State of the art

Contents

2.1 Introduction

Multi-Robot Systems (MRS) have been widely studied since the 90s [2], [3]. Many advantages can be expected from using a group of robots: increase and share of the payload [4], reduction of the time needed for the achievement of a task [5], fault tolerance and resilience of the system [6], [7], or use of simpler and cheaper robots for adaptability to the environment [8].

However, making a group of robots evolving together in the same environment entails the resolution of some problems. Coordination is one of the most critical ones, especially considering decentralized settings. Coordination between the members of the group can be achieved through various ways, behavior or force-based methods, Virtual Rigid Body (VRB) structure building, or Leader-Follower (LF) schemes.

According to the classification of [9], the interactions of a group of entities can be of several types: collective actions, collaboration, cooperation or coordination. Additionally another type of interaction can be competition within .

According to that classification, the main interest of this thesis is the coordination type, when robots attempt to form a given pattern or attempt to fly according to a given formation.

There are several perspectives for addressing the coordination problem. One of which is the control engineering perspective, where formation control is taking charge of such systems. Another perspective is the algorithmic, where swarm engineering is investigated. There are approaches such as flocking and others that is addressing such problem.

As mentioned in the introductory chapter, research on multi-agent systems, and in particular multi-robot systems (e.g., ground and aerial mobile robots), has achieved many advances over the last decades with a number of theoretical and experimental accomplishments. In other terms multiple vehicles, including unmanned aerial vehicles, unmanned ground vehicles and unmanned underwater vehicles have been studied by many communities, through different perspectives. Many challenges were overcame and much more were introduced in their place. Both technological advances and fundamental theoretical maturity in multiple disciplines of science, have contributed for such achievements.

Among many topics that were discussed in the literature, consensus, formation control, optimization, task assignment, and estimation; are of great importance and interest.

From technological side, onboard sensing, communication and computing capabilities, have opened new frontiers for scientific methods to be validated and tested [10, 11, 12, 13, 14]. For guaranteeing real missions success, the team of robots should preserve specifications and abilities, either similar or different from one robot to another. This can be sensing or actuating capabilities. Communication property in between the robots is of great importance to share information, sensing values, robot states, and commanded control values. The communication can be centralized or decentralized considered as one or multi hop. Some other algorithms rely only on the sensing network in between the robots. The later algorithms are harder to be achieved. Other algorithms are taking into consideration all the realistic aspects of communication range, sensing range, uncertainty encountered in these processes, as well as delay in computation and processing. In the survey [15], a list of communication approaches were discussed. Notably the survey showed in the context of MRS, different routing algorithms, and protocols. As well as the extension between the classical communication approaches and cloud servers. Either in communication or sensing network, connectivity in between mobile robot group is considered vital for mission success and control convergence. The problem of maintaining this connectivity is of great importance as well, and it carry good added value in the recent literature.

Another category of common questions are addressed lately, like having a global frame of reference or common frame of reference or not at all. Consensus over controllable properties of the robots in the fleet like synchronizing velocities, which is commonly known as velocity alignment or consensus. Another question is addressing a rendezvous problem with objective to let the robots in the formation meet in a point, go in a certain direction toward a goal or having a reference trajectory. Another definition could be the formation cohesion, where each agent tries to stay as close as possible to the nearby robots. This can be achieved by a controller that generates an attraction force

toward the neighboring robots. There are methods that are also popular in literature known as flocking algorithms. An exhaustive review can be found in [16].

Solutions proposed for the formation control can be categorized as centralized, decentralized, distributed from the perspective of computational decision making. Sometimes in the literature distributed and decentralized are interchangeably used. This arises due to the fact of the multi-disciplinary research fields, that the researchers are coming from. The problems are studied by pure computer science, control, information theoretic and robotics communities of researchers. Solutions are offered taking into consideration full dynamics robots with tight methods working on specific platforms, others are offered based on the kinematics of the robots or the most simplified robotics model of double integrator systems dealing with the robot as a point in the space.

In many tasks preserving a geometrical shape or properties like keeping a distance and angle in between the robots while navigating the environment is of great importance. This can act as naturally inter-robot collision avoidance and can give access to a lot of control convergence tools to ensure mission accomplishment. This emphasises the importance of the formation control problem which will be elaborated more in the next section.

2.2 Formation Control

Formation Control for teams of multiple mobile robots is a well discussed problem in literature. The controllers are trying to keep a desired relative positions and/or angles between robots in the group. Achieving the formation requires the solution of two problems: sensing of the relative pose and regulation of the relative pose to desired values. Also commonly known as formation realisation and formation maintenance.

Formation control can have different implementation through different architectures contingent upon scenario conditions. There will be a balance between requirements and limitations. Depending on the sensing and actuating capabilities, and the interaction topology of robots in the group, a variety of challenges have been studied in the literature which will be discussed in this chapter. Some renowned implementations of formation control are the leader-follower and flocking paradigms.

The system can be defined by prescribing a desired geometric shape or pattern with distance and bearing angle values. Then the controller will have one of two main tasks, either formation creation from arbitrary placed robots in an arena, or keep a rigid formation while navigating the body in cohesive form.

The paper [17] have presented a survey of spacecraft formation flying rather than an extensive survey of general multi-agent systems. Most of the results from the papers, are communication agnostic. But it has common fundamentals with the general MRS. Also they show well studied examples dealing with the commands delay, fault recovery routines and many more important building software blocks for real world systems.

The book [18] gives mature theoretic insight about the multi-agent networks with robotics perspective. Good graph theory notions is well defined. Formation control and distributed estimation is studied for mobile robots. A link between graph theory and linear algebra, which is commonly known as algebraic graph theory is well mentioned as well.

Several surveys and reviews were also part of the literature in the past decades. Survey of forma-

tion control of multi-agent systems are found in [19]. A review [20] on distributed multi-agent systems covers seminal contributions and results in the area of consensus, formation control, optimization and estimation.

Several approaches apply artificial attraction/repulsion fields as a building block to establish forces for each robots within a formation to preserve a shape. Others use gradient based controllers.

Coordination for motion control, like cooperation is a major challenge as well, the book [21] discuss many of them thoroughly. It shows several methods for path planning synchronisation, adaptive controllers for velocity matching and passivity based controllers as well. Formation problems for UAV swarm systems with nonlinear dynamics were addressed in [22]. The methods were tested on a group of ground mobile robots with omni directional camera onboard. It heavily relied on the dynamic model of the mobile robot, while this part shows that the method is not flexible to be transferred to other types of robots, the authors justify their point of view of not working on the linearized robot models. In the sequel, a brief definition of the general formation control architecture from the information flow point of view, is addressed.

Centralized control architecture

In this architecture, robots in the group are controlled by one centralized controller. This controller senses the states of all the robots, then command the corresponding control signals to all of them. Figure 2.1 illustrates the centralized control architecture. This controller can be commanded from a ground base station connected to all the robots in the group or on the leader robot in the formation. Literature specially before the latest decades is full of methods in that investigated this architecture in thorough detail.

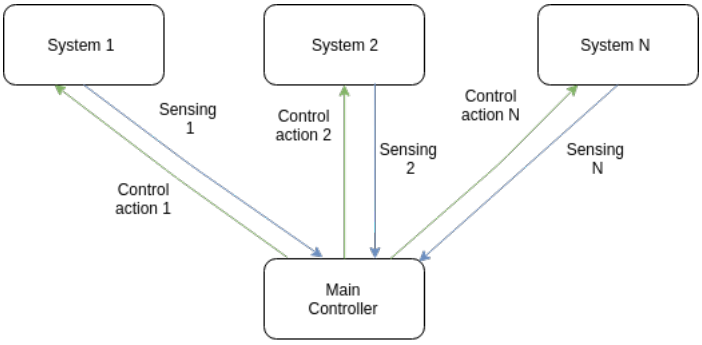


Figure 2.1: Centralized control architecture.

Distributed control architecture

Interaction between the robots in the team is happening through sending some of the sensing or the desired control setpoint (but not the control commands). It can be visualised in figure 2.2. Here in this architecture a huge reduction in the communication bandwidth is achieved, with the setback of having more complicated controllers to be implemented by researchers and engineers. Moreover, this

architecture is more resilient and fault tolerant than the previously mentioned centralised one. It has attracted researchers for many decades in the robotics community.

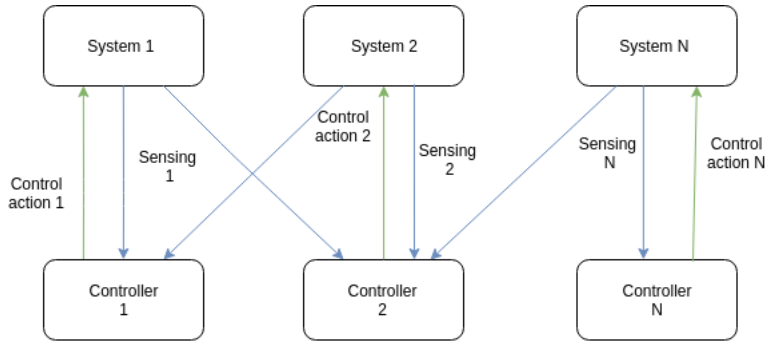


Figure 2.2: Distributed control architecture.

Decentralized control architecture

Similar to the previous architecture, the decentralized one is doing team interaction through interaction between robots in the team. The major difference is the less reliance on the sensing measurements from other robots in the group. Only pure local sensing is used to control the robot with communicating the desired formation task or without communication through the predefined artificial intelligence rules embedded onboard. Figure 2.3 shows more clearly the separation of interest property in the architecture.

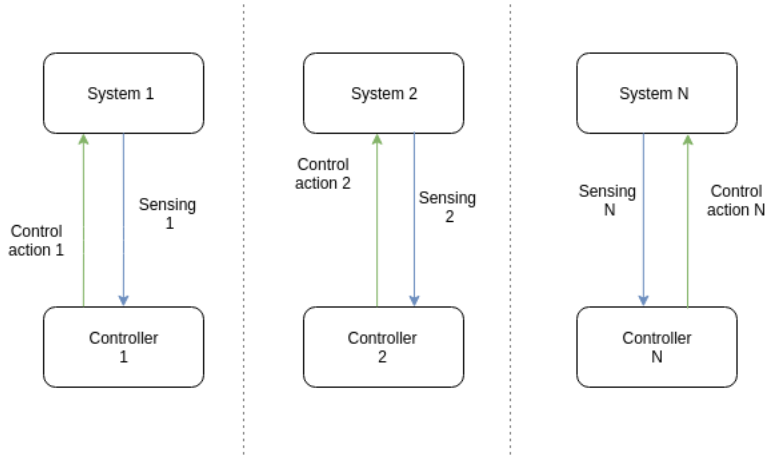


Figure 2.3: Decentralized control architecture.

Another prespective away from the sensing controllable states of the robots is by the task allocation, can be viewed in figure 2.4.

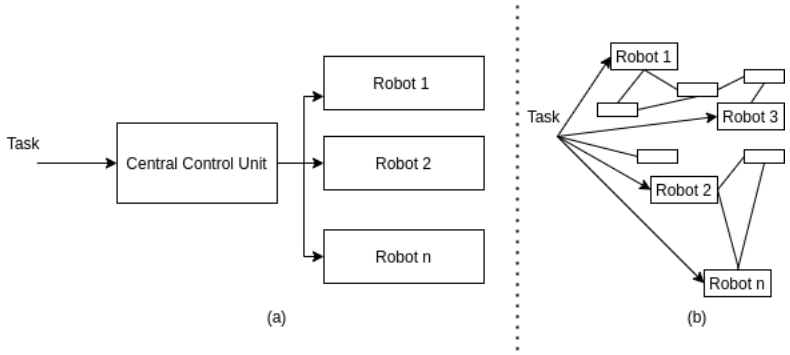


Figure 2.4: Task allocation in Centralized (a) and Decentralized (b) architectures.

2.2.1 Methods in Literature

Leader Follower

Some researches in the multi-robot systems society focuses on Leader-Follower (LF) approach where, at least one robot is considered as the leader of the group and the rest of the robots are considered as followers. The followers track the position and/or orientation of the leader with some desired distance and bearing. The leader robot can be commanded manually or follow a predefined trajectory. Multiple leaders are also used in literature. Leader can either be predefined to the mission or elected dynamically.

The vision based leader follower formation control paper such as [23], implement algorithms that are based on input-output linearization of the robot model and require the estimation of the relative angle between leader and followers. The leader's linear and angular velocities are estimated as well without explicitly transmitting those values to the follower robots. A decentralized LF for a group of ground mobile robot with markers attached vertically on its board was realised in [24].

In [25] a sliding mode controller is proposed to control the translational dynamic of the robots and a linear PD controller provides the desired orientation for the UAVs in the group. A control law is designed to minimize the errors according to x,y and yaw to converge to zero even in the presence of uncertain measurements, sensed values, and inaccurate controller. The control inputs were the followers velocities. The major drawback is utilising the full dynamics of the quadrotor, limiting the formation control to this specific type of robots with very specific dynamics. Attitude stabilization was taken into consideration for the formation stability. This very sensitive controller to be extended to other platforms or other types of mobile robots. Even implementing on the same type of robot with similar hardware and software architecture, will require substantial parameter tuning.

Virtual Structure

The main goal in this family of schemes, is to construct a desired virtual formation shape. Each robot in the group has its own planned trajectory. This can be planned in a centralized, distributed or decentralized manner. They can as well be planned completely offline a priori to the mission, or can have a reactive behavior during the mission. Generally, no interactions between robots are considered.

This method is widely used in formation control for spacecrafts like [26], where the use of Virtual Spring damper mesh was proposed as a structure maintaining controller. Examples of experimental works of virtual structure control strategies can be found in [27] and [28].

This structure exhibits some drawbacks that could concern mission success. The nature of the controller where there are no interaction between robots in the formation could lead to collisions in the presence of perturbations. Most of the solutions proposed in literature was centralized control, except for minor papers where [1] is the most prominent one. The authors propose distributed trajectory planning for a group of multi aerial robots in 3D space. The major drawback is its reliance on global positioning system and common ground frame of reference. In the paper a simple formation switching mechanism was as well developed. Figure 2.5 is showing the visualisation of the realistic experiments mentioned in the paper.

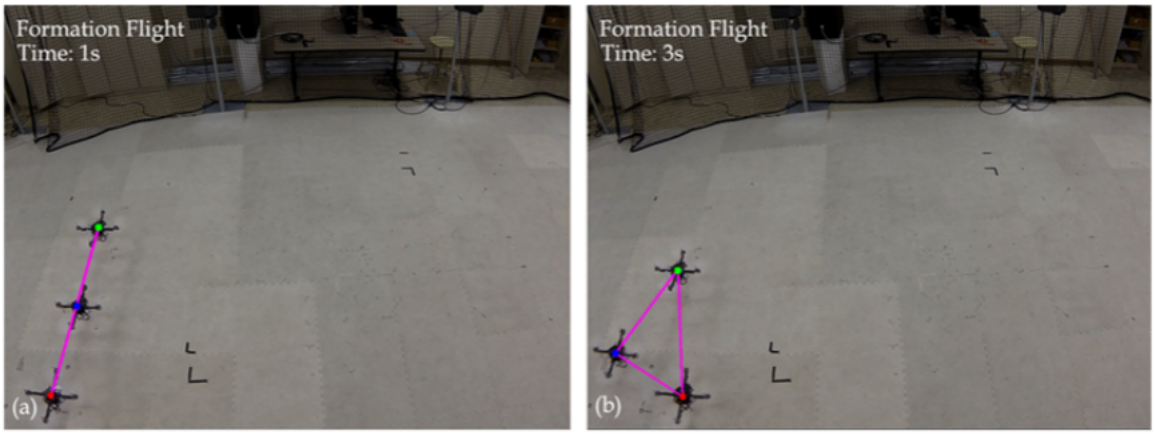


Figure 2.5: Virtual Rigid Body visualisation in (a), and Formation switching in (b) [1]

2.2.2 Multi Robot Systems Rigidity

Since virtual structure and formation control are of great advantage, it was very important to formally study the system rigidity, which is the case of this section. The multi robot control literature can have the following taxonomy based on the sensing and measurement utility; Position-based control, Displacement-based control, Distance-based control and Bearing based control.

- Position-based control: Robots can sense and have access to their own global positions with respect to a common coordinate system. They sense and command their own positions to achieve the desired formation, which is calculated with respect to the global coordinate system. This can be mostly considered as centralized scheme.
- Displacement-based control: Each robot is able to sense relative positions of its neighbouring robots with respect to the common global coordinate system. The desired formation is specified by desired displacements for each robot in the group. Robots command the control signals as displacements of their neighbouring robots to achieve the desired shape. There is a noticeable

feature in this type of control which is, robots need to align their orientations globally, sometimes with respect to a common reference frame. However, the robots does not require to know its absolute pose with respect to a global coordinate system. The formation is well described in the displacement interaction in between the robots. It is depending on the implementation to consider this scheme as centralized or decentralized.

- Distance-based control: Inter-agent distances are actively controlled to achieve the desired formation, which is given by the desired inter-agent distances. Individual robots are assumed to be able to sense relative positions of their neighbouring robots with respect to their own local coordinate systems. The orientations of the robot's local coordinate system are not necessarily aligned or matching.
- Bearing-based control: Inter-agent presence and connectivity of the graph is the main concern but not the exact distance. This can be sub categorized in angular bearing rigidity or distance bearing rigidity. The later use at least one distance to estimate the scale of the formation in the spatial space.

Except the last control scheme, there is always a trade-off between the amount of interactions among the robot's group and the sensing capabilities of a singular robot. Some confusion also happens in the relatively older papers (2005 and older papers) when developing or mentioning bearing based control. They rely on at least one distance or direct measurement. That is why in the more recent publications the term Bearing-only formation control is emphasized. Formation control using distance-only measurements can be found in [29]. In order to achieve such desired shape by the robots a popular gradient-based technique for formation-control in both of the two popular variants: position-based and distance-based can be very obviously found.

In the position-based strategy the robots control the relative positions (vector quantities) between them, whereas in distance-based strategy the robots just control the distances (scalar quantities) between them.

Rigidity Theory

As mentioned above coordination problems in multi-robot systems are vital for the stability of the formation. The coordination is highly dependent on the sensing and communication mediums available. Noise and uncertainty affect the performance of one robot operation. In the system of systems, as in the case of MRS, the effect is more impactful.

Graphs can encode behaviors that encounter interaction geometrical structures by abstracting the information exchange that those systems use. Examples of these behaviors can be regularization of geometrical patterns, synchronization, tracking robots, modelling noises and uncertainities in the system.

Rigidity of the formation is important in order to accomplish many of the assigned tasks. The purpose of this tool is to help to achieve the goal of keeping a desired shape of a multi agent formation. Rigidity graph theory plays an important role in the design of a desired rigid shape for the formation of the robots. The graph rigidity may sometimes be mentioned in the literature as dimensional bar-and-joint framework.

The theory which is well described in [19] has proven to be an effective theoretical framework for the analysis and resolution of formation control problem and cooperative localization problem as

well. There are several methods that rely on local sensing only such as relative distance, can be found in [30, 31, 32, 33].

In the following paper, the authors propose a solution that can infer heading information of the robot from the measured angles with neighbours [34]. Showing a comparison between bearing formation instead of distance based formation, this paper at its time was focusing on network topologies of robots in 2D space with the assumption that at least, there exist one node that knows the global coordinate system. This information is then communicated to the other nodes/robots in the formation.

The previous paper was advanced in [35] by proposing a gradient based control law depending on bearing-only constrained graph rigidity and parallel drawings. However, the proposed controller was still practically requiring both bearing and one distance measurements in order to be implemented. The distance measurement was required for the purpose of scale disambiguation. That drawback was addressed and enhanced theoretically in [36]. Though the authors shows only simulation results; they took into consideration primitive sensor model and the actual measurement conditions, like noise, inconsistency,etc. So they discussed the use of only the bearing measurements in their control law approach. Their main contribution can be summarized in analyzing the connection between scalability, minimality and rigidity. They have considered the formation control of planar kinematic robots by assuming a common reference frame and an undirected topology for the measurement graph (i.e., all measurements are assumed to be reciprocal)

In [37], authors introduced a decentralized bearing formation controller that does not require presence of a common reference frame. The formation is defined as an undirected sensing topological graph by exploiting the (body- frame) measured bearings and a single distance measurement among an arbitrary pair of robots. But they have a special requirement and assumption which is structure for the chosen measurement graph which, among others, must contain two special robots able to measure and to be measured by any other robot in the group.

Cooperative localization from local relative measurements in the absence of central sensing and processing units is achieved in [32], where every robot is only able to obtain relative measurements and communicate with a certain subset of the whole group. rigidity of the formation is a necessary requirement for recovering from the available relative measurements a consistent solution of the localization problem in a common shared frame.

This assumption of having reciprocating measurement modality was generalized in [10]. Where they proposed a fully decentralized bearing formation controller that only requires presence of a generic (directed) bearing rigid topology in the beginning of the formation.

In the previous papers it was important to get at least one distance measurement even if the controller was concerned with bearing measurement only. The sole reason behind that is to achieve the scale of the formation shape.

This can be solved by measuring the distance by any means, for instance, through depth sensor or LIDAR onboard of the robots. Another solution is to infer the depth through the structure from motion techniques. This work was mentioned in [11] with some physics based simulation results with camera model included. In that paper one major assumption was taken into consideration, which is presence of (at least) one pair of robots in mutual visibility; which is some kind of reciprocating assumption.

The drawback here is that they do not mention formation self scale knowledge. Also they didn't take into consideration collision avoidance between the robots. We can eliminate the limited field of

view constraint by deploying an Omnidirectional image sensing or possibly a pan tilt unit (gimbal).

In the paper [12] partially centralized method was proposed, where a linear program is solved at every step of motion in order to mix the derivative of the second smallest eigenvalue of a weighted Laplacian (known as algebraic connectivity, or Fiedler eigenvalue) and the k-connectivity of the system. They were considering the time-varying interaction graph. This second smallest eigenvalue has an important role in assuring the maintenance of the formation rigidity and will be elaborated more later.

Bearing Based Formation Control

These reciprocal bearing measurements which was discussed in [13] can be achieved by having an omni-visual information system. They consider a simple sensing model in which each robot can only measure the angle, relative to its own orientation, to neighbouring robots.

In this apprehended paper, the authors consider a weak sensing model where each robot is only capable of measuring the angle, relative to its own heading, to each of its neighbour. By this way they achieve a scale-free coordinates as an alternative coordinate system for multi-robot systems

Parallel rigidity used in the paper [14, 35] is another way to address the bearing rigidity. The reason behind is the parallelism between vectors required to guarantee that the graph is bearing equivalent and congruent. These are two fundamental features of the graph that can guarantee its rigidity.

An advance has been achieved, by proposing a decentralized formation controller that uses both bearing-only measurements and constraints (does not rely on distances) [36]. The authors draws an analysis for connecting between scalability, minimality and rigidity of the robotics networked graph.

According to the paper [38], generic minimal rigidity methods were developed for non robotics purposes. The paper was mainly motivated by some problems happening in Computer Aided Design (CAD) field. There work regarding spherical bar-and-joint structure is of importance for what is considered in this section of bearing based rigid formation control.

In the following paper [39], a distributed bearing-only formation control law which ensures local exponential or finite-time stability is being proposed. They only rely on the local bearings measurements between the robots without communications in between them. The controller was also tested in simulation. An important comparison between distance based rigidity and bearing based rigidity was also studied.

The paper [40] from GRASP laboratory shows that measurements of the bearing angles between the robots are sufficient for reaching a balanced circular formation. They demonstrate their finding by implementation on a 2D planner mobile robots.

The paper [41] uses angle only constraints to achieve triangular formation in a distributed control fashion. Their drawback is the limitation to three robots only through all the paper and approach. Also they were addressing the formation of triangle shape only. Another example of triangle formation can be found in [42]. That was inspiring for some other papers and was advanced and mentioned before in [36]. Also an advancement is made in the paper [43], where the controller is only relying on local information. They as well draw a comparison between distance based, bearing based and mixed rigidity controller. Limited results were shown for the problem of time varying control commands. Since it was only in simulation, it can be acceptable not to be elaborated more.

Maintenance of formation

One key factor of ensuring a successful mission accomplishment of a swarm robots is to keep connectivity of all the robots in the formation.The issue of maintaining/preserving rigidity of the formation (which is a global property like graph connectivity) during the robot motion despite the possible presence of sensing constraints. Moreover, in a dynamic uncertain environment, a team of robots may not be able to maintain the same communication and sensing graph throughout the duration of a mission. Another problem such as, occlusions that appears while robots are moving can also be troublesome. Sensing range limitations should also be incorporated in the control scheme, so the links between robots drop out should be modeled. Furthermore, collisions with obstacles should be avoided during motion. Convergence of any formation MRS control scheme run by each robot in the group is dependent upon the formation rigidity.

The maintenance parameter was taken into consideration in 2D world in simulation only in [44]. Then it was extended in [45]. Where it was experimented with realistic quadrotor flying robot. In which they implemented a controller that consists of a decentralized gradient descent action based on a suitable "degree of bearing rigidity" which is directly related to the spectral properties of the bearing rigidity matrix. This bearing rigidity matrix was mentioned before with more details in [10]. For every robot there is a weighted body-frame rigidity matrix representing connectivity. The strategy followed to preserve infinitesimal bearing rigidity of the formation by ensuring fulfilment of the rank condition for the weighted bearing rigidity matrix. They have added 3 constraints to be taken into consideration. As mentioned (C1) minimum/maximum range, (C2) limited field of view and (C3) occluded visibility.

Connectivity Measurements

Connectivity maintenance of a graph is of vast importance to ensure mission success as well as formation readiness to be controlled. There are several methods to do so and can be categorized in

- Conservative methods: which aim at preserving the initial graph topology during the task [46, 47, 48]
- Flexible approaches: which allow to switch anytime among any of the connected topologies as in [49]. In this approach, a suitable recasting of decentralized connectivity maintenance based on the second smallest eigenvalue of the graph Laplacian matrix was realised.

2.2.3 Formation Evaluation and Analysis

Benchmarking in robotics is challenging, taking into consideration the uncertainty, noisy, system-specific nature of the field.

Formal metrics for MRS cooperation, formation and system performance, as well as for grades of cooperation, are noticeably missing from the literature. Standards are not very common among the field. Some metrics like collaborative and absolute localization can recently be measured and get quantifiable distinguishable results.

2.2.4 Stability notions

The global stability of the proposed formation control laws was proved by employing the Poincare -Bendixson theorem as mentioned in [41, 42, 50].

Problems that are concerned with this challenge can be summed up in stabilization of infinitesimally rigid formations paper [30]. The study [51] shows early results regarding stability analysis for leader and follower approachs in MRS. In [52] a exhaustive survey about stability and efficiency for networked formation is delivered. This can be considered an abstraction for the MRS networked formations.

When position measurements are available for formation control, a globally stable control law has been proposed in [53] to stabilize formations in desired dimensions with fixed topology.

Convergence in finite time is important as well for formation control for MRS. The robots should be steered or aggregated in definite time in several scenarios and cases to show that the controller is functioning efficiently.

2.3 Swarm, Algorithmic Methods for Pattern Formation

In this section, an overview is given, over the higher level algorithms that take place in the MRS domain inspired from Multi Agent System(MAS). MRS society gets inspiration from the MAS methods taking into account the hardware and the environment that such systems are going to be deployed.

2.3.1 Consensus

The main challenges with consensus based control laws can be narrowed down that all the robots should have a common sense of direction since the input biases are defined in a common reference frame. In really such common reference frame as discussed is not practically viable all the time.

Many formation control strategies have been designed based on consensus protocols, for example as in [54, 55].

Surveys on consensus based formation control can be found in [56, 57]. In these surveys, there are many studies toward the cooperative control of different vehicles. Some papers study the effect of the kinematics on the global desired goal, some papers study the collision avoidance, and many more aspects.

In literature, there exist some papers that handle the challenge of formation control or swarm of a group of robots under the influence of communication pitfals. Some papers model the information flow between the robots as a network with time delay such as [58].

Such paper focuses on the idea of having consensus control without having a leader robot in [59]. This leaderless control is providing a more emerging intelligent performance.

To this end regarding consensus controller, there is an obvious gap as in the other methods when dealing with the presence or not of communication channels among robots funcioning in a group,i.e: MRS.

2.3.2 Multi Robot Path Planning and Task Allocation

Another very important perspective in the literature of robotics is the mobile robot path planning. Normally it is a layer of planning above the low level control layer in the robot's architecture. This can give the system artificial intelligence from a higher level point of view of the scheme. It is as well of vital importance in MRS and widely known as multi robot path planning. In this scope, path coordination method essentially plans paths by scheduling the motion of the robots in the group. This is dependent on many factors such as, robots initial configuration, space-time resources, robots motion and sensing models, desired configuration, etc. Since this is not the main focus of this work, but for the sake of completeness, hereby in the following article [60], a recent study of the state of the art for multi-robot task allocation.

CHAPTER 3

Decentralized High-Level Controller for Formation Flight Control of UAVs

Contents

It is proposed in this chapter, a design of a decentralized and tuning-less high level controller able to maintain without tracking errors a Leader-Follower (LF) configuration in case of lack or degraded communications (latencies, loss...) between the leader and followers UAVs. The high level controller

only requires simple tunings and rests on a predictive filtering algorithm and a first order dynamic model to recover an estimation of the leader UAV velocities and avoid the tracking errors.

Keywords: Distributed/Decentralized control, Formation control, Leader-Follower, Autonomous vehicles, UAVs

3.1 Introduction

Coordination in multi-robots systems is a crucial issue that may be addressed by different approaches: behavior-based or force-based methods, Virtual Rigid Body structure building or Leader-Follower schemes. In the present chapter, we investigate the problem of building and maintaining a Leader-Follower (LF) configuration in case of degraded communications between the leader and the followers. The question of building and maintaining such a formation without communication has been addressed many times during the last two decades and most of the time the proposed solution relies on a sensor-based mechanism for compensating the absence of communication. Very often, work are considering cameras; pan controlled camera [61] or omnidirectional camera like in [62], [63], or other types of sensors like RGB-D (Kinect) [64]. From the data obtained by the sensors, each follower of the formation attempts to estimate the pose of the global or of its local leader. In [61], the authors consider this problem with a pan-controlled camera. Since no communication occurs, the leader's velocities are unknown. The goal of their work is the design of both an adaptive formation controller and an adaptive camera controller. The authors in [63] proposed a method for LF formation control considering a robotic system in which each follower is equipped with an uncalibrated omnidirectional or perspective camera. No communications are allowed and the velocity of the leader is considered unknown as well. In that work the authors present an adaptive estimator based on several feature points. In [64], the goal is to build and maintain a LF formation without communication between the robots. Each robot is equipped with only one type of sensor, a Kinect, for estimating the pose (relative orientation and position of the leader). The choice of the Kinect is justified by the fact that it provides color and depth information from which can be directly deduced angles and distances measurements but it suffers from a limited field of view. The controller depends on both the visibility maintenance and the minimization of position errors. With respect to these previous works, our main contribution is the design of a decentralized and tuning-less high level controller able to maintain a LF configuration. The proposed controller combines an omnidirectional camera with filtered and predicted measurements and a first order dynamic model of the UAV to get the leader velocities (even in case of degraded communications (latencies, loss) between the leader and followers UAVs) and avoid the tracking errors. Practical experiments are delivered utilizing perspective camera, but the algorithm is generic to work on the rectified images of an omnidirectional camera.

3.2 Problem Statement

The goal is to construct a geometrical shape sometimes known as Virtual Rigid Body (VRB) in order to establish the desired formation. The virtual structure between several robots aims at facilitating the agile control of several multi-rotor formations. The goal of each follower UAV is to follow the leader by keeping predefined separation distances X_{SP1}, Y_{SP1}, Z_{SP1} as illustrated on Fig. 3.1. The leader

UAV is assumed to track perfectly a virtual target which moves along a defined trajectory (straight line, circle, etc.). The controller of the leader UAV is not described in this work. The description of the high level controller will be done for the follower UAV number 1. The vision-based system provides the distance (d_1) and the bearing angle (θ_1) of the leader UAV.

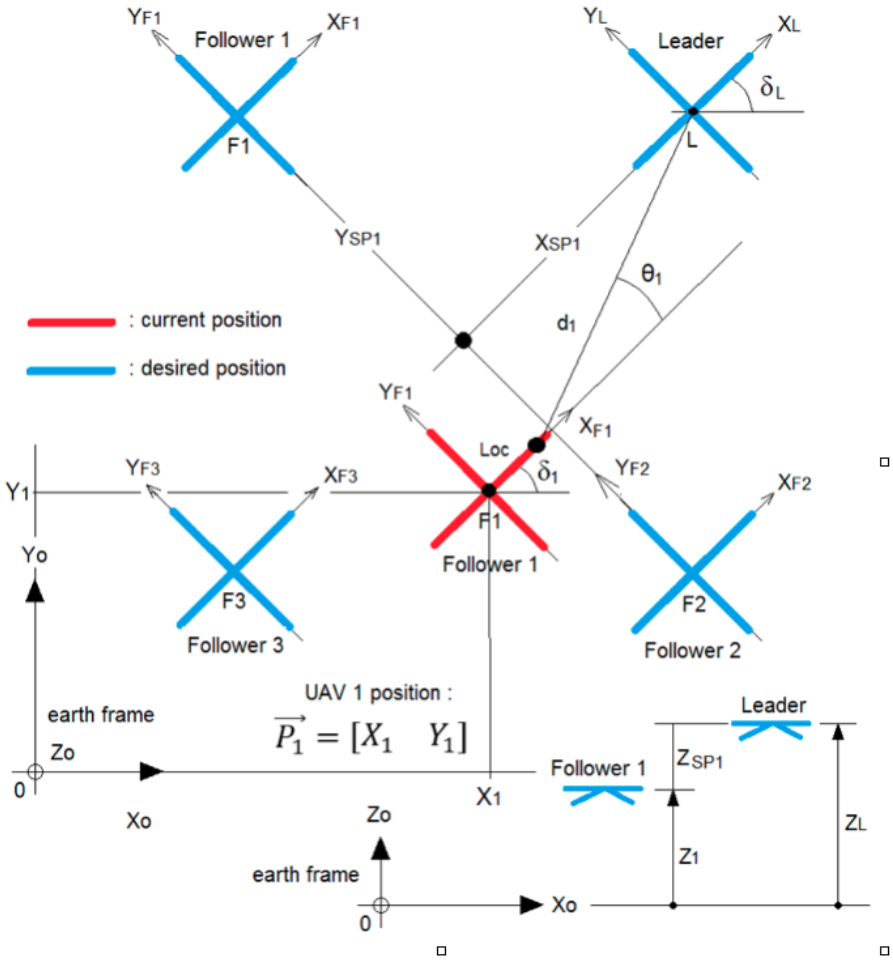


Figure 3.1: Frames relations between Leader and follower UAVs. As well as reference to the ground.

3.3 Design of The High Level Controller

3.3.1 Double Exponential Smoothing (DES) algorithm

The proposed method assumes the availability of an estimation of both the distance (d_1) and of the bearing angle (θ_1) whose measurements are naturally degraded by noise and other uncertainties. A

filtering stage to overcome these problems is mandatory. Such an algorithm has to be fast, accurate, robust to different motions, and easy to implement. We assume that the leader UAV moves regularly over time with a trend. In [65], a comparison is drawn between Double Exponential Smoothing (DES) algorithm and Kalman filter. The authors show that DES algorithm runs faster than Kalman filter with equivalent prediction performances and simpler implementations. That is why DES algorithm can be a good choice to generate smoothed estimates and forecast of distance (d_1) and bearing angle (θ_1). The DES algorithm at time instant $n.Te$, where Te is the sampling period and n is the discrete-time index, implemented for both the distance (d_1) and bearing angle (θ_1) measurements is given as follows:

$$\hat{d}_{1n} = \alpha_{d_1} \cdot d_{1n} + (1 - \alpha_{d_1}) \cdot \left(\hat{d}_{1n-1} + b_{d_{1n-1}} \right) \tag{3.1}$$

$$b_{d_{1n}} = \lambda_{d_1} \cdot \left(\hat{d}_{1n} - \hat{d}_{1n-1} \right) + (1 - \lambda_{d_1}) \cdot b_{d_{1n-1}} \tag{3.2}$$

$$\hat{\theta}_{1n} = \alpha_{\theta_1} \cdot \theta_{1n} + (1 - \alpha_{\theta_1}) \cdot \left(\hat{\theta}_{1n-1} + b_{\theta_{1n-1}} \right) \tag{3.3}$$

$$b_{\theta_{1n}} = \lambda_{\theta_1} \cdot \left(\hat{\theta}_{1n} - \hat{\theta}_{1n-1} \right) + (1 - \lambda_{\theta_1}) \cdot b_{\theta_{1n-1}} \tag{3.4}$$

where:
d_{1n}, θ_{1n} are the values of d_1 and θ_1 at n^{th} sample,
$\hat{d}_{1n}, \hat{\theta}_{1n}$ are the smoothed values of d_{1n} and θ_{1n},
$b_{d_{1n}}, b_{\theta_{1n}}$ are the trends values of d_{1n} and θ_{1n}

$$\tilde{d}_{1n+m} = \hat{d}_{1n} + m.b_{d_{1n}} \quad \tilde{\theta}_{1n+m} = \hat{\theta}_{1n} + m.b_{\theta_{1n}} \tag{3.5}$$

The initial values given to $\hat{d}_{1n}$, $\hat{\theta}_{1n}$, $b_{d_{1n}}$, and $b_{\theta_{1n}}$ are:

$$\hat{d}_{11} = d_{11}, \hat{\theta}_{11} = \theta_{11}, b_{d_{11}} = d_{12} - d_{11}, b_{\theta_{1n}} = \theta_{12} - \theta_{11} \tag{3.6}$$

Usually, α is called the data smoothing factor and λ is called the trend smoothing factor. A compromise has to be found for the values of α and λ. High values make the DES algorithm follow the trend more accurately whilst small values make it generate smoother results. For the follower UAV 1, the filtered equation 3.7 and predicted equation 3.8 for both distance errors (x, y axes) are the following ones as illustrated in figure 3.1:

$$\hat{e}_{1x} = \hat{d}_1 * \cos\left(\hat{\theta}_1\right) - X_{SP1} \quad \hat{e}_{1y} = \hat{d}_1 * \sin\left(\hat{\theta}_1\right) - Y_{SP1} \tag{3.7}$$

$$\tilde{e}_{1x} = \tilde{d}_1 * \cos\left(\tilde{\theta}_1\right) - X_{SP1} \quad \tilde{e}_{1y} = \tilde{d}_1 * \sin\left(\tilde{\theta}_1\right) - Y_{SP1} \tag{3.8}$$

3.3.2 Design of the tuning-less high level controller

We assume that the low level controllers (inner loop) implemented in the UAVs allow them to reach the desired references velocities (according to x and y axes) and the desired references positions (according to z axis (altitude, yaw angle)) with a short response time and without static errors, damping or oscillations.

All the UAVs are assumed to fly at the same altitude with the same yaw angle (altitude and yaw control can be controlled separately and are not detailed in this work). The control objective is to regulate the distance errors as follows:

$$\lim_{t\to\infty} \hat{e}_{1x}(t) = 0 \quad \lim_{t\to\infty} \hat{e}_{1y}(t) = 0 \tag{3.9}$$

To fulfill this control objective, we propose the following error dynamics equation:

$$\begin{bmatrix} \dot{\hat{e}}_{1x} \\ \dot{\hat{e}}_{1y} \end{bmatrix} = - \begin{bmatrix} A\left(\hat{e}_{1y}\right) \cdot f\left(\bar{D}_X \cdot \sum_{i=1}^{3} \beta_{xi}, \bar{V}_X - V_{Lx}, \bar{D}_X\right) \\ f\left(\bar{D}_Y \cdot \sum_{j=1}^{3} \beta_{yj}, \bar{V}_Y - V_{Ly}, \bar{D}_Y\right) \end{bmatrix} = R \cdot \vec{V}_L^T - \vec{V}_1^T \tag{3.10}$$

Where: $R = \begin{bmatrix} \cos\left(\delta_L - \delta_1\right) & -\sin\left(\delta_L - \delta_1\right) \\ \sin\left(\delta_L - \delta_1\right) & \cos\left(\delta_L - \delta_1\right) \end{bmatrix}$,
and $A\left(\hat{e}_{1y}\right) = e^{-Ay \cdot |\hat{e}_{1y}|}$

A_y is an attenuation parameter which makes decrease the effect of the linear control (x axis) when the lateral error (y axis) is not null, and these terms are defined as follows:
$\overrightarrow{V_L} = [V_{Lx} \quad V_{Ly}]$ is the leader velocity vector.

- $\beta_{xi} (\in [-1 \ldots +1])$
- $\beta_{yj} (\in [-1 \ldots +1])$
- $\beta_{x1} = f\left(\frac{e_{x1}}{\bar{D}_X}, 1, 1\right) \quad \beta_{x2} = f\left(\frac{e_{x2}}{\bar{D}_X}, 1, 1\right) \quad \beta_{x3} = f\left(\frac{e_{x3}}{m \cdot Te \bar{V}_X}, 1, 1\right)$
- $e_{x1} = \hat{e}_{1x} \quad \dot{e}_{x2} = \frac{\tilde{e}_{1x} + \hat{e}_{1x}}{2 \cdot m} \quad e_{x3} = \tilde{e}_{1x} - \hat{e}_{1x}$
- $\beta_{y1} = f\left(\frac{e_{y1}}{\bar{D}_Y}, 1, 1\right) \quad \beta_{y2} = f\left(\frac{e_{y2}}{\bar{D}_Y}, 1, 1\right) \quad \beta_{y3} = f\left(\frac{e_{y3}}{m.Te.\bar{V}_Y}, 1, 1\right)$
- $e_{y1} = \hat{e}_{1y} \quad \dot{e}_{y2} = \frac{\tilde{e}_{1y} + \hat{e}_{1y}}{2 \cdot m} \quad e_{y3} = \tilde{e}_{1y} - \hat{e}_{1y}$

The saturation function f is defined below. It allows to take into account the UAV limitations 3.10. $\bar{V}_X$, $\bar{V}_Y$ are the maximum allowed velocities according to x and y axes. The maximum allowed velocity of the follower is reached as soon as the components of its distance, to the leader, are greater or equal to $\bar{D}_X$, $\bar{D}_Y$. The saturation function f is also used to normalize the values of β_{xi} and β_{yj} between -1 and +1.

$$f(\mathrm{eps}, \mathrm{Vm}, \mathrm{Dm}) = \begin{cases} -Vm & \text{if} \quad \mathrm{eps} < -Dm \\ \frac{Vm}{Dm} \cdot eps & \text{if} \quad -Dm \leq eps \leq +Dm \\ +Vm & \text{if} \quad eps > +Dm \end{cases} \tag{3.11}$$

From 3.10, we get the following control law:

$$\overrightarrow{U_1}^T = R \cdot \overrightarrow{V_L}^T + \begin{bmatrix} A(\hat{e}_{1y}) \cdot f\left(\bar{D}_X \cdot \sum_{i=1}^{3} \beta_{xi}, \bar{V}_X - V_{Lx}, \bar{D}_X\right) \\ f\left(\bar{D}_y \cdot \sum_{j=1}^{3} \beta_{yj}, \bar{V}_Y - V_{Ly}, \bar{D}_y\right) \end{bmatrix} \tag{3.12}$$

In steady state 3.10, the follower will fly at the speed of the leader without tracking errors only if it receives, through a communication network, the leader velocity$\vec{V_L}$ (without disturbances, latencies or losses.

3.3.3 Leader UAV velocities estimation

To compensate the tracking errors due to disturbances, latencies or loss of communication, the leader UAV velocity vector $\vec{V_L}$ is estimated by using a first order dynamic model of the UAV in the following equation:

$$\dot{\overrightarrow{V_{1m}}}^T = \begin{bmatrix} -\frac{1}{\tau_x} & 0 \\ 0 & -\frac{1}{\tau_y} \end{bmatrix} \cdot \overrightarrow{V_{1m}}^T + \begin{bmatrix} \frac{1}{\tau_x} & 0 \\ 0 & \frac{1}{\tau_y} \end{bmatrix} \cdot \overrightarrow{U_{1m}}^T \tag{3.13}$$

$\vec{V}_{1m} = \begin{bmatrix} V_{1mx} & V_{1my} \end{bmatrix}, \vec{U}_{1m} = \begin{bmatrix} U_{1mx} & U_{1my} \end{bmatrix}$
are the velocity and control input vectors of the model respectively. In the earth frame as shown in figure 3.1, its position is:

$$\dot{\overrightarrow{P_{1m}}} = R_o \cdot \overrightarrow{V_{1m}} \quad \text{with} \quad R_o = \begin{bmatrix} \cos(\delta_1) & -\sin(\delta_1) \\ \sin(\delta_1) & \cos(\delta_1) \end{bmatrix} \tag{3.14}$$

Considering the following filtered distance errors (Fig.1):

$$\overrightarrow{\hat{e}_{1m}} = \left[\begin{bmatrix} \hat{d}_1 * \cos\left(\hat{\theta}_1\right) \\ \hat{d}_1 * \sin\left(\hat{\theta}_1\right) \end{bmatrix} + R_o^{-1} \cdot \left(\overrightarrow{P_1}^T - \overrightarrow{P_{1m}}^T\right) \right] - \begin{bmatrix} X_{SP1} \\ Y_{SP1} \end{bmatrix} \tag{3.15}$$

with: $\overrightarrow{\hat{e}_{1m}} = [\hat{e}_{1mx} \quad \hat{e}_{1my}]$
The control objective is to regulate the distance errors as follows:

$$\lim_{t \to \infty} \hat{e}_{1mx}(t) = 0 \quad \lim_{t \to \infty} \hat{e}_{1my}(t) = 0 \tag{3.16}$$

The following error dynamics equation is proposed to fulfill the control objective:

$$\dot{\overrightarrow{\hat{e}_{1m}}}^T = -\begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \cdot \overrightarrow{\hat{e}_{1m}} = R \cdot \vec{V}_L^T - \overrightarrow{V_{1m}^T} \tag{3.17}$$

with: $K_x = \frac{\overline{V}_X}{\overline{D}_X}$, $K_y = \frac{\overline{V}_Y}{\overline{D}_Y}$
In steady state (3.13, 3.15), $\vec{U_{1m}} = \vec{V_{1m}}$, and $\vec{V_{1m}} = R \cdot \vec{V_L}$.

The tracking errors 3.16, will be null only if the dynamic model get the leader UAV velocities without disturbances, latencies or losses. Otherwise, the tracking errors will depend on the values of the proportional gains K_x, K_y. Indeed, let us consider the following candidate Lyapunov function:

$$V = \frac{1}{2}\overrightarrow{\hat{e}_{1m}} \cdot \overrightarrow{\hat{e}_{1m}} = \frac{1}{2} \cdot \left(\hat{e}_{1mx}^2 + \hat{e}_{1my}^2\right) \tag{3.18}$$

Its time derivative can be written as: $\dot{V} = e_{1mx} \cdot \dot{e}_{1mx} + e_{1my} \cdot \dot{e}_{1my}$ Thus, $\dot{V} < 0$ if $|\hat{e}_{1mx}| \geq \left|\frac{V_{Lx}}{\kappa x}\right|$ and $|\hat{e}_{1my}| \geq \left|\frac{V_{Ly}}{Ky}\right|$ so that the size of the tracking errors $|\hat{e}_{1mx}|$ and $|\hat{e}_{1my}|$ are uniformly and ultimately bounded by $\frac{\vec{V}_X}{Kx}$ and $\frac{\bar{V}_Y}{Ky}$ respectively, where $\vec{V}_X$ and $\vec{V}_Y$ represent bounds on the leader UAV velocity components satisfying $|V_{Lx}| < \bar{V}_X$ and $|V_{Ly}| < \bar{V}_Y$. These tracking errors do not have any effects since they concern the dynamic model. This interesting result allows to get the leader UAV velocities even in case of degraded (latencies, loss, ...) communications. From 3.16, we get for the dynamic model the following control law:

$$\overrightarrow{U_{1m}} = R.\overrightarrow{V_L}^T + \begin{bmatrix} Kx & 0 \\ 0 & Ky \end{bmatrix} \cdot \overrightarrow{\hat{e}_{1m}} \tag{3.19}$$

Finally, by combining equations 3.19, and 3.12, we get:

$$\overrightarrow{U_1}^T = \overrightarrow{U_{1m}}^T + \begin{bmatrix} A\left(\hat{e}_{1y}\right) \cdot f\left(\overline{D}_X \cdot \sum_{i=1}^{3} \beta_{xi}, \overline{V}_X - V_{Lx}, \overline{D}_X\right) \\ f\left(\overline{D}_y \cdot \sum_{j=1}^{3} \beta_{yj}, \overline{V}_Y - V_{Ly}, \overline{D}_y\right) \end{bmatrix} \tag{3.20}$$

3.4 Simulation Results

3.4.1 Matlab-Simulink

The simulations have been carried out on Matlab-Simulink software. They are based upon a dynamic model of the UAV [66] combined with usual low level controllers (PID control for horizontal velocities and Phase Lead control for altitude and angular (yaw) positions). The simulation parameters are the following ones:

Simulation time: T=800s - Sampling period: Te=0.01s Altitude (after the take-off phase): 10m Omnidirectional camera position: Loc = 0.15m Attenuation parameter A_y=0.3 Maximum velocities: Maximum distances: DES algorithm: α_d ,$\theta = 0.5$, λ_d ,$\theta = 0.5$, m = 10 Time constants (first order model): $\tau_x = 0.2s$, $\tau_y = 0.2s$ Take-off conditions (m, deg) / linear-circle trajectories :

- Leader (X_L, Y_L, θ_L) : [-82,-82,0°] - [-95,0,90°]
- Follower 1 (X_1, Y_1, θ_1) : [-90,-82,0°] / [-95,-25,90°]
- Follower 2 (X_2, Y_2, θ_2) : [-90,-74,0°] / [-110,-15,90°]
- Follower 3 (X_3, Y_3, θ_3) : [-90,-90,0°] / [-80,-15,90°]

In order to study the LF formation and the global aspect of the Virtual Rigid Body (VRB), linear and circle trajectories (radius of the circle: 90m - leader velocity according to x and y axes: 1m/s) have been chosen. The VRB requirements are the following ones:

circular trajectory (diamond):

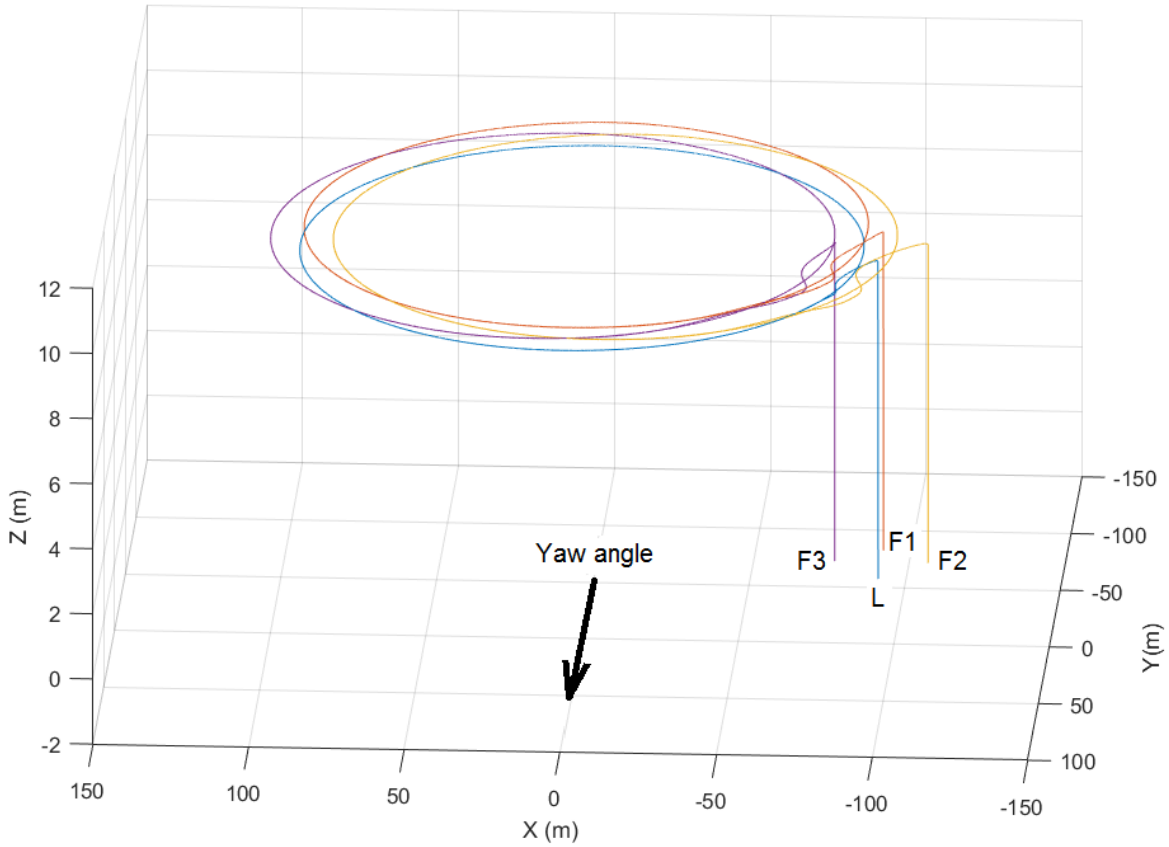


Figure 3.2: Circular trajectory

- $X_{SP1} = +20m$ | $Y_{SP1} = 0m$ | $Z_{SP1} = 0m$
- X_{SP2} = +10m | Y_{SP2} = -10m | Z_{SP2} = 0m
- X_{SP3} = +10m | Y_{SP3} = +10m | Z_{SP3} = 0m

UAVs yaw angle: $\theta_{1,2,3} = 90°$

Linear trajectory (line):

- X_{SP1} = +10m | Y_{SP1} = 0m | Z_{SP1} = 0m
- X_{SP2} = +10m | Y_{SP2} = -10m | Z_{SP2} = 0m
- X_{SP3} = +10m | Y_{SP3} = +10m | Z_{SP3} = 0m

UAVs yaw angle: $\theta_{1,2,3} = 0°$

The simulation results in Figures 3.2,3.3 show that the UAVs follow properly (for different velocities and orientations) the circular and linear trajectories after the take-off phase and fly according to the desired VRB requirements (a triangle for the linear trajectory and a diamond for the circular trajectory). Then, to study separately and in details the effects of the leader velocity estimation with a first order dynamic model, the following scenario has been implemented for both trajectories: Before t=200s, the leader UAV velocity is (in m/s). Between t=200s and t=600s, the leader UAV acceler-

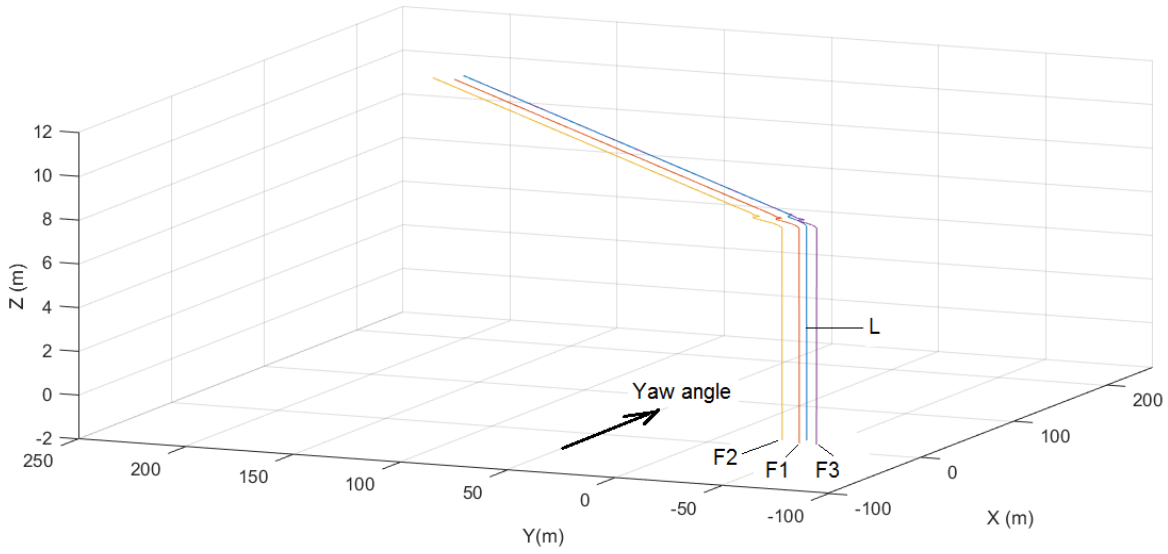


Figure 3.3: Linear trajectory

ates (in m/s). After t=600s, the leader UAV recovers the previous velocity (in m/s). From t=400s to t=800s, the communication between the UAVs is not available. The simulation results correspond to the follower UAV 1. They are similar for the others.

On Figures 3.4, 3.5, 3.6, 3.7 the tracking errors are shown according to x and y axes converge toward zero when the leader UAV flies at different speeds and even when the communication between the leader and the followers is not available. As long as the communication is available, the model errors are null and the follower UAV 1 uses the speed information sent by the UAV leader. When the communication is lost, the model errors are not null and allows to recover the speed information.

3.4.2 Gazebo

In order to simulate the results with a more dynamic realistic environment, a ROS-Gazebo simulation is being utilized. An open source package known as ROTORS [67] is chosen to be the main building block as it matches to great extent both the software standards, e.g., ROS and the hardware used for experiments in the lab. It relies on px4 open source flight stack controller mimicking pixhawk as software in the loop[68].

Each robot has the capability to navigate autonomously relying only on its on-board sensing capabilities; ie: vision and inertial measurement unit. Each follower robot is utilizing the onboard embedded monocular perspective camera to track and detect the leader robot. The leader robot has a fiducial marker primarily to do identification. The choice of the marker was fallen on Apriltag [1], a more efficient and faster than ARtag. These fiducial markers or cues are very relevant in robotics

[1]AprilTag 2: Efficient and robust fiducial detection, Wang et al, IROS 2016, `https://github.com/AprilRobotics/apriltag`

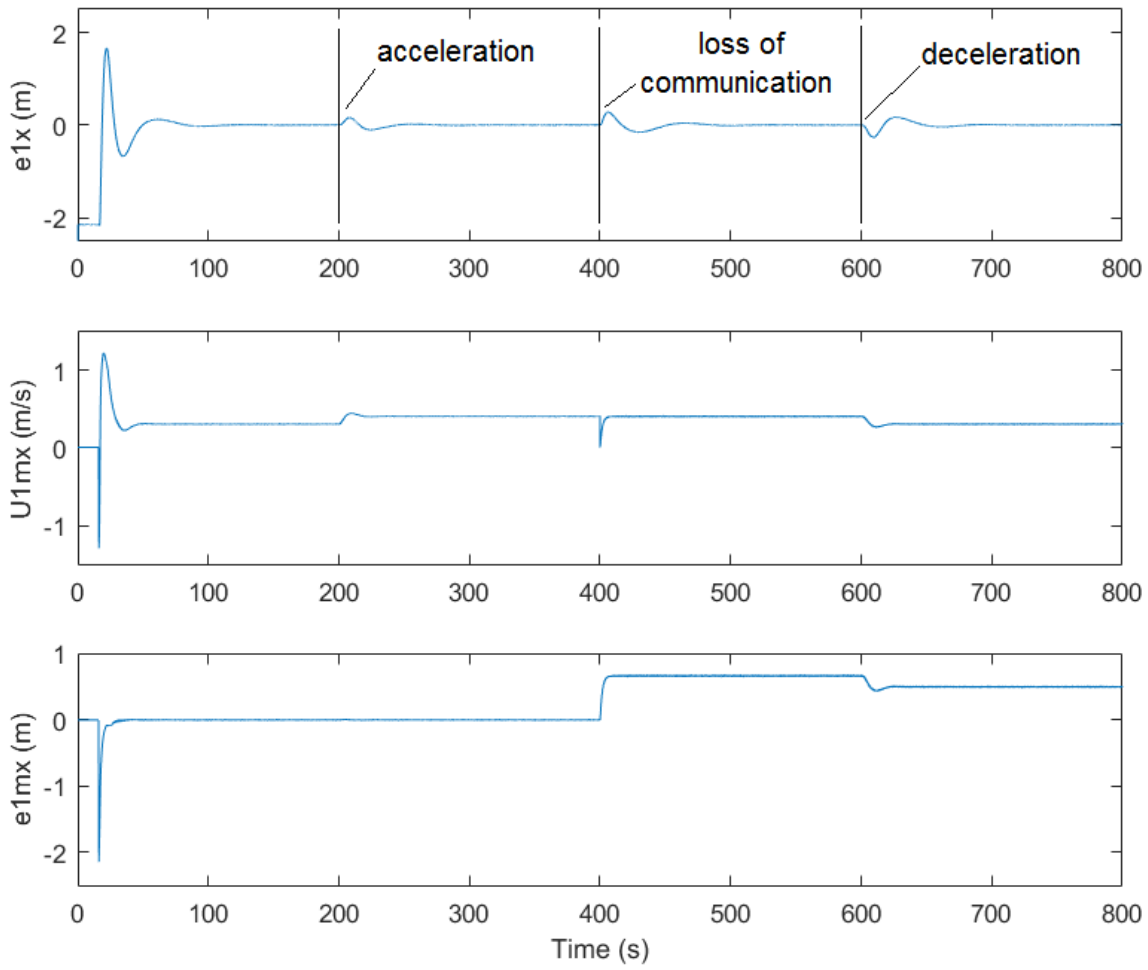


Figure 3.4: Follower 1 (linear trajectory) x uav tracking error (e_{1x}) | model control input (U_{1mx}) | x model error (e_{1mx})

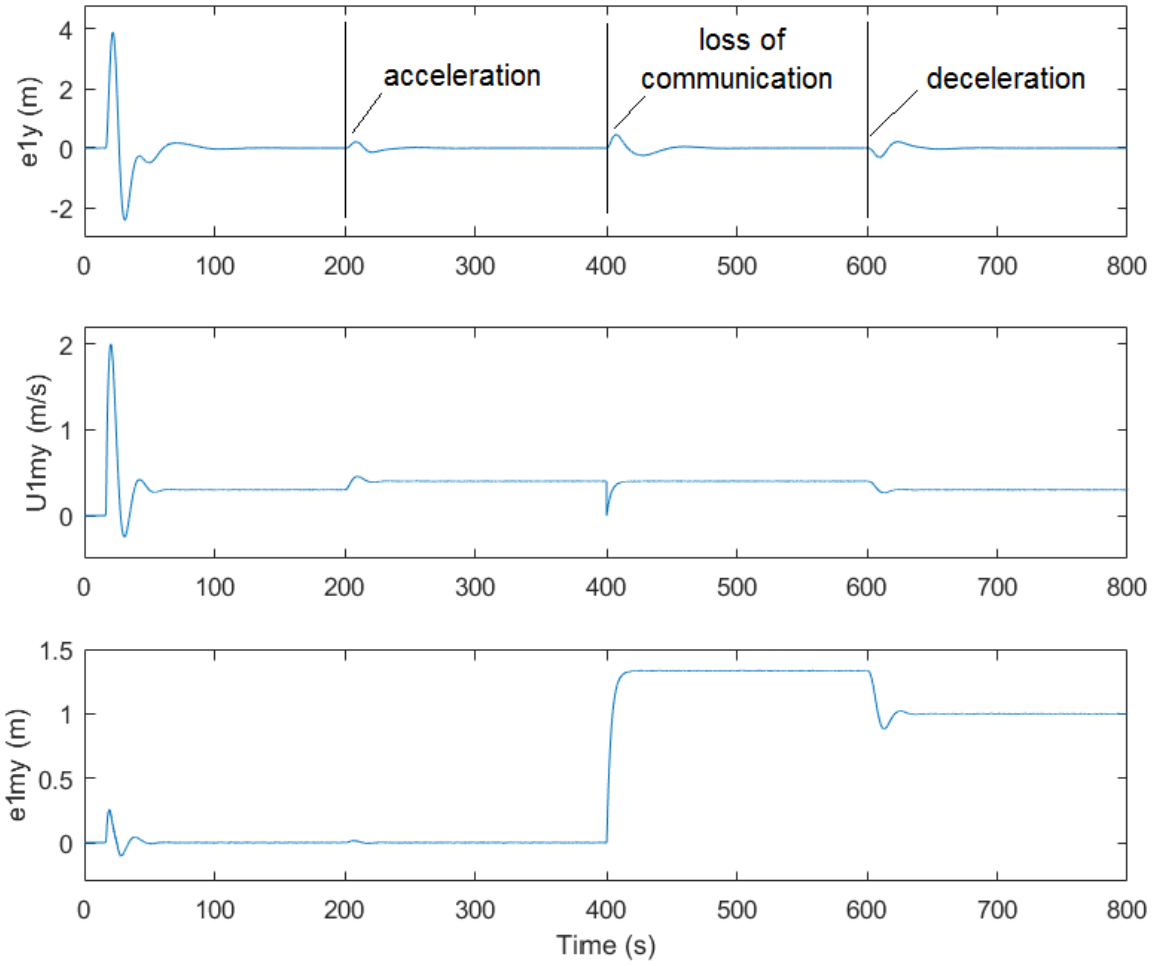


Figure 3.5: Follower 1 (linear trajectory) y uav tracking error (e_{1y}) | model control input (U_{1my}) | y model error (e_{1my})

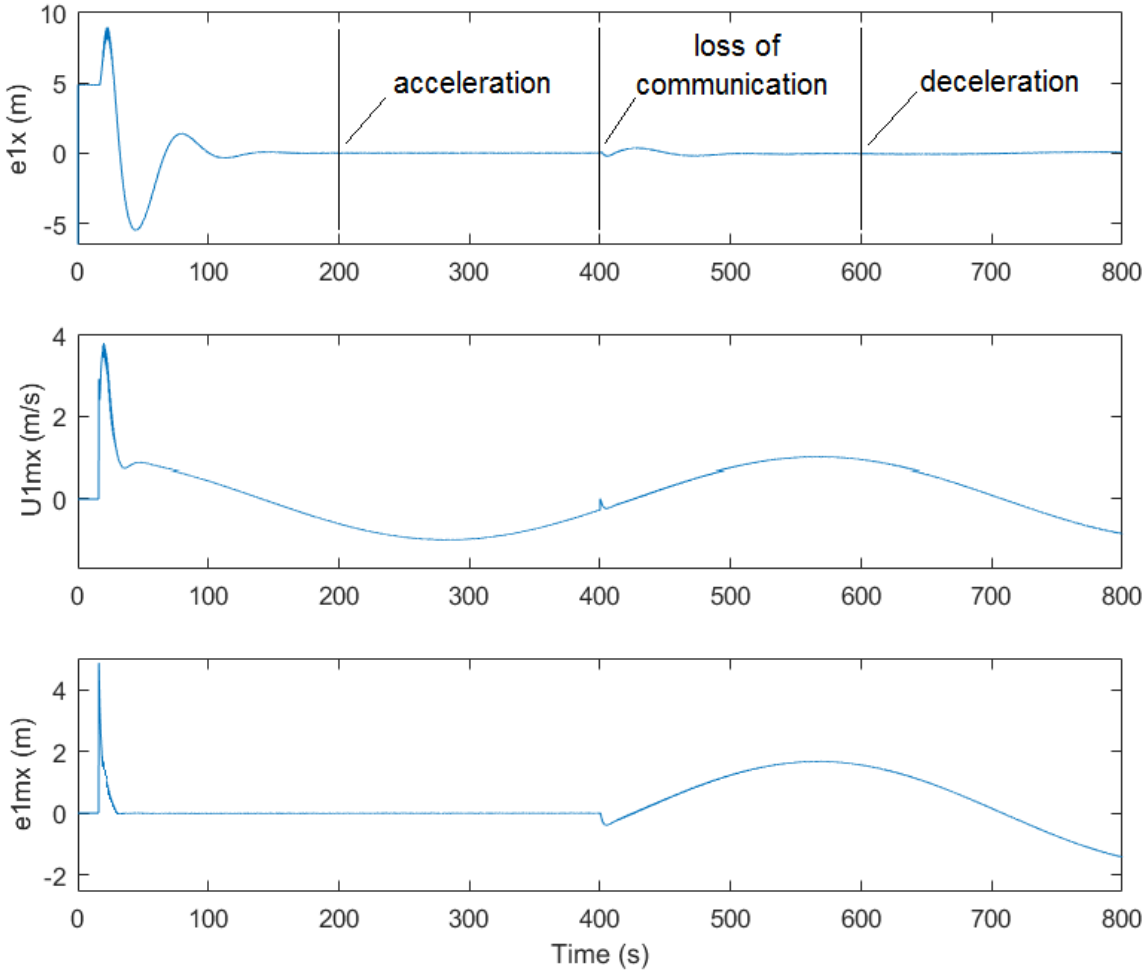


Figure 3.6: Follower 1 (circular trajectory) x uav tracking error (e_{1x}) | model control input (U_{1mx}) | x model error (e_{1mx})

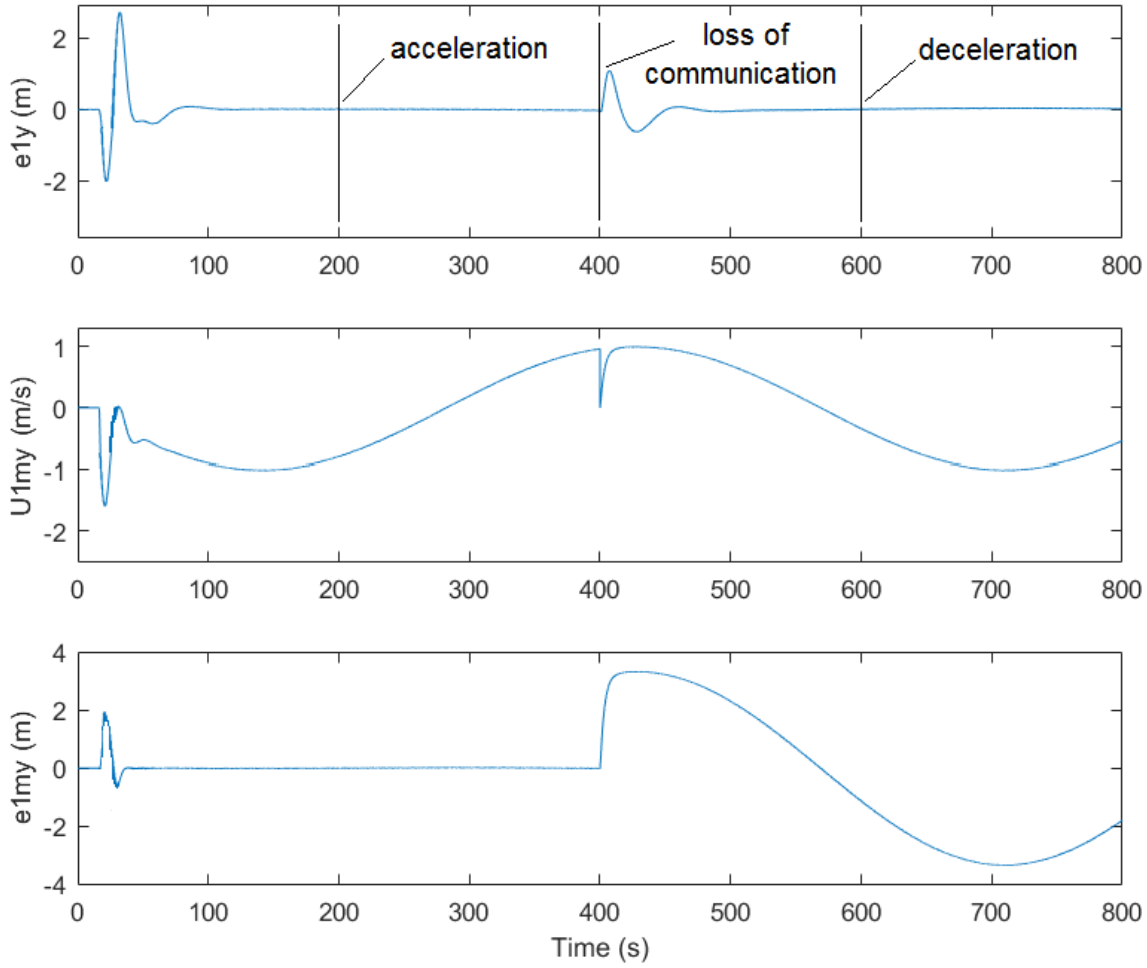


Figure 3.7: Follower 1 (circular trajectory) y uav tracking error (e_{1y}) | model control input (U_{1my}) | y model error (e_{1my})

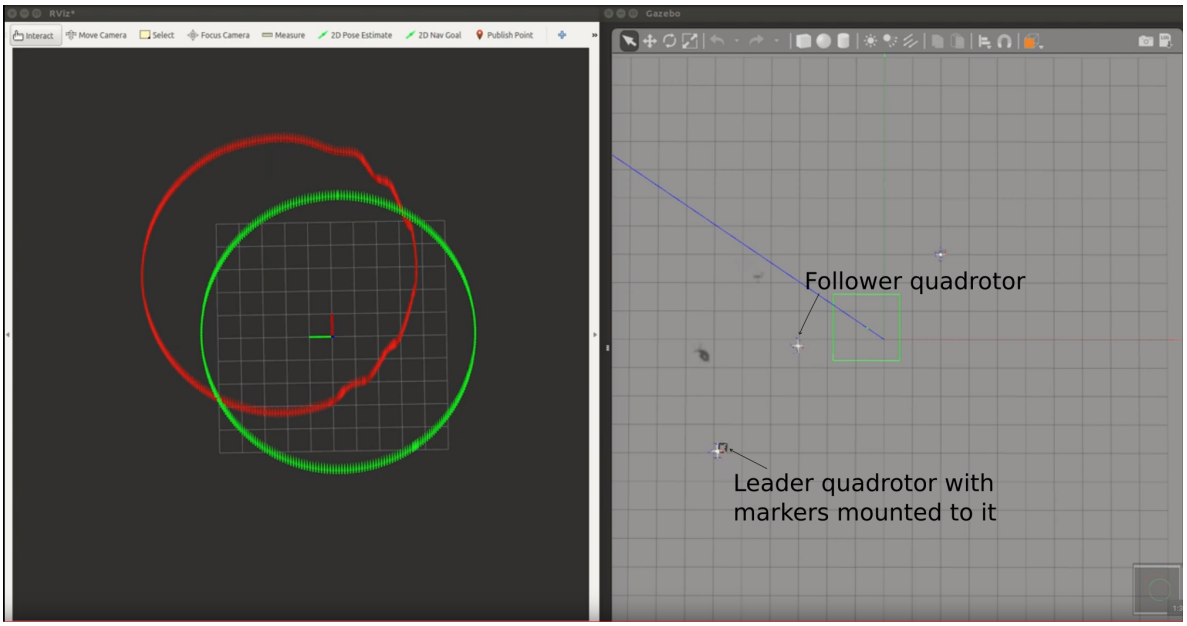


Figure 3.8: Gazebo Simulation for LF.

domain. They are even on planet Mars, aiding the robot arm automatic identification and automatic calibration [69]. Their ease of mounting and use made them ubiquitous in many systems. they can be engraved or attached to the robotic system during the assembly phase. As well they provide convenient direct process of extracting distance and bearing measurements.

In figure 3.8, the left hand side represents the visualization of the paths, leader UAV in green, follower in red. The right hand side represents the Gazebo environment simulating the robots dynamics and interactions. Fiducial marker is attached to the leader as a cube to be detected from all the sides while the robot is rotating.

The desired spacing is - 2.5 in x and y. According to the graph this value have been tracked with some fluctuation around the desired -2.5 since there is no communication between the leader and follower drone. As shown in the plot in 3.9

3.5 Experimental Results

3.5.1 Description

As described in the simulation, the goal of the experiment is to validate the follower's robot algorithm able to track and maintain a desired distance and bearing to the leader robot.

3.5.2 Setup

The quadrotor is considered as the follower, the omni-directional ground mobile robot as the leader. The leader robot can either be commanded by a joystick or with a trajectory generation algorithm to

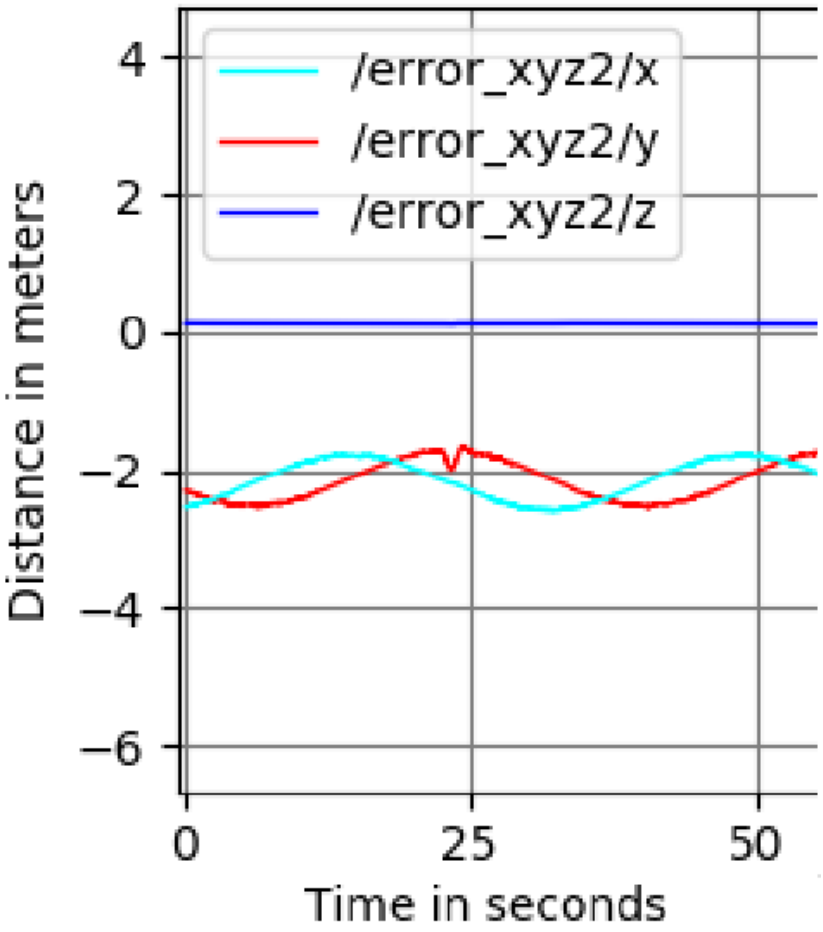


Figure 3.9: Desired distance in between the leader and follower.

Figure 3.10: Real Experimental setup.

perform specific navigation. The quadrotor was placed on the ground randomly apart from the the mobile robot. The taking off position should respect the detection distance of the marker from the quadrotor's onboard camera. In such a scenario, it was effective within 4-5 meters away from the ground mobile robot.

3.5.3 Procedure

Then the experiment starts with the drone takeoff stage, then sanity check of the sensors. Afterwards the formation control algorithm is activated. The required formation could be shown as in figure 3.10

In figure 3.10, the quadrotor drone (Parrot Bebop) is executing the following controller commands to keep track of the omni-mobile robot that is controlled by a user.

The scenario of that experiment was taken in one shot but with varying leader's velocities mainly in the global Y direction which induces the error in x axis of the image.

3.5.4 Results

In the figure 3.11 the graph shows the relation between the distance error measured by the quadrotor with respect to time. During the first 20 seconds in this experiment, the quadrotor robot is trying to track and maintain the formation following the take off phase, while the ground mobile robot is stationary. The leader is commanded a velocity of 0.1 m/s at around 55 seconds from the time the quadrotor took off and the follower's error calculation from the visual feedback starts to bump a little bit above zero. The controller was able to converge back toward zero which implies the quadrotor is able to follow. The leader only stopped at the time 80 seconds.

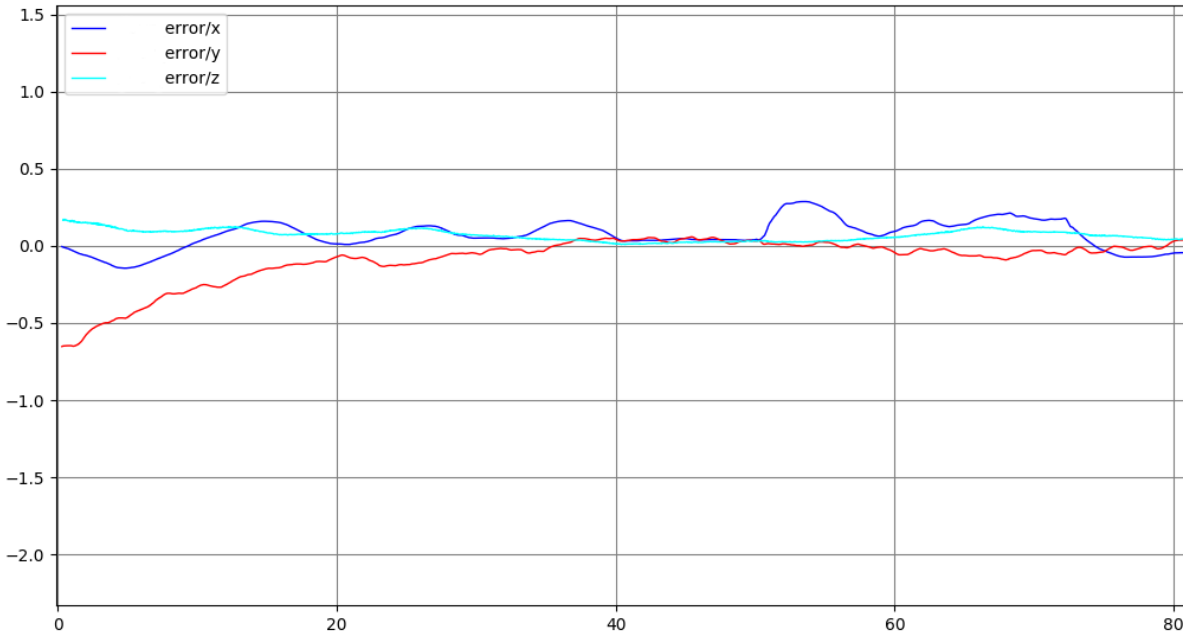


Figure 3.11: Error according to x,y,z in meters vs time in seconds.

At around 80 seconds the leader is commanded negative velocity of -0.15 m/s the error is bound and the follower is trying to maintain the cohesive motion.

To test the framework in a harsher scenario, a rapid switching velocity was commanded to the leader from -0.15 m/s to 0.20 m/s inducing a big error at around 115 seconds as shown in figure 3.12. The follower was able to tackle such scenario with some error but was able to converge to near zero eventually.

The main concern in this experiment as stated was related to the error of x. But due to the irregularity of the ground that the mobile robot is navigating there are some uninduced error according to Z and Y, but that does not affect the cohesive motion tracking. As shown in the figures these desired distances are maintained as well.

As expected, as the velocity of the leader increases, the error peak increases as well. But the same controller with the same gains without extra tuning was able to track the leader and keep the desired formation. Some irregularity as well and the error is not absolute zero as in reality the drone is not in

a perfect hover.

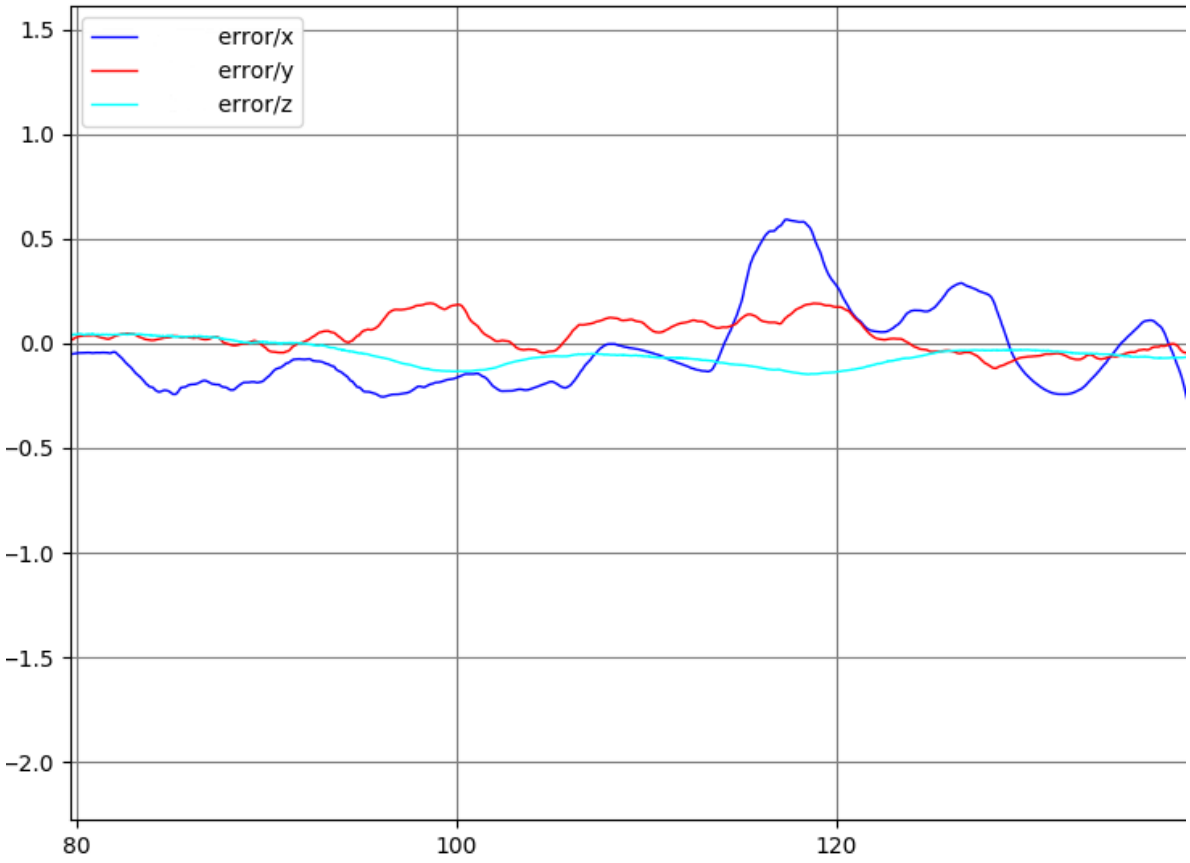


Figure 3.12: Error according to x,y,z in meters vs time in seconds.

3.6 Conclusion and Perspectives

In conclusion, the formation control can work in a fully decentralized way. Indeed, unlike other VRB methods, which rely on the group barycenter to steer the formation [70], this proposed approach requires neither positioning nor tracking from a centralized point.

Simulation results for several scenarios/trajectories were provided to assess the algorithm. In addition, these results were corroborated experimentally using formation in between an omni-directional mobile robot with a mounted tower and attached marker, and quadrotor UAV.

The major drawback in this method, as in others in the literature is relying on distance measurements for the algorithm to generate appropriate control signals. Moreover the fiducial marker used can be replaced, when the identification problem in robotics is solved. This challenge is an open problem with many proposals in literature.

CHAPTER 4

Image Based Visual Servoing for Multi-Aerial Robots Formation

Contents

4.1 Introduction

The numerous advantages of flying in group over using single robot in mission execution was discussed in previous chapters and extensively in literature, but usually algorithms are demanding the communication of different measurements in between the robots. The crucial issue of coordinating the motion between the robots can rely on sharing information such as distance, position, orientation, velocity, and attitude.

In the previous chapter, it is shown that visual tracking and prediction can be used to achieve a formation control even in the presence of intermittent communication. The controller was using the distance and bearing angle between a leader and follower robot. In this chapter, the goal is to propose method that can deliver similar performance but without having direct distance measurement or relying on specific marker in a more natural way like humans advanced perception systems to track targets. An Image Based Visual Servoing (IBVS) technique is utilized for building and maintaining a Leader-Follower (LF) configuration of multi aerial vehicles (UAVs) without communication. An Extended Kalman Filter (EKF) is used as well for the prediction of the leader's motion as well as follower robot state estimation and noisy measurement filtering.

The coordination between group of robots can be handled by solving the tracking and mutual localization between robots in the same group through formation control. In this chapter a method is proposed in the direction of solving such challenge of tracking and mutual localization in a unified framework.

Eventually, according to the targeted application, it may be necessary or desirable that drones fly following a given geometric shape (line, diamond, etc.), a property as well of formation control. Building and maintaining a spatial geometric shape while evolving within the environment usually requires extensive communications between the robots for coordinating their movements. In this work we focus on the use of an Image Based Visual Servoing (IBVS) technique for building and maintaining a Leader-Follower (LF) configuration of multi aerial vehicles (UAVs) without communication. While most IBVS techniques either require rigorous camera calibration or regulate the error according to the three robot axes, our approach avoids the calibration phase by relying on image moments features to provide a vision-based predictive compensation method. The follower robot's solution works in GNSS-denied conditions and can run using only on-board sensors. The method is validated through simulations for a group of three quadrotors, and another with two quadrotors. Then these results were coroborated experimentally with two holonomic robots as presented in section 4.6. The experiments where carried out by a quadrotor aerial mobile robot as a follower robot representing the main controller of the follower robot. An omni-directional ground mobile robot with a mounted marker (for robot identification reason) is used as a leader robot.

4.2 Visual Servoing

Visual servoing is an old technique that merge both the visual information in a control loop binding the effort of having a camera attached on a robot. Early work of visual servoing started to emerge in 1980's. It became more mature later with the results of the work of adaptive visual control of a robotic arm [71], and then with the work of Peter Corke in [72]. The control design can be described that at

first a set of points have to be selected, then an analytical form of the interaction matrix is proposed. Then a goal or a desired image represents the final camera pose along with an initial pose have to be defined too. Visual Servoing later became more mature then with the results of the work of adaptive visual control of a robot arm [71], and the work of [72, 73].

There are major techniques that classify the field of Visual Servoing (VS) like Image Based Visual Servoing (IBVS) and Position Based Visual Servoing (PBVS). There are other branches that emerge from the pros and cons of both primary branches like. IBVS is considered to use information directly in the image plan to do control while on the contrary PBVS is using the pose of a target to control the robots. Configuration of the camera with respect to the robot, is having also an important classification on the techniques, formally there are two configurations: eye-in-hand and hand–eye/eye-to-hand. The first configuration is having the camera attached to the robot while the later means the camera is observing the robot in a scene. It depends on the application and the requirement on which technique could be used. In this thesis, the eye-in-hand configuration is going to be used due to the assumption of the availability of cameras on aerial robots. The work in literature during 1990's was more concerned about the interaction matrix evolution, and feature selection for the model of the target; such as (point,line, edge, centroids, etc).

4.3 Related Work of IBVS for LF Framework

In this section a general view about the related work in literature that have investigated some results related to VS in the context of MRS.

MRS have been widely studied since the 90s [2, 3]. Many advantages can be expected from using a group of robots: increase and share of the payload [4], reduction of the time needed for the achievement of a task [5], fault tolerance and resilience of the system [6, 7], or use of simpler and cheaper robots for adaptability to the environment [8]. However, making a group of robots evolving together in the same environment entails the resolution of some problems. Coordination is one of the most critical ones, especially considering decentralized settings. Coordination between the members of the group can be achieved through various ways, from behavior-based or forces-based methods to Virtual Rigid Body (VRB) structure building, through Leader-Follower (LF) schemes.

In the present work we focus on the last approach. We consider a set of UAVs equipped with ubiquitously available sensors for flight controller, a perspective camera, but, no possibility to communicate with each others. In addition to the absence of communications, leader's velocity is considered unknown and no beforehand camera calibration is done.

VS require camera calibration in order to estimate the depth of the features in the image used. In the context of multi robot system, that would be a tedious time consuming task, some algorithms utilized in the method proposed in this chapter are tested against the non-accurate calibrated cameras bypassing pitfall that can occur. On another side note, camera self calibration is not yet widely availably used.

PBVS works as minimizing the error in 3D space. Its major disadvantage is when the feature is lost from the image during the control loop. On the contrary, IBVS is working directly in the image plane and the controller error is defined as the difference between the observed and desired feature coordinates. The disadvantage however in IBVS is stability and convergence problems that

may happen due to the singularity of the image jacobian where the controller will fall in local minimum at points with unrealizable image motion [74].

In the context of building and maintaining LF formation control without communication, most of the time, compensating for such information absence, solutions rely on a sensor-based mechanism. Pan-controlled camera [61] or omnidirectional camera like in [62, 63, 64], or other types of sensors like structered light (i.e: kinect) [64] are often considered. Note first that the camera provides each follower the bearing angle with respect to the leader. In 2010 Fabio Mobidi and his colleagues show that an estimation of the distance between the follower and the leader, a.k.a. range estimation, can be obtained from an omnidirectional camera. However, the proposed method requires a communication between the leader and the followers. More recent works avoiding communications and considering cameras were proposed. From the data obtained by the sensors, each follower of the formation attempts to estimate the pose of the global or of its local leader. Unlike in our work, the design of the controller is often based on this estimation for deriving both the linear and angular speed for maintaining the configuration.

In 2015, Chen and Jia have presented an adaptive controller for a LF configuration without relying on communications [61]. An active vision sensor was used to track the leader, and they proposed both a controller for the active vision actuation and one for the tracking. However, an accurate calibration of the camera was required for the distance measurement for leader's velocity estimation. This calibration is also a critical point for other methods relying on mutual pose estimation or relative pose estimation [75]. This makes a clear difference with our approach, which does not require camera calibration, that needs only four non co-linear points in the image plane to determine, in the follower's camera frame, the desired state of the leader.

More recently, in 2017, Guo and his colleagues proposed a method for LF formation control considering a robotic system in which communications are not allowed, leader's velocity is considered unknown and each follower is equipped with an uncalibrated omnidirectional or perspective camera [10], thus settings very similar to our current work. However, the described adaptive estimator, based on several feature points, failed when the movement of the leader is co-linear to the follower's camera axis, a situation for which the method proposed in this chapter offers a solution as explained in Section 4.4.

The last notable work under this set of constraints is due to Liu and his colleagues who consider robots equipped with kinects [64]. This sensor allows an estimation of the pose (relative orientation and position of the leader) and provides directly both angles and distances measurements but at the price of a very limited field of view, incompatible with many LF configurations.

In that context, our main contribution is a vision-based predictive compensation method for building and maintaining a Leader-Follower configuration under a set of restrictive constraints: no communications between the UAVs, no prior accurate calibration of the camera and considering the leader's velocity as unknown. The method is based on the use of image moments as features for Image Based Visual Servoing (IBVS), and unlike most of existing works our approach does not use explicit distance measurement or camera calibration, thus can also be applied in a GNSS-denied environment.

The remainder of the chapter is organized as follows. Section 4.4 presents the problem and the resolution method proposed. Validation of the approach through simulations, considering three aerial robots with a six Degrees of Freedom (DoF), is described in Section 4.5. Concluding remarks followed

by a discussion about going from simulation to real-world experiments are presented in the last section.

4.4 Problem statement

4.4.1 Problem description

The main task to be achieved by navigating a group of robots can be described as the ability to construct a geometrical shape sometimes known as Virtual Rigid Body (VRB). The virtual structure between several robots aims at facilitating the agile control of several multi-rotor formations. The goal of the follower's robot is to follow the target, the leader in such case, by keeping a predefined separation distance d and bearing angle θ, as illustrated by Figure 4.1. Additionally we control the follower UAV without directly measuring distance and utilising the image plane of the camera that is mounted on the robot. The UAV leader is assumed to track perfectly a virtual target which moves along a defined trajectory (straight line, circle, etc) or with a human in the loop motion planning (description of leader's controller is out of the scope of this chapter). Unlike other VRB methods, which rely on the group's barycenter to steer the formation [70], in our case each robot is planning its own trajectory contributing to a distributed formation control, such that no positioning and tracking from a centralized point is required.

We adopt a multi-layer control scheme composed of a single robot layer and a formation controller layer. The single robot control, which is responsible to provide the control signals to the robot, can be achieved with off the shelf flight controller.

Figure 4.1 shows a Bird-eye view of the formation to create a triangle or V-shape with the leader in front carrying a fiducial marker. The follower is having a perspective camera viewing the leader and its marker. The camera image view is shown in figure 4.3.

In the context of unavailable communication in between the robots, the formation control should estimate the value of the leader's velocity by one way or another. In [76], we designed a high level controller which only requires simple tunings and rests on a predictive filtering algorithm and a first order dynamic model to recover an estimation of the leader's velocities and avoid the tracking errors. The proposition of this current chapter is to develop and test a system that does not directly measure the distance to leader and at the same time preserves the virtual structure as concise as possible. For this constraint an Image Based Visual Servoing (IBVS) will be used.

4.4.2 Approach description

Image moments feature

First we have to define the features that are selected carefully for IBVS on an underactuated system. The features used for the algorithm are based on the popular image moments proposed in [77]. They are represented starting from defining the object in the field of view as having n points, each point $p_k = (x_k, y_k)$ represents the coordinates in the image plane. The moments m_{ij} and the centered moments μ_{ij} of this object are defined as:

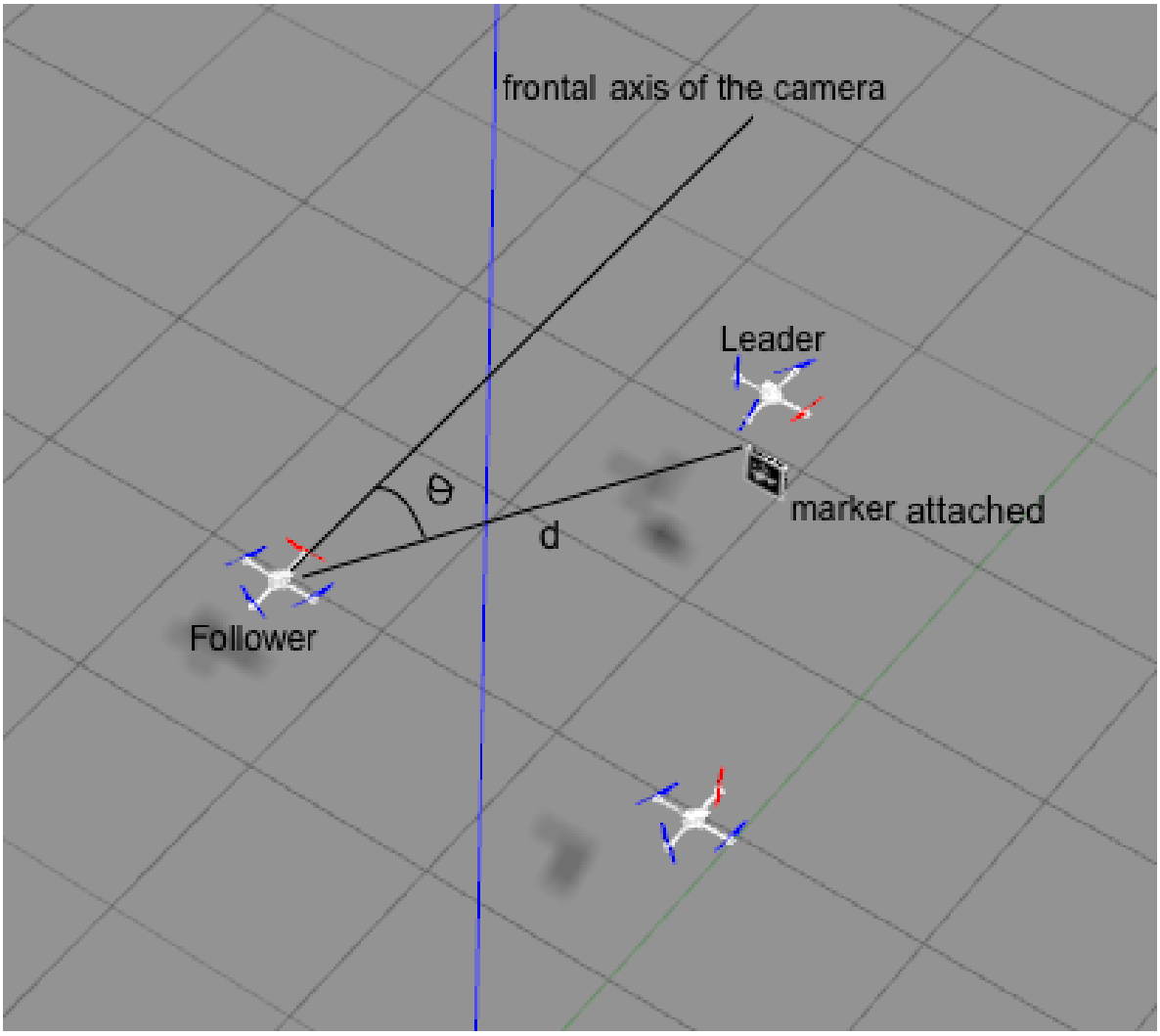


Figure 4.1: Gazebo simulation for LF.

$$m_{ij} = \sum_{k=1}^{n} x_k^i y_k^j \tag{4.1}$$

$$\mu_{ij} = \sum_{k=1}^{n} (x_k - x_g)^i (y_k - y_g)^j \tag{4.2}$$

Where i, j represents the index of the moments, $x_g = m_{10}/n,\ y_g = m_{01}/n$ and $n = m_{00}$. Since the considered shape is composed of a discrete set of points, then it's area can be expressed as:

$$a = \mu_{12} + \mu_{21},\ \ a^* = \mu_{12}^* + \mu_{21}^* \tag{4.3}$$

Where a and a^* are the current and the desired areas of the shape. To choose visual features to control the translational Degrees Of Freedom (DOF), we can make use of the features proposed in [77] and validated with more results in [78].

As mentioned in the introduction, if the target moves in the direction of the frontal axis of the center of the camera (z-axis), this can imply several singularities in the tracking. This was noted as a pitfall in the work of [63].

For avoiding such situation, a normalized version of the features are considered as mentioned in [77] which suggests that the direct use of the shape's area a to control the robot's motion in the z-axis can give bad behavior because the interaction matrix relating a and v_z is not constant therefore, the dynamics between them will be different, not like the case of x_g and y_g. Where in fact they can provide a good control for their respective velocities v_x and v_y. So to avoid that, a better choice would be by taking a normalized version of these features and are defined such that:

$$a_n = Z^* \sqrt{\frac{a^*}{a}},\ x_n = a_n x_g,\ y_n = a_n y_g \tag{4.4}$$

Where a^* and Z^* are the area and the depth of the target at the desired pose. And x_n, y_n are the normalized centroid of the shape.

Errors that we are going to regulate later through the controller. A current and a desired states of the image have to be defined.

$$e = s - s^* \tag{4.5}$$

where $s = (x_n, y_n, a_n)^T$ and $s^* = (x_n^*, y_n^*, a_n^*)^T$ be the current and the desired features' states respectively, where x_n, y_n and a_n are expressed in equation (4.4).

The relation between the image dynamics and the velocity of the camera can be defined as:

$$\dot{e} = L_s V_c \tag{4.6}$$

Since the velocities according to x,y, and z axes are the ones to be controlled, the interaction/Jacobian matrix becomes as follows:

$$L_s = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.7}$$

Normally it is designed to control 6 DOF robot arm, but since the multi-rotor UAVs are considered underactuated systems, there are 2 DOF that can not be directly controlled. For a multi-rotor UAV, hover condition means the pitch or roll angle of the UAV with respect to the world plane should be zeros. A classical quadrotor model can be commanded with twist velocity vector of four. These velocities are the velocity according to x,y,z axes and rotational velocity around z. This will lead to omitting the forth and fifth columns and rows of the stack of the original interaction matrix leading to 4.7. Taking into account the fact that a kinematic model was considered. Three inputs delivered to the low level velocity controller such that $V_c = (\nu_x, \nu_y, \nu_z)^T$, and the resulted interaction matrix is given in 4.7.

To make sure that the error is decreasing exponentially so that

$$\dot{e} = -\lambda e \tag{4.8}$$

where λ is a positive control gain, substituting in (4.6) and solving for V_c we get the velocity of the camera from the dynamics of the image:

$$V_c = -\lambda L_s^{+} e \tag{4.9}$$

The transformation from the robot frame to the camera frame is fixed and known by design.

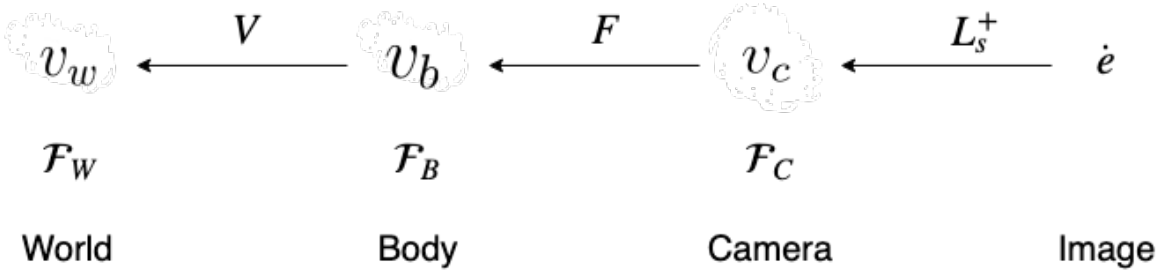


Figure 4.2: Velocity frames transformation from the image to the world

Since this camera is mounted on a UAV (in our case quadrotor), this velocity twist vector should be translated and rotated with respect to the mounting details to the center of mass of the robot. The used kinematic model of a quadrotor was taken into consideration unlike most of the other IBVS implementations for quadrotor that can be found in literature. That gives more freedom for the controller to be interchangeable to other types of quadrotors or even different multi-rotor vehicles without the need to redesign the gains, but rather fine tune them. The equation 4.6 is valid for static target. An additional aspect that should be included to this error equation to compensate for the motion of the leader. The equation of the time derivative of the error $\dot{e}$ will be:

$$\dot{e} = \dot{s} - \frac{\partial e}{\partial t} \tag{4.10}$$

we consider $\partial e/\partial t$ to be the change in the leader's position in image plane of the follower robot. The full dynamics equation is given by substituting $\dot{s}$ by it's motion equation leads to:

$$\dot{e} = L_s V_c - \frac{\partial e}{\partial t} \tag{4.11}$$

Then finally the velocity vector can be represented as:

$$V_c = \widehat{L_s^+}(-\lambda e - \widehat{\frac{\partial e}{\partial t}}) \tag{4.12}$$

where $\widehat{L_s^+}$ represents the pseudo-inverse of the interaction matrix.

Let us mention that the gains in the previous controller are adaptive, meaning that the gain value changes depending on the the error value:

$$\lambda = (\lambda_{\max} - \lambda_{\min}) \left(\frac{|e_t|}{|e_{\max}|} \right) + \lambda_{\min} \tag{4.13}$$

Where $\lambda_{\max}$ is the upper limit of the gain, $\lambda_{\min}$ is the lower limit of the gain, $|e_t|$ is the norm of the error vector at time t and $|e_{max}|$ is the norm of the maximum error value at the first iteration of the control loop. According to [79], the authors show the positive effect and smoothness of adaptive gains on different visual servoing scenarios. It was adopted by default in our developed framework.

The Velocity vector V_c is then passed to the low level controller of the flight controller(e.g., PX4 on Pixhawk) mounted on the UAV controlling its specific dynamics. Moreover to guarantee that the robot's velocity do not exceed the physical limit (in some cases security limit), a saturation function was introduced. This function will smooth and bound the velocities.

The $\widehat{\partial e/\partial t}$ is an estimation of the value of $\partial e/\partial t$. This term can be calculated in different methods dependant on the application. For instance if the target moves at a constant velocity, this value can be calculated as:

$$\widehat{\frac{\partial e}{\partial t}} = \gamma \sum_i e_i \tag{4.14}$$

Where γ is a positive gain. It should be adaptive as well as predictive to change online depending on the leader's velocity. A simplified version of that gain can be constant by tuning in the case where leader's velocities are constant, which is not our case.

Since there is no communication in between the robots, the follower robots should predict leader's velocity from the direct observation of the image moments. Building on the assumption that the robots are equipped with Attitude and Heading Reference System (AHRS) sensors, the robot can have good estimate of its own velocity in 3D space. This is an actual assumption that should hold for the robot to be able to stabilize itself using its low level flight controller. To get the prediction of the leader's velocity from the direct observation of the image moments, a common nonlinear Extended Kalman Filter (EKF) is implemented.

The main components of the EKF can be defined as follows, Fundamental matrix $\mathbf{F}$ can be defined as a nonlinear function $f(\mathbf{x}, \mathbf{u})$, and the linear expression $\mathbf{Hx}$ is replaced by a nonlinear function $h(\mathbf{x})$:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) + n_x$$
$$\mathbf{z} = h(\mathbf{x}) + n_z$$

where n_x and n_z are the noises.

$\mathbf{x}$ is the state vector. It can be defined as the velocities according to x,y,z axes and their corresponding angular velocities w_x, w_y, w_z.

The main equation of the filter can be written as: $\bar{\mathbf{P}} = \mathbf{FPF}^\mathsf{T} + \mathbf{Q}$

F is considered the state transition matrix, with values choosen empirically depending on the observation of the experiments and can be defined as:

$$\begin{bmatrix} 1 & 0 & 0 & 0.005 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.005 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.005 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Then the Kalman gain matrix can be defined as: $\mathbf{K} = \bar{\mathbf{P}}\mathbf{H}^\mathsf{T}(\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^\mathsf{T} + \mathbf{R})^{-1}$

where $\mathbf{R}$ is the measurement noise covariance, is considered the prior and defined as identity matrix of 6x6 and $\mathbf{P} = (\mathbf{I} - \mathbf{KH})\bar{\mathbf{P}}$

Process and measurement uncertainty/noise matrix can be 6x6 identity matrix.

Measurement model which is presented in the matrix $\mathbf{H}$ can be represented by 2 horizontally stacked 3x3 matrices: First matrix is the follower's robot self velocity measurement coming from a feedback sensor as optical flow or other techniques. The second matrix is represented as 3x3 zeros matrix, since Yaw control is not considered in this chapter. This will build the standard predict/update loop which will iterate to provide $\widehat{\frac{\partial e}{\partial t}}$.

Relating back to [76], we showed that even in the leader's velocities scarcity in the framework according to a Lyapunov function can be uniformly bounded. In that paper, a velocity compensation makes best use of a double exponential smoothing to have a prediction of the leader's velocity. The velocity error estimation presented in this paper can be compared to that compensation.

4.5 Simulation

4.5.1 Preliminaries

- Three UAVs are considered, one leader and two followers.
- The aggregation phase between the robots is not considered.
- UAVs are equipped with a good low level controllers, e.x. open source px4 stack. The control inputs of our concern can be summed up in vertical, lateral, linear and angular velocities.

- Each robot is equipped with embedded monocular perspective camera which can provide leader robot detection through tracking a fiducial marker like Apriltag [1], or ARtag.
- The algorithm should be initialized by posing the leader in the desired pose with regard to the follower. Then after capturing the image, the desired image moment will be deduced from it. That method can be considered as teaching by showing.

4.5.2 Design

In this section, we present the simulation results for both the Gazebo/ROS simulator. The model of the UAV is a available to the scientific community as a ROS package [67], namely the AscTec Hummingbird model was used. We have mounted a perspective camera on the follower robots and fiducial marker as a tag on the leader to facilitate the detection of it. It has to be noted that, the algorithm is marker agnostic. It is only required to have 4 discrete points, that represent the shape that should be tracked (in our case for simplicity of experiments it was the corners of the marker). The desired image moments with respect to the detected one as seen from the follower UAV can be viewed in figure 4.3. The desired moments are in green and the detected ones are in red.

Camera view of the follower UAV observing the leader UAV is shown in the figure 4.3. Image moments visualization is emphasized. The desired features location is in green, and the detected features are in red which are the corners of the marker.

4.5.3 Scenario

The values of the simulation model parameters used, are the following:

- Simulation time: 50s starting from 280 till 320 seconds in figure 4.5, and from 420 till 470 in figure 4.4. The results were recorded to a rosbag file, then the plots where generated continuously, that is the reason behind different time stamps which can be normalised to start from 0 seconds and end at 50 seconds.
- Camera mounted distance from the center of UAV according to z-axis = -0.15m according to x-axis = 0.15m
- Altitude: 7m
- Maximum allowed velocity: 0.5 m/s

The initial conditions of the quadrotors are:

- Take off positions (x,y,z) in meter:
  - leader : 0,3,0; follower 1 : 0,1,0 ; follower 2 : 0,-1,0
- Relative positions (x-spacing,y-spacing, z-spacing) between the different followers and the leader are represented in meters as follows:
  - follower 1 at 2,1,0 ; follower 2 at 2,-1,0

which is done by posing the leader in the desired spacing then capturing the image of the follower's camera, which will then image moments will be extracted and considered as features.

[1] AprilTag 2: Efficient and robust fiducial detection, Wang et al, IROS 2016, `https://github.com/AprilRobotics/apriltag`

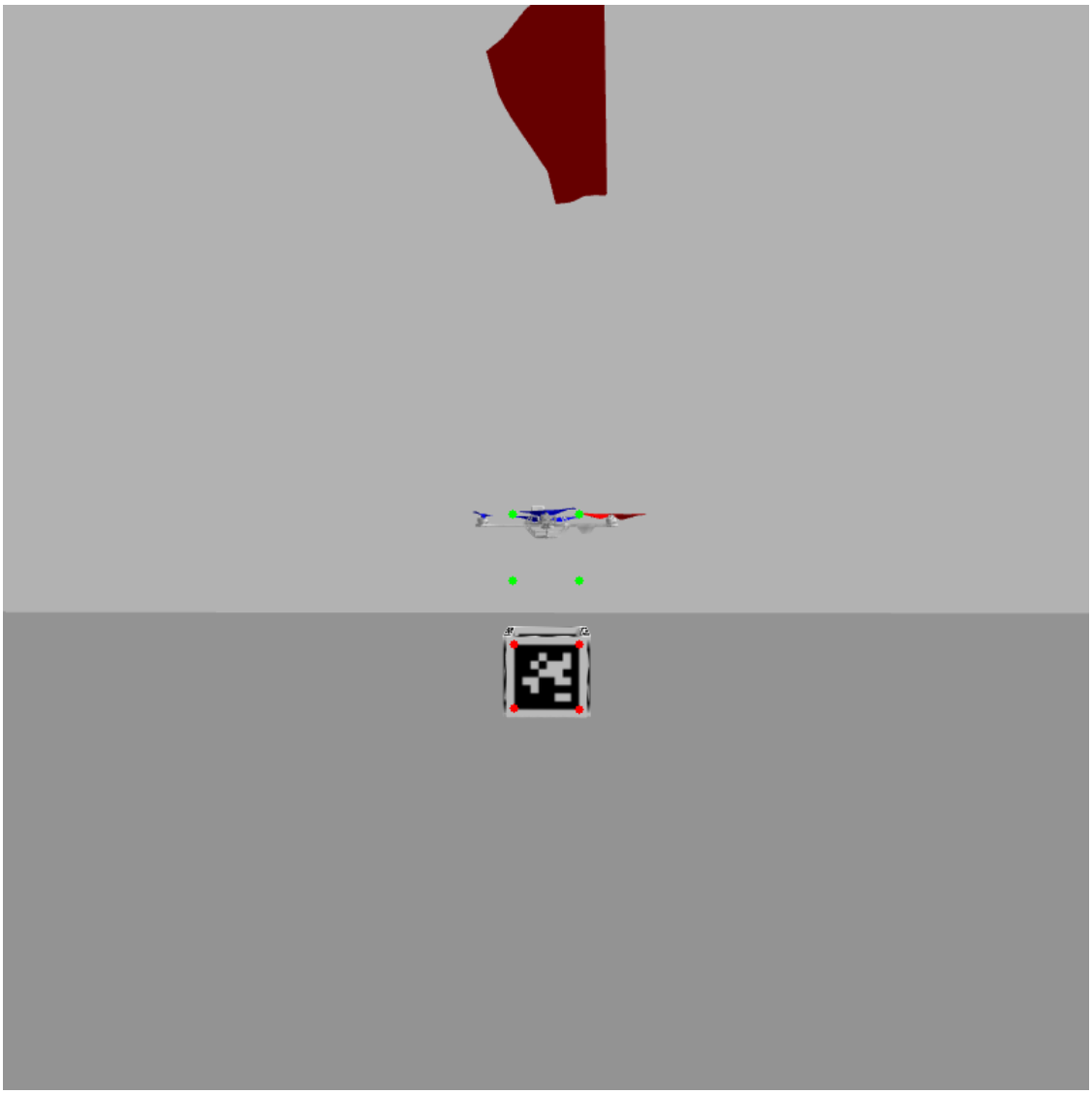


Figure 4.3: Image Moments desired vs detected.

4.5.4 Simulation Results

Results obtained from the Gazebo simulation are reported in this section with two main analysis. Absolute distance error in figure 4.6. Another important aspect is the error according to the X,Y,Z of the two follower UAVs as shown in figures 4.4,4.5.

It shows consistent followers tracking to the same desired commanded velocities eventually after their take off throughout the whole trajectory up till the landing phase.

In the figure 4.6, to validate the algorithm and analyse its effectiveness, a logging of the absolute spacing distance ($\sqrt{x^2 + y^2 + z^2}$) between the leader and a follower UAV is being studied. N.P: The

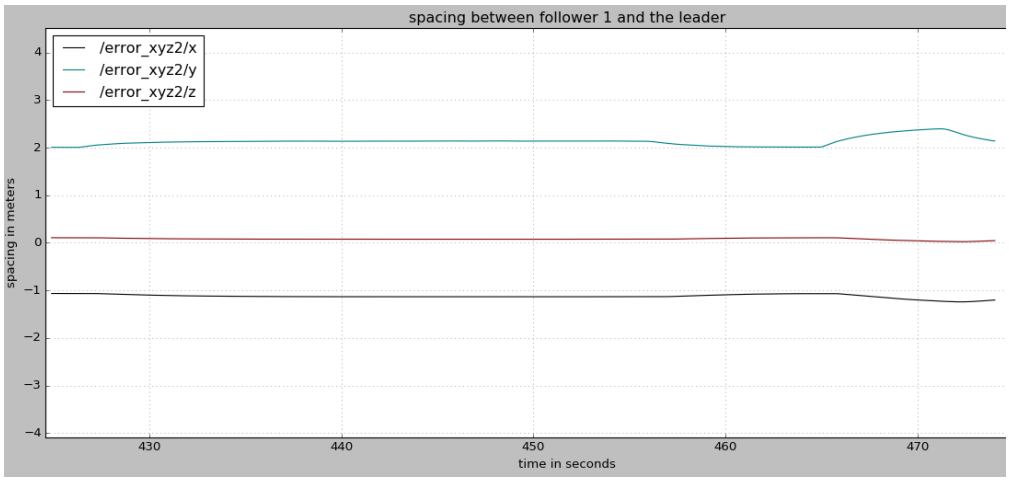


Figure 4.4: Distances spacings according to the desired (X, Y, Z) between robot 2 "considered as follower 1" and the leader. The final phase starting at 465 seconds is for landing

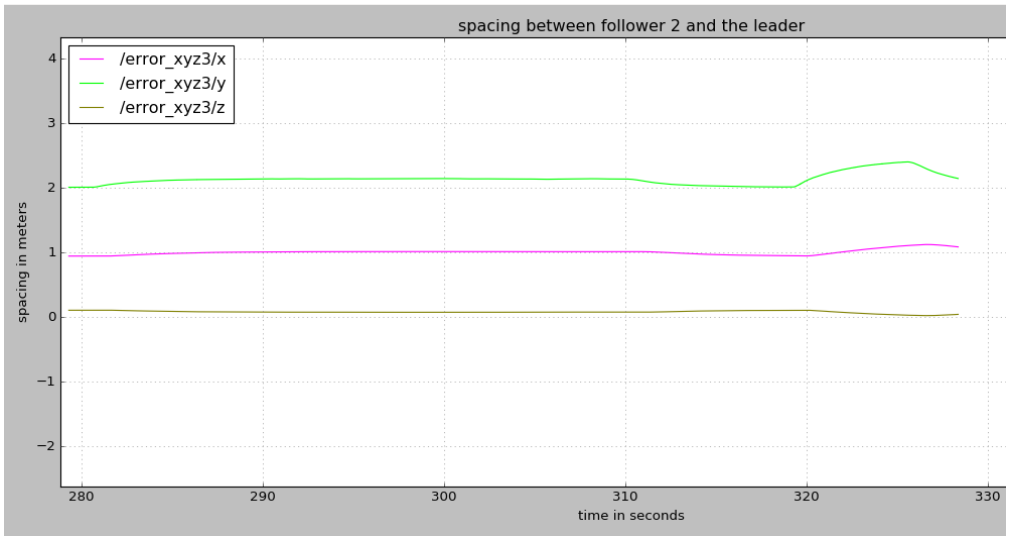


Figure 4.5: Distances spacings according to the desired (X, Y, Z) between robot 2 "considered as follower 1" and the leader. The final phase starting at 465 seconds is for landing.

distance spacing is not used by the controller. The experiment starts with leader velocity of -0.3 m/s in the absolute y direction of the world. The follower will maintain a spacing distance of 1.75. Then the leader will be commanded to stop, so a velocity of zero was commanded at around 142 seconds. The follower UAV as shown will immediately react trying to achieve the same distance spacing. The prediction phase oscillates and then achieve the proper spacing. To tackle realistic scenarios, and test the robustness of the predictor. Even before it settle down to the appropriate spacing, the leader was commanded velocity of 0.3 m/s at around 150 seconds. As shown the algorithm was able after

some minor oscillation to achieve the desired spacing. Then at around 165 seconds the leader was commanded with -0.3 m/s in the absolute Y of the world which according to 4.1 is in the same direction as the forward looking axis of the camera. Again and lastly the system was able to recover such variation in velocity and direction (from forward 0.3 m/s to backward 0.3 m/s of the leader). As observed the oscillation is more apparent due to the toughness of such scenario.

In the graph represented by the figure 4.6, the absolute distances with regard to the changing velocities can be shown. Absolute distance spacing between robot 2 "considered as follower 1" and the leader is analysed over time. The green line represent the commanded velocity to the leader. The follower robot has no knowledge of that velocity as there is no communication. The magenta line is the separating distance between leader and follower. As shown the distance is always around the desired 1.75 meters spacing. There exist oscillations at the time of changing the velocity of the leader.

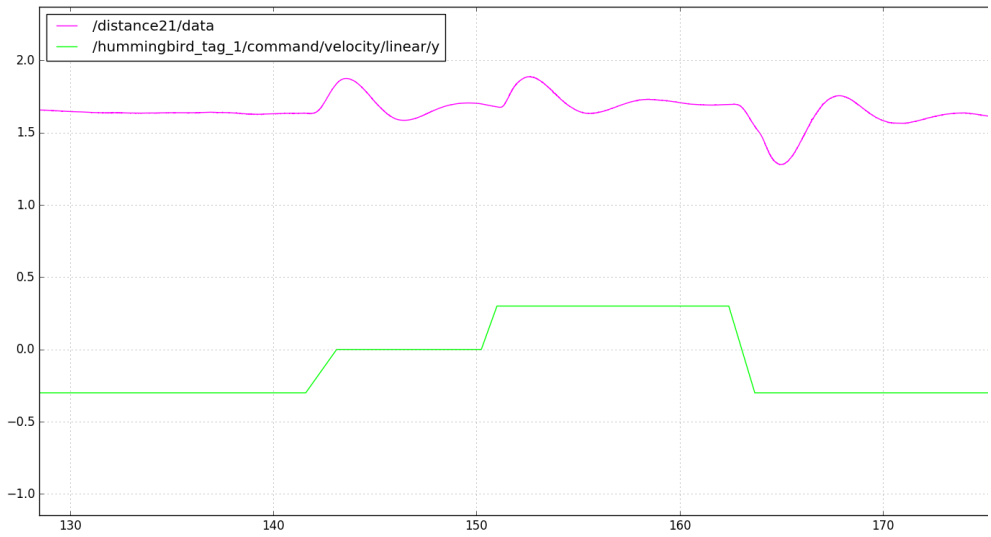


Figure 4.6: Absolute distance spacing vs velocities commanded.

4.6 Experimental Results

To assert the IBVS predictive formation controller, a similar setup as the experiments in the previous chapter, is considered.

4.6.1 Description

As a sum up of such system, the quadrotor is considered as the follower, the omni-directional ground mobile robot as the leader which will be commanded directly. The quadrotor follower should maintain an error that is near zero. More information about those particular robots can be found in appendix C.

4.6.2 Procedure

The experiments started with the quadrotor placed on the ground randomly apart from the the mobile robot. Then the drone takeoff phase starts, after sanity check of the sensors. Following that phase, the formation control algorithm is activated. The scenario of that experiment was taken in one shot but with varying leader's velocities mainly in the global Y direction which induces the error in x axis of the image.

In the figure 4.7, a drone as a follower robot, omni directional mobile robot with a marker attached on a mounted tower as a follower, ground features for the use of optical flow technique, an operator that operates the leader using manual commands. The drone is following the marker mounted on the mobile robot profiting from the velocity commands outcome from the EKF.

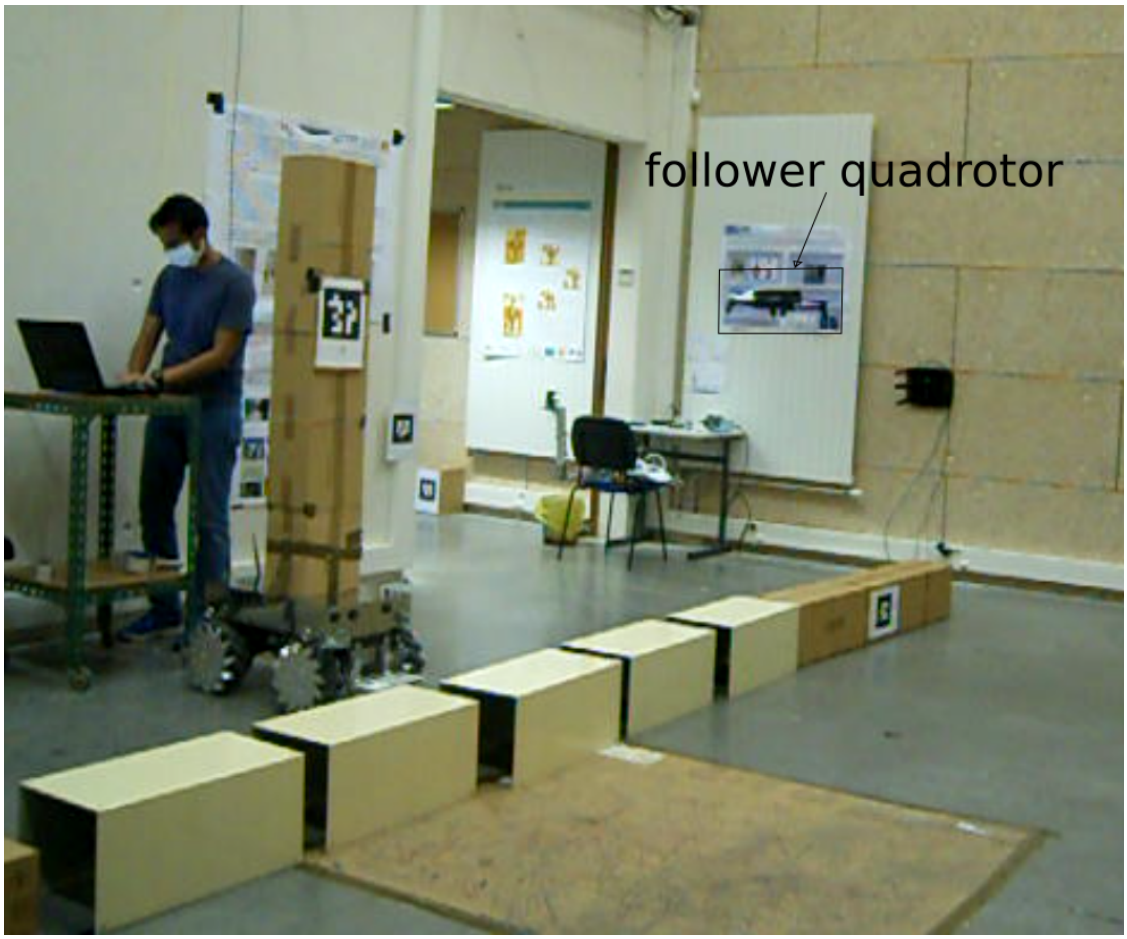


Figure 4.7: Real experimental setup

Results

In the figure 4.8, the result of a simple scenario is shown. After the drone stabilizes itself in front of the Leader mobile robot. The leader start to navigate with 0.1 m/s in the global Y direction (according to world coordinates), the drone detects that motion and the controller allows it to converge till the velocity of the leader is inverted. From the second 20, the leader will move with a velocity of -0.1 m/s. Again the drone was able to track that with a small fluctuation. The errors are in normalized pixels which spans from -1 to 1. In the sense the -1 according to X is the maximum points in the image view on the X-axis, and similar for all the axis. It is important to keep in mind absolute zero error is practically not very possible even in the indoor setup because the drone in its hover state is not ideally perfect.

Pixel errors of the IBVS tracking algorithm for formation control is shown in 4.8. Errors on the horizontal axis in normalized pixel values, and time is presented on the vertical axis.

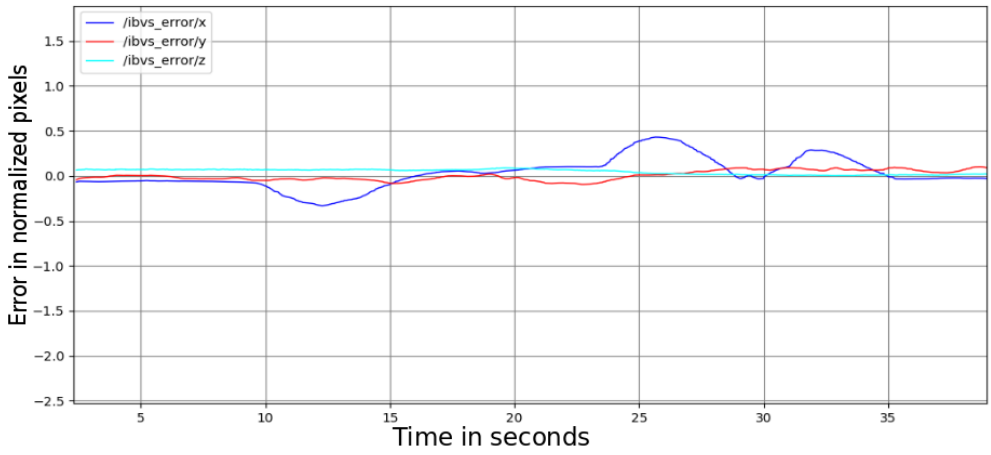


Figure 4.8: Pixel error vs time in LF framework.

Another experimental results as shown in 4.9 with similar context, but different commanding velocity for the leader robot. As shown the peak of the errors increased but the follower's controller was able to converge and track the leader. Particularly from the second 250 till 280 seconds, it represent the takeoff phase. Then from 280 till 310 seconds, the controller of the follower is activated but the leader didn't start to move yet. Later the leader was commanded velocity in the positive global Y direction from 310 till 325 seconds. Then a pause of 5 seconds from 325 till 330. Following this pause, the leader was commanded velocity of negative global Y direction from 335 till the end. The positive notch that happened around 350 seconds is considered as overshoot from the predicted state.

Link to one of the demo videos for more clarification of the setup and the experiment is in: `https://vimeo.com/431051702`.

Another demo video can be found in `https://www.youtube.com/watch?v=xQrFoV1jpc8`

Another important demo which shows side by side, the first man view of the drone following the robot and on the right hand side the image feedback as streamed onboard of the drone: `https:`

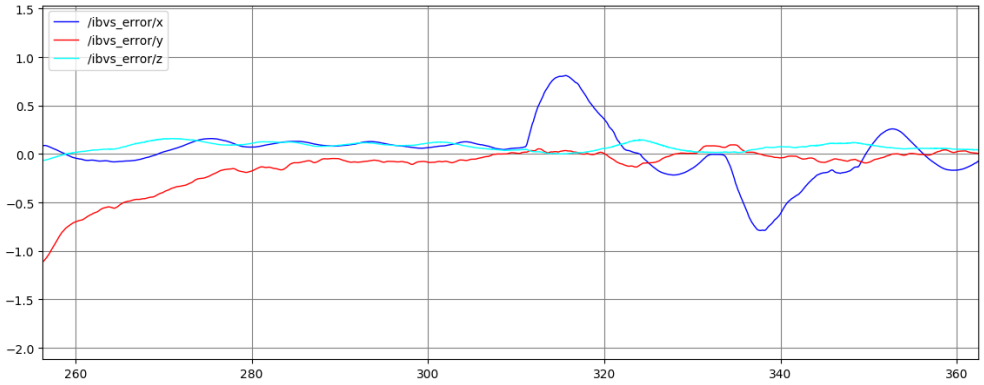


Figure 4.9: Pixel error vs time in LF framework at faster velocity.

`//www.youtube.com/watch?v=Iy5bhXGhiYE`. The results of the last aforementioned video can be shown in the figure:

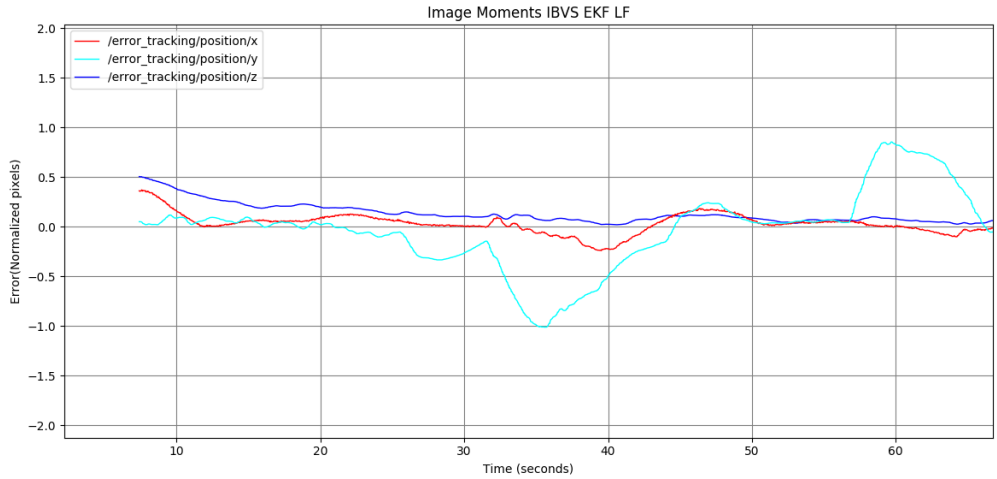


Figure 4.10: Pixel error vs time in LF framework.

4.7 Conclusion and Perspectives

In this work we focused on the problem of building a model for formation of UAVs and maintaining this formation under the constraint of using only on-board local sensing measurements to detect image moments features. The use of such features, omits the need for explicit distance measurement, which can not naturally be done by monocular cameras without sophisticated algorithms. No infrastructure is required for the controller to be operational, such as GNSS outdoor or an indoor motion capture system. The proposed solution is based on a Leader-Follower scheme.

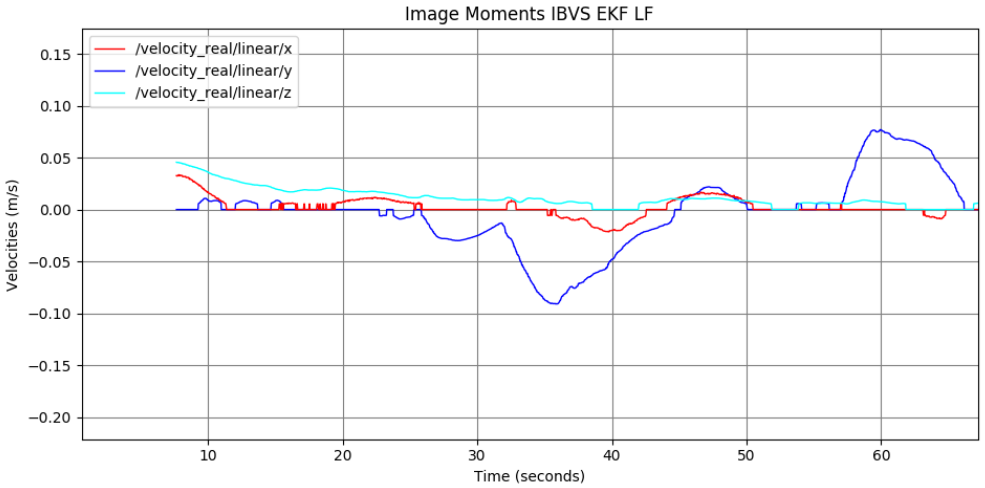


Figure 4.11: Velocities output of the drone vs time in LF framework.

The reported results, obtained from the simulations of Gazebo are written in python and C++ under ROS, and show that the formation is maintained during the navigation mission. Real experiments corroborated these simulated results. The hardware setup was utilizing Bebop robot where the system of ROS is offboard. The quadrotor communicate its data through ROS wireless network to that offboard computer. That makes the system prone to drop in communication or noise, a future better hardware setup is designed based on DJI F450 quadrotor airframe equipped with px4 low level controller stack on pixhawk cube flight controller unit, with the offboard computer connected through serial link and perspective camera.

The future perspectives are to test Reinforcement Learning vs the classical method proposed in this chapter. It was noticed as well that the EKF tuning was time consuming. The investigation of UKF is very much recommended to tackle such pitfall. As well as, UKF is usually in literature able to tackle non linearity better than EKF.

CHAPTER 5

Alignment of Three Robots without Communication nor Localization

Contents

5.1 Introduction

This chapter presents a decentralized method for aligning three robots without communication and without explicit localization. Each robot is assumed to be equipped with an omnidirectional camera and is assumed to be able to detect the two other robots in the images produced by its own camera. Each robot decides of its own movements based on the continuous measures of angles between itself

and the others. We present the algorithm that performs such a task, its theoretical analysis and some simulations. We prove the correctness of the algorithm when the robots are reduced to points, when collisions are not considered and when the initial formation is not an equilateral triangle. When collisions are considered, the probability for the algorithm to be successful is greater than $1 - \frac{1}{36}$. We then present some simulations for measuring the success of the approach when a minimum security distance d_{sec} has to be respected. During the execution of the algorithm by each robot, if the distance between any two robots is lower than this security distance we suppose that a collision is likely to occur and consider that the Algorithm has failed. Simulation results report that the larger d_{sec}, the bigger the percentage of failures. However runs are always successful when $d_{\text{sec}} = 0$ which suggests that the theoretical bound is not tight and could be improved. In addition, simulations reveal that the main source of failures was not the one expected by the theoretical analysis.

5.2 Related work

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

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

Sugihara and Suzuki were among the first to propose a distributed method for controlling multiple mobile robots [80]. In their work, the authors address the generic problem of the geometric pattern formation, including line formation. The robots are considered as identical and they are not supposed to communicate with each other. The method developed by the authors is not fully autonomous since the user, for the line formation, has to explicitly choose the two robots that will be the extremities. In addition it is assumed that each robot is able, thanks to a sensor, to determine the position of the other robots, in its own reference frame. The model used in this work was more formally defined in [81]. The robots are assumed to be reducible to a point in 2D space. Each robot is anonymous, has a memory, and its own coordinate system. In addition, after observing its environment the position of the other robots are known in its own coordinate system. The authors show in particular that oblivious robots can achieve the task of gathering altogether in a single point of the plan, a problem also addressed with mainly the same model in [82]. In the same line, in 2001 Paola Flocchini and her colleagues refined and constrained the model, but after the observation phase, a robot knows the position, in its own coordinate system, of the robots located in its surrounding. If some assumptions regarding the model are discussed in other papers, obliviousness, sense of direction, chirality, etc. [83, 84], the localization knowledge after the observation phase is never questioned.

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

One of the founding work is due to Balch and Arkin [85]. They propose a method for building different topologies: line, column, diamond and wedge. Every proposed approach requires the knowl-

edge of the position (using GPS coordinates or dead reckoning) of some other robots. The methods consist for each robot to maintain its position with respect to a particular point that could be either, the central point of the formation or, the position of a leader or the position of a specific neighbor. Other approaches include the computation of mutual localization which is mainly obtained through information exchanges between robots. This mutual localization can also be achieved based on the use of bearing angles, as it was exposed in [86], but in that case the robots have to communicate with each other. In [87], based on some prior information about the robots height and other items in the environment, each robot performs a distance estimation and run some control law to form some specific pattern from any initial configuration. This is achieved without the need for the robots to communicate with each other. The harsh underwater conditions for both wireless communications and mutual localization have led Sousselier and his colleagues to propose a line formation algorithm based on local localization information using ultra-wave ping. Their swarm is supposed to work in a synchronous way.

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

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

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

The Alignment Algorithm is described in the next Section. We theoretically prove, in Section 5.4 that this algorithm is successful 97% of time. Simulations results are presented and analyzed in Section 5.5. Finally some directions for improving and extending the current method and some perspectives are discussed in the Conclusion.

5.3 Alignment of Three Robots without Communication nor Localization

5.3.1 Introduction

Given three robots as in Figure 5.1 located on the same plane. As previously mentioned, each robot is assumed to be able to measure the angle it forms with the two other robots which corresponds to angle γ on the Figure.

The γ angle is the key element on which the alignment algorithm is built on.

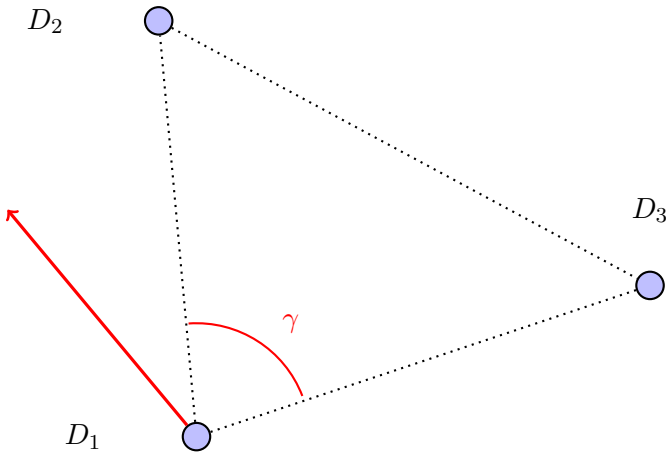


Figure 5.1: Arbitrary position of the three robots. Each robot is able to measure its γ angle, the angle it forms with the two other robots.

5.3.2 Alignment Algorithm

The algorithm presented in this section is independently executed by each robot and ends only when the alignment is obtained. It can be considered as a continuous process for reaching a predefined state, this state beeing a line. From the point of view of any robot, a line is obtained as soon as its γ angle is equal to 0 or to π. In this latter case, the robot is located between the two other ones.

Before reaching this desired state, at any moment the moving behavior of a robot is driven by both γ value and $d\gamma/dt$. This means that the movement of a robot may change if the value of γ is changing because of its own movements or because of the movements of the other robots (as illustrated by Figure 5.2):

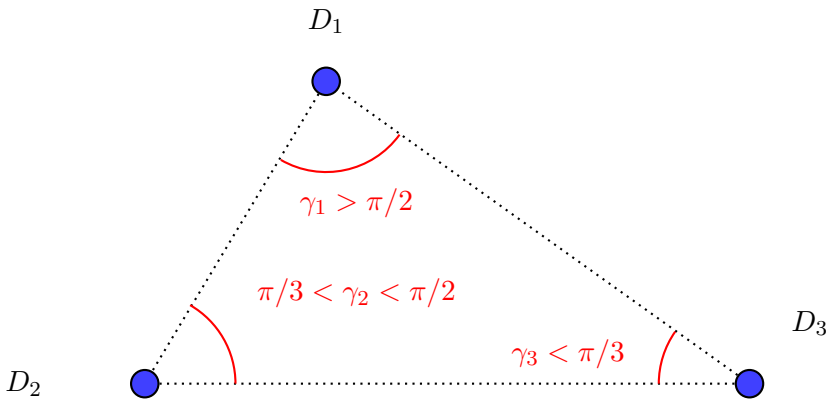
1. if $\gamma = 0$ or $\gamma = \pi$, the robot stops,
2. if $\gamma \geq \pi/2$, the robot moves into a direction corresponding to the bisector of this angle,
3. if $\gamma \leq \pi/3$ the robot stops moving,
4. if $\pi/3 < \gamma < \pi/2$, the robot moves into the direction of one of the two other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

5.4 Theoretical Analysis

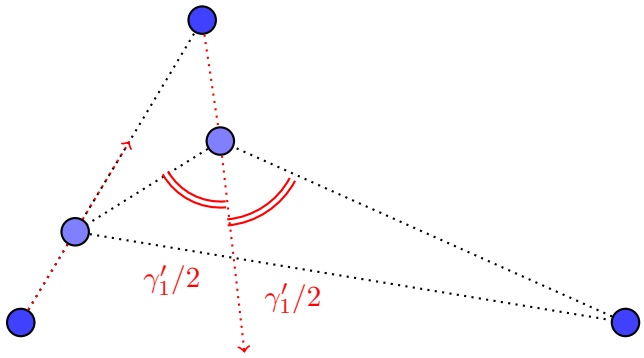
In this section we propose a theoretical analysis of Algorithm 1. We always consider that robots are reduced to points but we distinguish two possibilities, when collisions are considered (at least two

Algorithm 1 Alignment Algorithm

```
1: procedure ALIGN
2:     state ← INIT
3:     D⃗ ← NULL
4:     while ((γ ≠ π)&&(γ ≠ 0))&&(state ≠ STOP) do
5:         if (γ ≥ π/2) then
6:             if (dγ/dt ≥ 0) OR (state = INIT) then
7:                 move in the direction of the bisector of γ
8:                 state ← MOVE
9:             else
10:                state ← STOP
11:        else
12:            if γ ≤ π/3 then
13:                state ← STOP
14:            else
15:                if (dγ/dt ≥ 0) OR (state = INIT) then
16:                    if D⃗ ≠ NULL then
17:                        T ← robot randomly chosen
18:                        D⃗ ← direction towards T
19:                        state ← MOVE
20:                    move according to D⃗
21:                else
22:                    state ← STOP
```



(a)



(b)

Figure 5.2: Initially, D_3 will not move since $\gamma_3 < \pi/3$, D_1 starts moving in the direction of the bisector of its γ value since $\gamma_1 > \pi/2$ and D_2 starts moving in the direction of one of the two other robots (D_1 is randomly chosen in the present case), since $\pi/3 < \gamma_2 < \pi/2$ (a). While $d\gamma_1/dt \geq 0$ and $d\gamma_2/dt \geq 0$, both D_1 and D_2 keep on moving (b). Note that γ' denotes the new value of the γ angle.

robots are located at the same position) or not.

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

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

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

In the sequel we only consider non equilateral triangles.

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

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

All possible combinations are:

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

Based on the algorithm, and due to triangle properties (there cannot be three angles strictly greater than $\pi/3$) at most two robots may move at the same time, thus D_3 is not moving since γ_3 is the smallest γ angle.

For combinaisons 2 and 3 and according to the algorithm, only D_1 moves which leads to Scenario 1 on Figure 5.3. For the case 4, D_1 is moving in the direction of the bisector of its γ angle and there exist two possibilities for the movement of D_2. Either D_2 moves in the direction of D_3, leading to Scenario 2 on Figure 5.3, or in the direction of D_1 corresponding to Scenario 3. Finally, for the last case (5), there exist three possible combinations of movements: (i) D_1 and D_2 are moving in opposite directions: Scenario 4, (ii) only one of the two robots, D_1 or D_2, is moving to D_3 leading to Scenario 5, or (iii) both D_1 and D_2 are moving in the direction of D_3, corresponding to Scenario 6. □

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

Proof. When $\gamma_2 = \gamma_1$, the triangle is isosceles. For scenario 4, if D_1 and D_2 start simultaneously and are moving exactly at the same speed then, both γ_2 and γ_1 angles increases up to $\pi/2$ and both robots will be positioned at the same point. If collision is not an issue, we can consider this final configuration as an alignment since the three robots belong to a same line. For scenario 6, both robots are moving toward D_3 and, if D_1 and D_2 start simultaneously and move at the same speed, both γ_2 and γ_1 remain identical during the move. The movement stops when the three robots are positioned at the same point. Which may also be considerated as a degenerated line. If collisions are considered, in both situations, as at least two of them are located at the same position, we can consider that they collide and thus that the algorithm fails. □

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

Proof. For Scenario 1, $\gamma_1 > \pi/2$, only D_1 is moving and it moves in the direction of the bisector of its γ angle which entails a decrease in both γ_2 and γ_3 angles as illustrated by Figure 5.4(a) and an increase of γ_1, thus $d\gamma_1/dt > 0$. After some time, γ_1 reaches π and simultaneously $\gamma_2 = 0$ and $\gamma_3 = 0$, thus

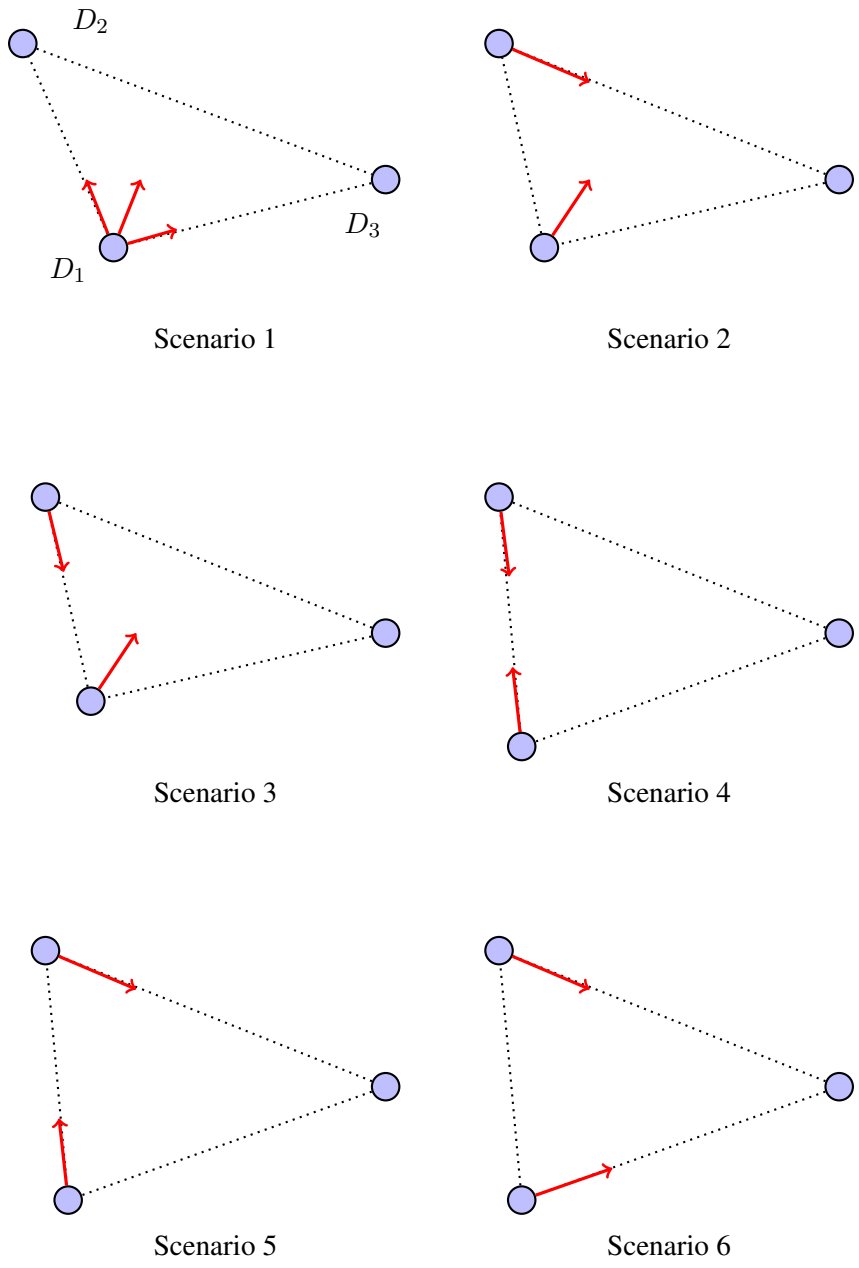
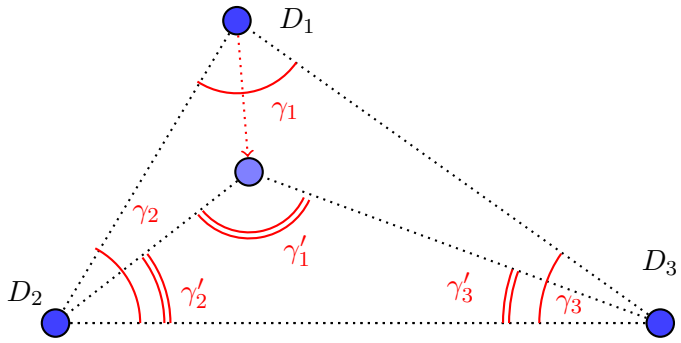


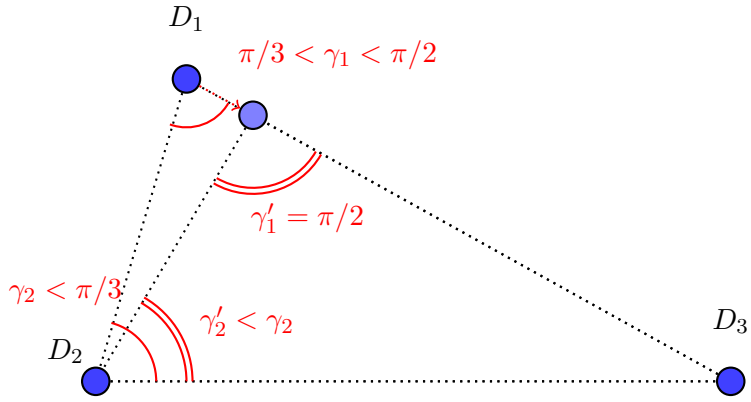
Figure 5.3: The six different moving scenarii according to Algorithm 1.

every robot stops the execution of its own algorithm. Thus for this case the algorithm is successful since it converges to the alignment of the three robots. When $\pi/3 < \gamma_1 < \pi/2$, either D_1 moves in the direction of D_2 entailing a decrease of γ_3, or it moves in the direction of D_3 entailing a decrease of γ_2. Without loss of generality let us suppose that D_1 is moving in the direction of D_3. When D_1 moves

in the direction of D_3, $d\gamma_2/dt < 0$ and, as $d\gamma_3/dt = 0$ then $d\gamma_1/dt > 0$. In addition, $\gamma_3 < \pi/3$, thus after some time $\gamma_1 > \pi/2$ leading to the previous situation, as illustrated by Figure 5.4(b). As a consequence, for scenario 1, whatever the initial value of γ_1, given that $\gamma_3 \leq \gamma_2 < \pi/3 < \gamma_1$, the execution of the algorithm on every robot leads to the alignment of the three robots. □



(a)



(b)

Figure 5.4: Scenario 1. Case (a) $\gamma_1 > \pi/2 > \pi/3 > \gamma_2 > \gamma_3$ and case (b) $\pi/2 > \gamma_1 > \pi/3 > \gamma_2 > \gamma_3$

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

Proof. Given the assumptions that robots are starting to move at the same moment and that they are moving at the same speed, the evolution of Scenario 3 is illustrated by Figure 5.5. In this scenario, $\gamma_1 > \pi/2$ thus D_1 is moving in the direction of the bisector of his γ angle, while $\pi/3 < \gamma_2 < \pi/2$ and D_2 is moving in the direction of the initial position of D_1. As D_3 is motionless, as soon as D_1 and D_2 are moving, $d\gamma_1/dt > 0$ and $d\gamma_2/dt < 0$, leading D_2 to change its state from MOVE to STOP. The resulting situation, only D_1 is moving, corresponds to Scenario 1.

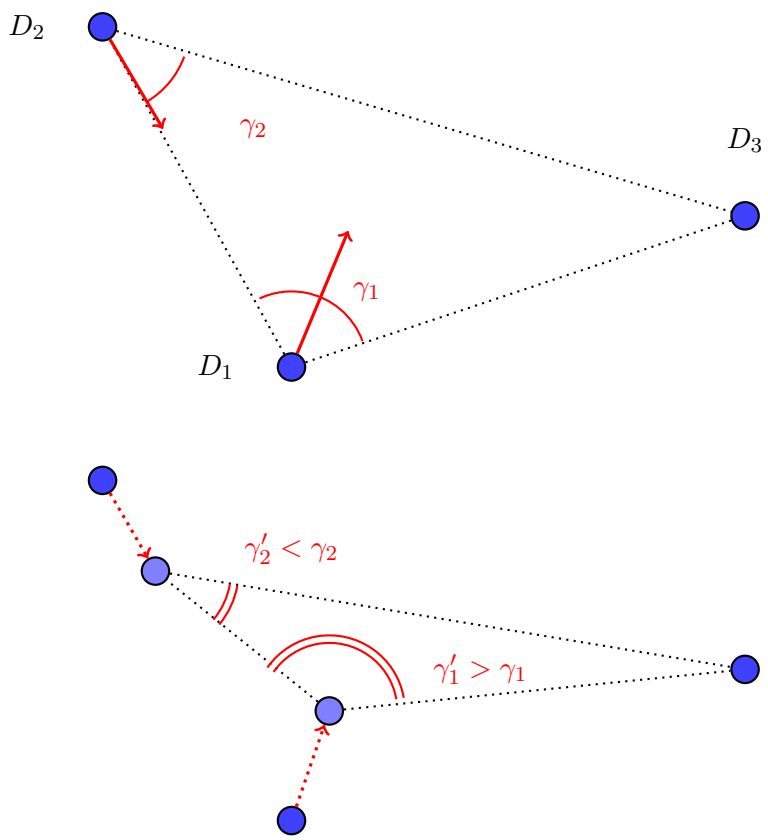


Figure 5.5: Scenario 3. Initially $\gamma_1 > \pi/2$ and $\gamma_2 > \pi/3$. When D_1 and D_2 are moving, γ_1 increases while γ_2 decreases, then according to the algorithm D_2 stops, leading to a situation corresponding to Scenario 1.

The evolution of Scenario 4 is described in Figure 5.6. Initially $\pi/3 < \gamma_2 < \gamma_1 < \pi/2$. D_1 and D_2 are moving in the opposite direction, but as D_3 is not moving, both $d\gamma_1/dt > 0$ and $d\gamma_2/dt > 0$. After some time, γ_1, which is greater than γ_2 reaches $\pi/2$. At that point, the direction of D_1 changes to the bisector of γ_1, a situation corresponding to Scenario 3 which evolution leads to Scenario 1 as it was previously proved.

The evolution of Scenario 5 is illustrated by Figure 5.7. The robot moving in the direction of the motionless robot (D_3) has its γ value increasing. On the contrary, the other robot has its γ value decreasing entailing a change in its state from MOVE to STOP. The situation corresponds then to Scenario 1, where only one robot is moving.

The evolution of Scenario 6 is illustrated by Figure 5.8. Both robots are moving in the direction of the motionless robot (D_3). As $\gamma_1 > \gamma_2$, the distance between D_1 and D_3 is lower than the distance between D_2 and D_3, then, as soon as they are moving, γ_1 increases while γ_2 decreases leading to a

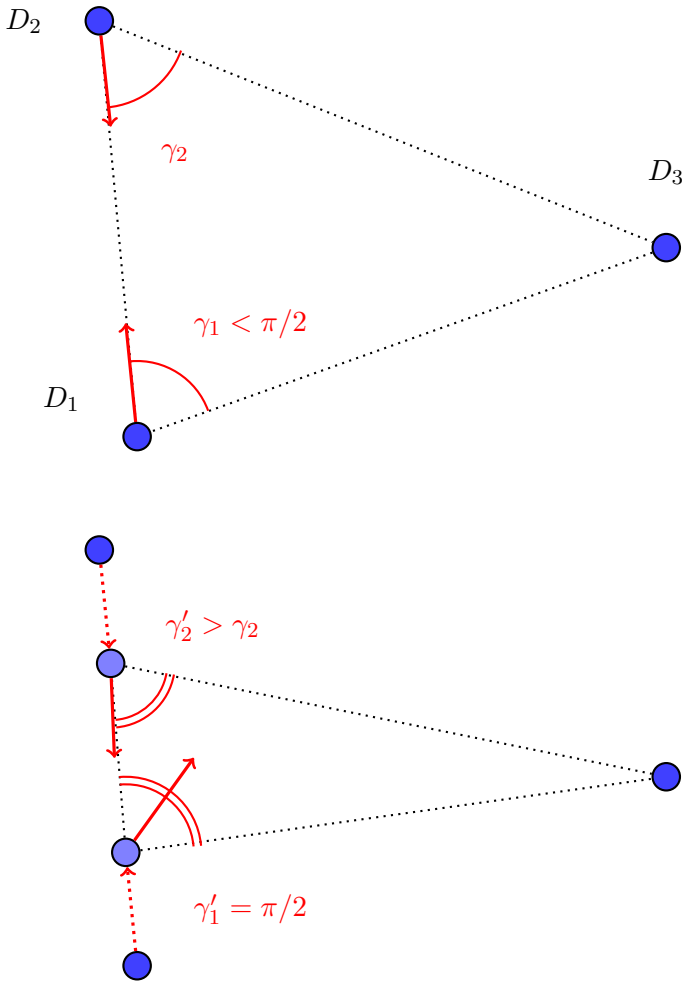


Figure 5.6: Scenario 4. Initially $\gamma_1 > \pi/3$ and $\gamma_2 > \pi/3$ and during the movement $d\gamma_1/dt > 0$ and $d\gamma_2/dt > 0$. After some time, $\gamma_1 > \pi/2$ leading to a situation corresponding to Scenario 3.

change in the state of D_2 from MOVE to STOP. From that moment, only one robot is moving, D_1, a situation corresponding to Scenario 1.

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

The last Scenario which has not been considered yet is Scenario 2. When both D_1 and D_2 are moving, as illustrated on Figure 5.9, it may happen that both $d\gamma_1/dt > 0$ and $d\gamma_2/dt > 0$, leading, potentially, to a situation where the two robots are positioned at the same point. In that case and if collisions are not considered, the situation is comparable to that described in Lemma 3 and the algorithm is successful. However, if collisions are considered, as two robots may be positioned at

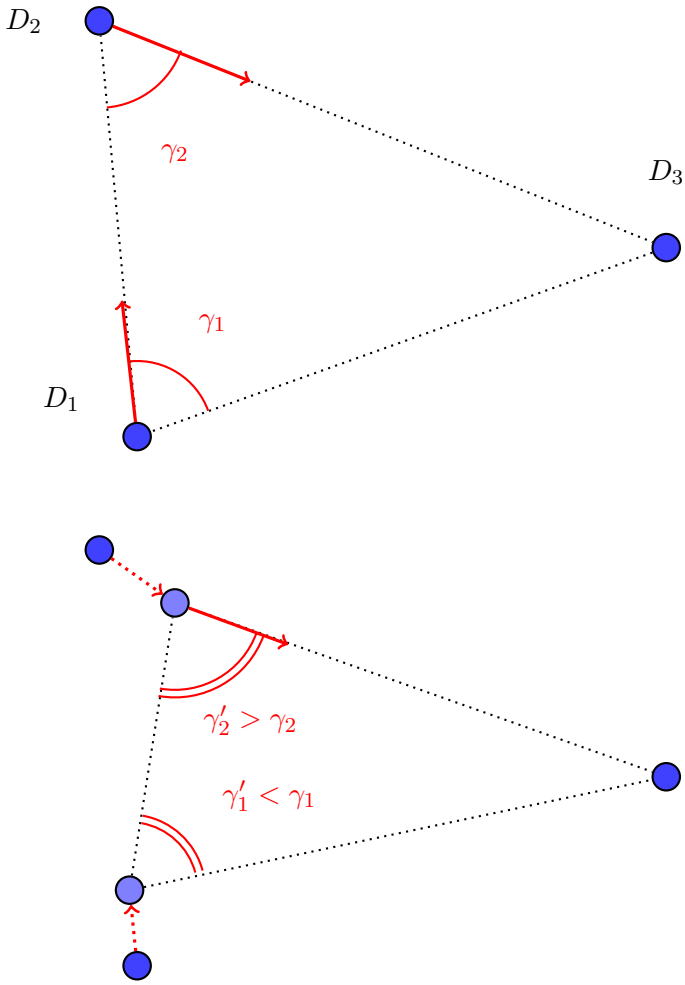


Figure 5.7: Scenario 5. Initially $\pi/3 < \gamma_1 < \pi/2$ and $\pi/3 < \gamma_2 < \pi/2$. As soon as D_1 and D_2 are moving, γ_2 increases and γ_1 decreases, leading to Scenario 1, a situation where only one robot is moving.

the same point, the algorithm may fail. A very detailed and careful analysis would probably help finding subcases for which this Scenario might be successful even for the collision case, however we want to find a lower bound for the probability of Algorithm 1 to be successful in that case, we thus consider that, for this scenario, the algorithm fails to align the three robots when collisions are taken into consideration.

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

Proof. Given three non aligned points, there exist only one circle that contains these three points. Without loss of generality we can consider that: the circle is the unitary circle, the point with the

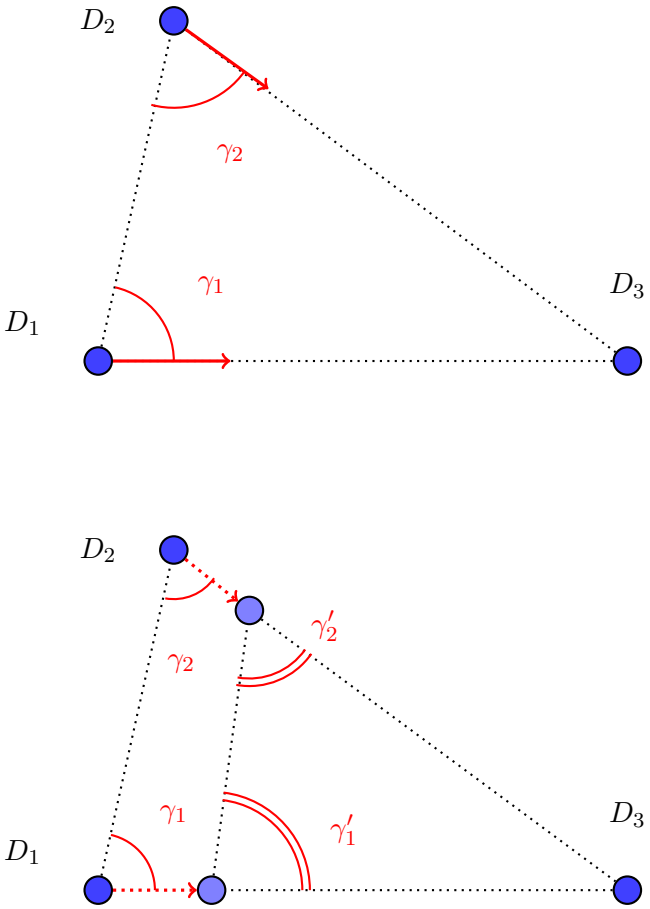


Figure 5.8: Scenario 6. Initially $\pi/3 < \gamma_2 < \gamma_1 < \pi/2$. As soon as D_1 and D_2 are moving, γ_2 decreases and γ_1 increases, leading to Scenario 1, a situation where only one robot is moving.

lowest γ value is D_3 the point with the largest γ is D_1 and that D_3 is located at $(1, 0)$. First note that as $\gamma_1 > \pi/2$, the three points have to be located on the same half-circle and that D_1 is located between D_2 and D_3. In addition, as $\gamma_3 < \pi/3 < \gamma_2 < \pi/2 < \gamma_1$ then $\gamma_3 < \pi/6$. One such situation is illustrated on Figure 5.10.

We note also that the position of D_2 is constrained by its γ angle that should be greater than $\pi/3$. Thus D_2 has to be positioned on the unitary circle between $2\pi/3$ and $4\pi/3$ as illustrated on Figure 5.11, otherwise D_1 cannot be positioned on the unitary circle between D_2 and D_3 without violating the constraint $\gamma_2 > \pi/3$.

Then, whatever the position of D_2 within that sector, the maximum sector in which D_1 can be positioned is $\pi/3$ as illustrated by Figure 5.12. This sector is maximum when D_2 is located in $(-1, 0)$ and reduces as soon as the position of D_2 gets closer to its extreme possible positions.

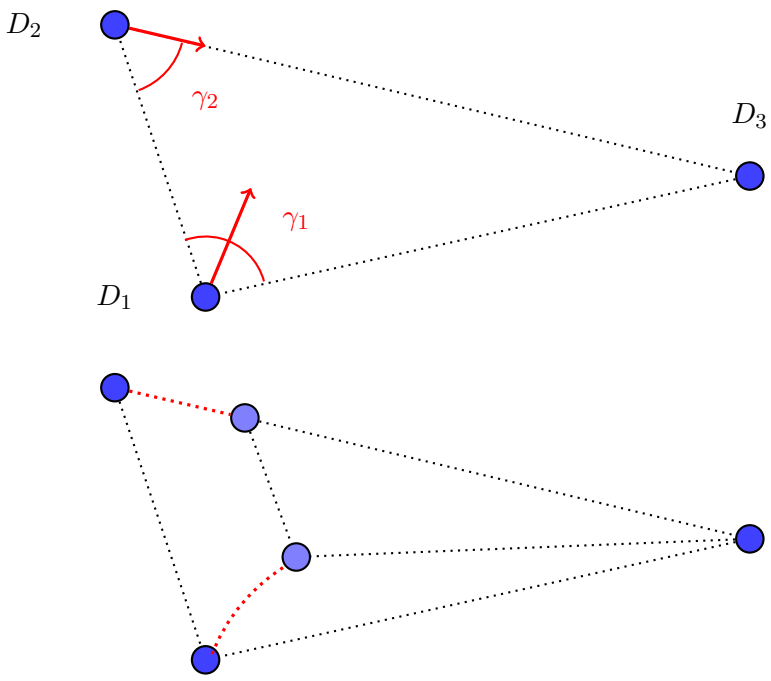


Figure 5.9: Scenario 2. Initially $\gamma_1 > \pi/2$ and $\gamma_2 > \pi/3$. When D_1 and D_2 are moving, both γ_1 and γ_2 may increase and the evolution may lead to a collision.

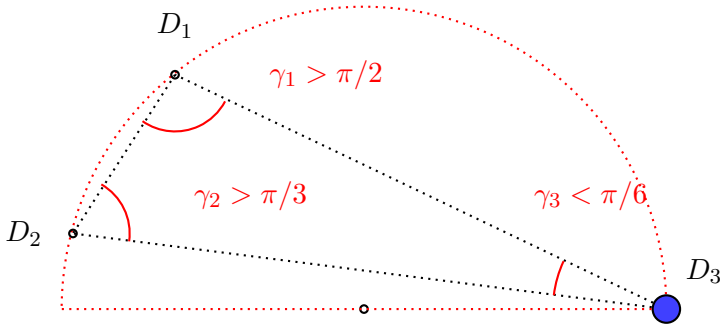


Figure 5.10: Illustration of a situation where all the constraints are met: $\gamma_3 < \pi/6$, $\pi/3 < \gamma_2 < \pi/2$ and $\pi/2 \leq \gamma_1$.

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

We are now considering the theoretical probabilty for the algorithm to fail according to the initial

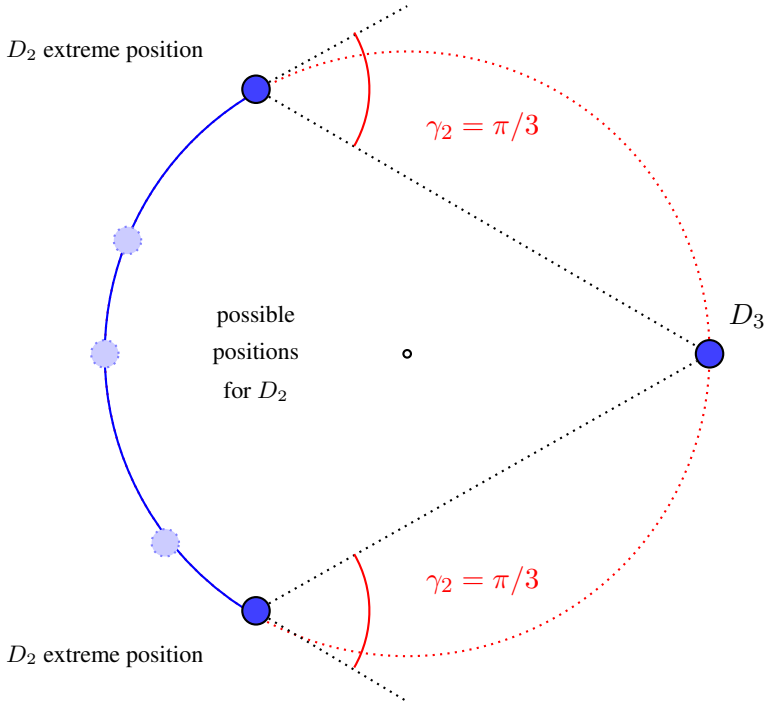


Figure 5.11: According to the constraint $\pi/3 < \gamma_2 < \pi/2$, the possible positions for D_2 are within the blue sector.

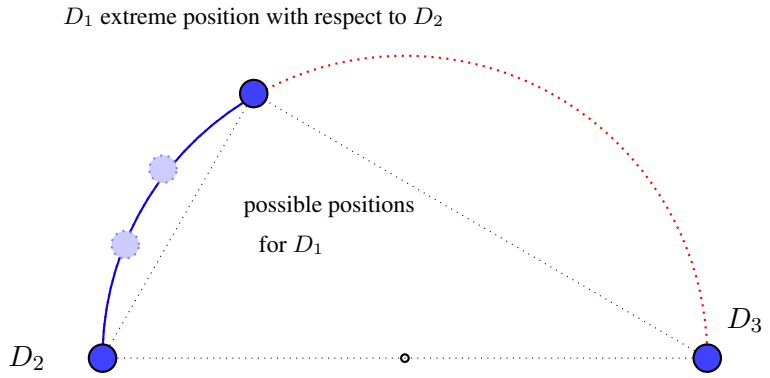


Figure 5.12: In order to respect the constraints: $\pi/3 < \gamma_2 < \pi/2 \leq \gamma_1$, the possible positions of D_1 are within the blue sector.

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

Theorem 1. *When collisions are not considered, Alignment Algorithm aligns the three robots, re-*

Algorithm 2 Aligning robots at the same altitude

```
1: procedure 3DTO2D()
2:     state ← GO_DOWN
3:     while state ≠ STOP do
4:         if (robot detection) then
5:             state ← STOP
6:         else
7:             if ground is reached then
8:                 state ← GO_UP
```

duced as points, iff the initial configuration is not an equilateral triangle.

Proof. Straightforward considering all the Lemmas. □

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

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

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

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

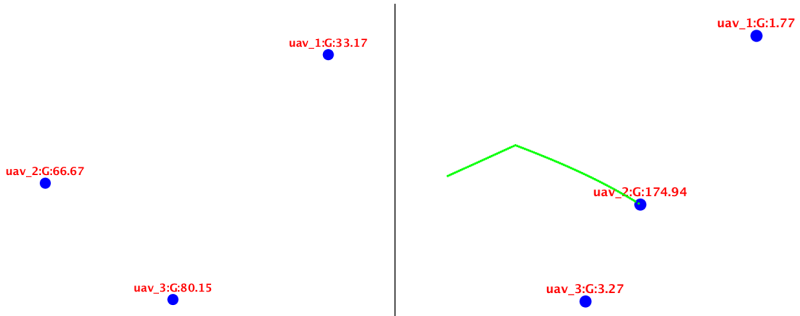
The second case corresponds to a subset of isosceles triangles. Given two points A and B in a 2D space, there are many different possibilities for the last point to form an isosceles triangle. Any point positioned on the bisector of the segment [AB] leads to an isosceles triangle as well as any point positioned on the circles which radius is equal to the length of the segment and the center is either A or B. However, both the bisector and the two circles are one-dimensional geometric objects. In comparison with the plane which is a two-dimensional object, the probability that the point C belong either to the line or to one of the two circles is again 0. So the probability that the algorithm fails because of case (ii) is 0 again. Remains the last case (iii) (Scenario 2 in Figure 5.3). According to Lemma 6, the probability to this scenario to occur is less than $\frac{1}{36}$. As a consequence, the probability that Algorithm 1 successfully aligns three randomly positioned robots in a 3D space is greater than or equal to $1 - \frac{1}{36} \approx 0.97$. □

5.5 Simulations

5.5.1 Simulation

For the simulations the environment is a square space of size $L \times L$. The three robots are randomly positioned within that space. The first simulations that were performed considered the theoretical

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



Initially, the robots are randomly positioned

After running Algorithm 1, the robots are aligned.

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

5.5.2 Results

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

The results presented in Table 5.1 are stemmed from the aggregation of all the runs for all combinations of parameters (with the notable exception of the case for which $d_{\text{sec}} = 0$.

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

Table 5.1: For each scenario: percentage of occurrences and percentage of unsuccessful runs.

The measures reported on the second line is the percentage of occurrences of each scenario according to the random initial positions of robots. Scenario 1 is far more frequent than the other scenarii

and it is the only scenario for which only one robot is moving.

The third line of Table 5.1 reports the percentage of unsuccessful runs for a given scenario while the values of failures for every combination of parameters are presented in Table 5.2. These results call for remarks. We first note that there is a noticeable difference between what was expected from the theoretical analysis and the results obtained from the simulation. Indeed, the analysis shows that Algorithm 1 should mainly fail for Scenario 2. From the simulation it appears that only 6% of the runs corresponding to Scenario 2 are unsuccessful, which represents a little bit more than $6\% \times 3.6\% \approx 0.2\%$ of the total number of unsuccessful runs. The main source of failures comes from Scenario 4, for which two robots are moving in their respective directions. Theoretically, since the probability of having an isosceles triangle is 0, a collision should not occur and the contribution of Scenario 4 to the total number of failures should be 0. However, in the simulation, because we want to use this algorithm for real experiments, we introduced a security distance that has to be respected between any two robots. And in the case of Scenario 4, if the difference of the γ angles of the two moving robots is small, after some steps the security distance is no more respected, thus a collision risk is supposed to be high, and the run is declared unsuccessful. This explain the 24% of unsuccessful runs for Scenario 4. In more detail we have 33.8% of unsuccessful runs for Scenario 4 when $d_{\text{sec}} = 15$, 24.5% for $d_{\text{sec}} = 10$ and 19.2% for $d_{\text{sec}} = 5$.

In Table 5.2 the percentages of successful runs for Algorithm 1 for each combination of the parameters are reported. As one hundred percent of unsuccessful runs are due to a collision risk, it is not surprising that the larger the environment and the lower the security distance, the better the results obtained by the algorithm.

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

Table 5.2: Percentage of success for Algorithm 1 according to the different combination of the parameters. Each cell of the table is the result of 100000 runs

These results have to be compared to the theoretical value of 97% computed in the previous Section when robots are reduced to points and show that the approach is relevant even for non-zero size robots. It is noticeable that for small environments and large security distances the percentage of unsuccessful runs can be a little bit greater than the theoretical value, but, as for the results of Table 5.1, the main source of failures is Scenario 4 and comes from the introduction of a security distance between any two robots that has to be respected during the simulation.

5.5.3 Gazebo

The algorithm was also developed on Gazebo within ROS [88, 89]. Such a simulator is better able to take into account more realistic constraints and is a step towards its implementation on a real platform.

The Gazebo simulator package ROTORS [67], provides realistic dynamic system to different family of multi rotors (quad, hexa, etc). It is widely used in the aerial robotics society for its high fidelity. The same codes can be transferred easily to the same type of robots mentioned in the package, and with some minor tuning to similar types of multi rotor vehicles.

A link for the demo of the algorithm can be found in: `https://www.youtube.com/watch?v=hC85JJcCE2o`.

The same demo in case of facing difficulty accessing youtube `https://drive.google.com/file/d/1HSzJs7w6RBY-iO4dj6iDgDo7jjo4bM5V/view?usp=sharing`.

5.6 Conclusion and Perspective

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

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

The perspectives of this work are manyfold. Current simulations were run using GraphStream, a dynamic graph library [90]. These results were corroborated with GAZEBO robotics simulator. In the future, the algorithm implementation should be tested on a fleet of UAVs (built upon F450 DJI frames).

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

assume the availability of an additional piece of information: the relative distance between robots, an information of the type: further, closer, similar distance.

CHAPTER 6

General conclusion

Summary of the Thesis

Autonomous Multi Robot System (MRS) are having beneficial tasks that could take place in our current world scenarios, yet MRS are not widely available to be deployed. As well, they are not easily scalable systems, as available solutions are centralized. Decentralizing the algorithms can bridge the gap of deployment, be scalable and efficient methodology for MRS to be available to fulfill tasks.

For a successful operation of MRS, two main fundamental capabilities are essential: relative location of nearby neighbors (formally known as mutual localization problem), and information flow between different peers through communication. These capabilities are going to be used by a relative collision avoidance module. A decentralized algorithm is utilizing such two capabilities in managing the fleet of robots for a common goal without the need of a centralized unit (neither for communication, sensing, nor for computation).

In case of communication failure such MRS would fail in accomplishing the defined tasks, and more worse robots in the fleet would collide together. That pitfall was a major motivation toward studying and finding methods that can act as a fail safe in the time of communication failure.

The main goal of this research and thesis was to explore the possibility of coordinating a group of robots that is constrained by intermittent communication in order to bring improvement in MRS robustness. The problem is diverse and interdisciplinary. The state of the art of these disciplines are vividly progressing globally by researchers, companies and practitioners. These advancements are fueled with the progress taking place in communication, sensing capabilities, and computing power.

Some examples of these advancements can be summed up as following. In communication field, there exist new technologies and research on 5G, LORA, etc. From the sensing capabilities point of view, there are conventional, unconventional imaging, time-of-flight, etc. Computation capabilities as well has seen widening use of GPU and miniaturization of these units that are now embedded in robots nowadays. Some research is going in the direction of spiking neuromorphic hardware. There are some research direction toward hardware acceleration using FPGA.

In this chapter a final review and wrap up are presented to what have been discussed in this thesis.

A perspective is finally given for the prospect directions that could be beneficial to be thoroughly investigated in future studies.

Contributions

These proposed algorithms were validated by different simulations, and corroborated with real world experiments.

High Level Decentralized Formation Controller

In chapter 3 a decentralized high level controller for formation flight control of UAVs was proposed. The main goal was to design a tuning-less high level controller able to maintain without tracking errors a Leader-Follower (LF) configuration in case of lack or degraded communications (latencies, loss ...) between the leader and followers UAVs. The high level controller only requires simple tunings and rests on a predictive filtering algorithm and a first order dynamic model to recover an estimation of the leader UAV velocities and avoid the tracking errors. The algorithm was tested on different simulators, and corroborated through a real world experiment. A quadrotor run the main algorithm tracking the leader which was represented by an omnidirectional mobile robot base with a mounted tower augmented by a fiducial marker.

IBVS using Image Moment for LF Formation control

In the work proposed in chapter 4 the focus was on an algorithm that could handle the LF configuration without the direct use of distance, relying mainly on bearing measurements. The bearing measurements are usually easier to be sensed by the ubiquitous sensors such as cameras. To achieve such system design, the use of the well known Image-Based Visual Servoing (IBVS) technique was exploited. While most IBVS techniques either require rigor camera calibration, the proposed approach avoids the calibration phase by relying on image moments features. In order to build and maintain a LF configuration of multi aerial vehicles (UAVs) without communication, a predictive component should be provided, to compensate for the missed crucial information. A classical extended Kalman filter was designed to compensate for the leader's robot velocity utilizing only the visibility of the leader. One important feature of the designed feature was that the follower robot's algorithm works in GNSS-denied conditions and can run using only on-board sensors without the need of absolute positioning. The method is validated in simulation and through experimental result with the same setup as the preceding chapter.

Alignment of three Robots

In the preceding two chapters, the MRS was designed around the LF configuration. As a matter of having a more naturally emerging algorithm, in this last proposition, an algorithm that rely only on the angles in between 3 robots, exploit the triangular to form a line. The algorithms didn't require any communication in between the robots. Only bearing measurements were exploited and required for such algorithm to succeed given the limited narrow cases in the chapter where it will fail. Such algorithm is tested in simulation. In the future could be tested practically, which eventually would

work given the appropriate equipment, since the simulator used was similar to the one used in the aforementioned proposed methods.

Future Perspectives

It would be much more recommended to use Unscented Kalman Filter (UKF) instead of the used Extended Kalman Filter. That would have positive impact on the predictive stage of the filter. Since it is a highly non linear state, UKF would be better handle it, than the EKF. A deeper theoretical analysis should be investigated

In the context of a LF MRS, another architecture for the estimation and prediction of the leader's velocity could be Model Predictive Controller (MPC) family of techniques. In the recent literature, MPC is being deployed in the robotics community.

There are numerous papers in literature that utilize Reinforcement Learning (RL) to do end-to-end autonomous flying policies. An important point is to test such policies instead of the classical hand crafted controllers. Another important study could be whether an end-to-end policy for multi system coordination possible in the current phase of scientific advancement. In this direction there are many problems to encounter like whether sim-to-real policy transfer possible or not. The efficiency of the policy to be able to run only using onboard sensing and computing power.

Since most of the real useful applications of a aerial MRS are outdoor, then the exploiting of the novel and emerging sensing capabilities should be studied as well. Among the direct and already studied systems are the papers [91] where ultraviolet Light-emitting diode (LED) are used as fiducial markers and camera with ultraviolet lens mounted on the follower quadrotor was used to detect such LEDs, then a tracking algorithm forms a LF configuration. The merge of such sensors with the algorithms developed in this thesis is very compatible with important benefits toward deploying in real world scenario.

While this thesis focused on autonomous solutions, the author is encouraging the wide adaptation of Human-In-The-Loop (HITL) in MRS. Currently there are researchers working on Human Robot Interaction (HRI), and some papers are valid within the framework on Human multi Robot Interaction. As well as teleoperation and bilateral teleoperation for MRS should be studied. This perspective can help shortening the gap between the researched algorithms and the deployment of such algorithms in real world scenario.

APPENDIX A

UAV Dynamic model

This section describes the dynamics of Drones using Newton's and Euler's equations. That would align much with the results of the paper and PhD thesis [92, 93] . As well as the described toolbox in [94]

The model can be described as follows. A vector of linear velocity and angular velocity. The linear frame's angular velocity is based on the frame of the body frame. The dynamics of the drones in relation to a world reference frame.

$[x \quad y \quad z \quad \phi \quad \theta \quad \psi]^T$ (linear/angular position)

$[u \quad v \quad w \quad p \quad q \quad r]^T$ (Inertial frame)

$\nu_B = [u \quad v \quad w]^T, \omega_B = [p \quad q \quad r]^T$

The angular transformation can be represented as:

$$T = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix} \tag{A.1}$$

where $t(\theta) = tan(\theta)$, $c(\theta) = cos(\theta)$, $s(\theta) = sin(\theta)$

$$\begin{aligned} \nu &= R\nu_B \\ \omega &= T\omega_B \end{aligned} \tag{A.2}$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ u \\ v \\ w \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ p \\ q \\ r \end{bmatrix} = \begin{bmatrix} w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - v[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\ v[c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)] - w[c(\psi)s(\phi) - c(\phi)s(\psi)s(\theta)] + u[c(\theta)s(\psi)] \\ w[c(\phi)c(\theta)] - u[s(\theta)] + v[c(\theta)s(\phi)] \\ rv - qw - gs(\theta) \\ pw - ru + gc(\theta)s(\theta) \\ qu - pv + gc(\theta)s(\theta) - F/m \\ p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\ q[c(\phi)] - r[s(\phi)] \\ r\frac{c(\phi)}{c(\theta)} + q\frac{s(\phi)}{c(\theta)} \\ \frac{I_y - I_z}{I_x}qr + \frac{\tau_\phi}{I_x} \\ \frac{I_z - I_x}{I_y}pr + \frac{\tau_\theta}{I_y} \\ \frac{I_x - I_y}{I_z}pq + \frac{\tau_\psi}{I_z} \end{bmatrix} \tag{A.3}$$

$$m(\omega_B \wedge \nu_B + \dot{\nu_B}) = \mathbf{f}_B \tag{A.4}$$

m mass, $\wedge$ cross product, $\mathbf{f}_B = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \in \mathbf{R}^3$

$$I = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \wedge \mathbf{R}^{3\times 3} \tag{A.5}$$

Dynamic model in the body frame

$$I.\dot{\omega}_B + \omega_B \wedge (I.\omega_B) = \mathbf{m}_B \tag{A.6}$$

$$\begin{bmatrix} f_x \\ f_y \\ f_z \\ m_x \\ m_y \\ m_z \end{bmatrix} = \begin{bmatrix} m(\dot{u} + qw - rv) \\ m(\dot{v} - pw + ru) \\ m(\dot{w} + pv - qu) \\ \dot{p}I_x - qrI_y + qrI_z \\ \dot{q}I_y + prI_x - prI_z \\ \dot{r}I_z - pqI_x + pqI_y \end{bmatrix} \tag{A.7}$$

APPENDIX B

Camera Model and Camera Calibration

Camera calibration can be summed up as the process of estimating the parameters of a pinhole camera model [95]. The important output is obtaining the correct values for the intrinsics and extrensics properties.

Intrinsic parameters matrix can be represented as follows: K = $\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$

where f_x, f_y is considered as focal length according to x and y axis respectively. While c_x, c_y represents the camera center x and y.

Camera calibration process for the ubiquitous perspective cameras can be done using the widely available open source library openCV, which found in `https://docs.opencv.org/4.5.3/dc/dbb/tutorial_py_calibration.html`. The calibration setup is rather simple, it just requires printing a checkerboard and mounting it on a flat surface, then following some straightforward proceedures.

For omnidirectional cameras, some recent advances such as [96], provides solutions to simplify the calibration process. The authors provide a library that can be found in: `https://github.com/ethz-asl/kalibr`

APPENDIX C

Real Robots

ROTORS [67] is chosen to simulate the real flight of a UAV. The simulation can give some realistic fidelity scenes which can then be captured by simulated cameras mounted on those UAVs. Particularly a pinhole model is used. It is an open source package widely known and used within the aerial robotics researchers and practitioners for its benefits in reducing the gap between simulation and real experiments.

It is considered nowadays, that the UAV/MAV has multiple layers of control. The low level layer handles basic flight operations, which includes attitude control, height control, and velocity estimation and control. These operations are done onboard in real time manner usually programmed on a custom DSP.

Real robots used to validate the algorithms experimentally are going to be discussed in the following section.

C.0.1 Quadrotor

Bebop is a quadrotor type of a UAV it is commercial off the shelf drone. It can be shown in the figure C.1. The drone is having many on board sensors that are essential for the flight controller to be successful.

It first came to market by Parrot in 2015. The version that is used in the experimentation is Bebop 2. It is suited with a developer SDK, which is then wrapped into a ROS package [1]. The codes developed can be in Python or C++ as a publisher/subscriber mechanisms complying with the ROS mechanisms. It is equipped with the common hardware sensing IMU, camera, optical flow sensor, etc. The Bebop drone is also simulated in Gazebo as the ROTORS framework in the paper [97].

C.0.2 Omni-directional Mobile Robot

An omni-directional mobile robot is also used in the experiments

[1] https://github.com/AutonomyLab/bebop_autonomy

Figure C.1: Bebop quadrotor on the experimental ground.

The mobile robot has a fiducial marker primarily to do identification. It is mounted on a tower attached to the mobile robot to imitate the motion of an omni-directional aerial robot (i.e: quadrotor). Such fiducial markers are widely used in the augmented reality research and development community. It is used either to simplify feature extraction in the environment or as a ground truth for markerless algorithms. Fiducial markers have been widely used in robotics society. They are on Mars indeed on an extraterrestrial robots. Curiosity rover from NASA is already engraved with many markers specifically AR tag among others [2]. That is used in the camera calibration algorithms, arm relative positioning, and robot intrinsic parameters calibration, i.e: wheels are wearing, motors efficiency is deteriorating or changing, etc.

There exist today several markers in literature. One of the studies that draws comparison between different types of these markers can be found in [98]. The choice of the marker was fallen on Apriltag [3]

[2] https://mars.nasa.gov/raw_images/944849?site=msl

[3] AprilTag 2: Efficient and robust fiducial detection, Wang et al, IROS 2016, https://github.com/AprilRobotics/apriltag

Figure C.2: Summit XL Steel omni ground mobile robot with a tower augmented with fiducial marker.

APPENDIX D

Quaternion

A quaternion is basically a type of complex number that consists of four values, one of which is real, while the other three are imaginary [22–25]. The quaternion space is denoted by H. Let **q** be a quaternion expresses by:

$$\boldsymbol{q} = q_0 + \bar{q} = q_0 + \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, q_0 \in \mathbb{R}, \bar{q} \in \mathbb{R}^3$$

where $\bar{q}$ represents the complex vectorial part of **q**, and q_0 denotes the scalar part of **q**. Note that if q_0 = 0, then q = $\bar{q}$, thus q $\in \mathbb{R}^3$, this implies that R^3 is a subset of H in which q_0 = 0.

The quaternion basic operations are:

- Quaternion multiplication

$$\mathbf{q} \otimes \mathbf{r} = (q_0 r_0 - \bar{q} \cdot \bar{r}) + (r_0 \bar{q} + q_0 \bar{r} + \bar{q} \times \bar{r})$$

- Norm of a quaternion q

$$\|\boldsymbol{q}\| = \sqrt{\boldsymbol{q} \otimes \boldsymbol{q}^*} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$$

In case that $|\boldsymbol{q}\| = 1$, **q** represents an unit quaternion

- Quaternion conjugate

$$\boldsymbol{q^{-1}} = \|\mathbf{q}\|^{-1} \boldsymbol{q}^*$$

- Quaternion inverse

$$\boldsymbol{q}^{-1} = \|\boldsymbol{q}\|^{-1} \boldsymbol{q}^*$$

If **q** is unitary, then the inverse and the norm are equivalent.

- The derivative of a quaternion

$$\dot{\boldsymbol{q}} = \frac{1}{2} \boldsymbol{q} \otimes \omega$$

where ω represents the angular velocity.

Any vector in a three-dimensional space can be rotated from one reference frame to another (for example a body frame), i.e.,

$$v' = \boldsymbol{q}^* \otimes v \otimes \boldsymbol{q}$$

where $v \in \mathbb{R}^3$ and $v' \in \mathbb{R}^3$ are in the fixed and body frames respectively. Any rotation can be expressed by its axis-angle representation, which is composed by a direction vector $\bar{e} \in \mathbb{R}^3$, $\|\bar{e}\| = 1$ and its angle $\theta \in \mathbb{R}$, as depicted in Figure 1. This yields to a single vector $\bar{\theta} = \theta\bar{e}$ with $\bar{\theta} \in \mathbb{R}^3$.

The Euler-Rodrigues Formula then relates this rotation vector to a unit quaternion as $q = \cos(\theta/2) + \bar{e}\sin(\theta/2)$

A logarithmic mapping can be used to change any unit quaternion to the axis-angle representation $\bar{\theta} = 2\ln q, \quad \dot{\bar{\theta}} = \omega,$

where

$$\ln q = \begin{cases} \ln\|q\| + \frac{\bar{q}}{\|\bar{q}\|}\arccos\frac{q_0}{\|q\|}, & \|\bar{q}\| \neq 0 \\ \ln\|q\|, & \|\bar{q}\| = 0 \end{cases}$$

Note that the Euler-Rodriques implies that $\|q\| = 1$, thus the scalar part of the quaternion logarithm is zero, implying that $\ln q \in \mathbb{R}^3 \forall q | q_0 = 0$. very useful quaternion visualization tool: [1].

Homogeneous coordinates is used widely to represent 3d points on 2d image in projective geometry. They can express transformations as matrices, for further doing computation on them. The second key importance is representing points at infinity which is useful for geometric reconstruction algorithms of visual systems.

A lot of geometric operations can be done in homogeneous coordinates like translation rotation scale change. As well as several transformations such as similarity, affine, and projective.

translation in matrix form:

$$\begin{bmatrix} I & t \\ 0^T & 1 \end{bmatrix}$$

Point shifted by t:

$$\mathbf{x}' = \top\mathbf{x}$$

[1] https://eater.net/quaternions/

APPENDIX E

Publications

- Bastourous M, Guerin F, Guinand F, Leader follower formations of multi quadrotors group October 2018, Bordeaux, France https://www.labri.fr/perso/chaumett/unmanned-and-swarming-2018/ (Best Paper award)
- Mark Bastourous, François Guérin, Frédéric Guinand and Eric Lemains, Decentralized High Level Controller for Formation Flight Control of UAVs, 2020 6th International Conference on Mechatronics and Robotics Engineering, Barcelona, Spain.
- Frédéric Guinand, François Guerin, Mark Bastourous, Alignment of three robots without communication nor localization, 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
- Mark Bastourous, Jaafar Al-Tuwayyij, François Guerin, Frederic Guinand, Image Based Visual Servoing for Multi Aerial Robots Formation, 2020 28th Mediterranean Conference on Control and Automation (MED)

Bibliography

[1] Dingjiang Zhou and Mac Schwager. Virtual rigid bodies for coordinated agile maneuvering of teams of micro aerial vehicles. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1737–1742. IEEE, 2015.

[2] Tamio Arai, Enrico Pagello, and Lynne E. Parker. Editorial: Advances in multi-robot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661, 2002.

[3] Y. Uny Cao, Alex S. Fukunaga, and Andrew B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4:7–27, 1997.

[4] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. *Cooperative Grasping and Transport Using Multiple Quadrotors*, pages 545–558. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[5] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. *Multi-robot Task Allocation: A Review of the State-of-the-Art*, pages 31–51. Springer International Publishing, 2015.

[6] Yong Zeng, Rui Zhang, and Teng Joon Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communications Magazine*, 54(5):36 – 42, 2016.

[7] Gurkan Tuna, Bilel Nefzi, and Gianpaolo Conte. Unmanned aerial vehicle-aided communications system for disaster recovery. *Journal of Network and Computer Applications*, 41:27–36, 2014.

[8] Tarik Tosun, Jonathan Daudelin, Gangyuan Jing, Hadas Kress-Gazit, Mark Campbell, and Mark Yim. Perception-informed autonomous environment augmentation with modular robots. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018.

[9] Lynne E. Parker, Daniela Rus, and Gaurav S. Sukhatme. *Multiple Mobile Robot Systems*, pages 1335–1384. Springer International Publishing, Cham, 2016.

[10] Fabrizio Schiano, Antonio Franchi, Daniel Zelazo, and Paolo Robuffo Giordano. A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5099–5106. IEEE, 2016.

[11] Riccardo Spica and Paolo Robuffo Giordano. Active decentralized scale estimation for bearing-based localization. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5084–5091. IEEE, 2016.

[12] Ethan Stump, Ali Jadbabaie, and Vijay Kumar. Connectivity management in mobile robot teams. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1525–1530. IEEE, 2008.

[13] Alejandro Cornejo, Andrew J Lynch, Elizabeth Fudge, Siegfried Bilstein, Majid Khabbazian, and James McLurkin. Scale-free coordinates for multi-robot systems with bearing-only sensors. *The International Journal of Robotics Research*, 32(12):1459–1474, 2013.

[14] Tolga Eren, Walter Whiteley, A Stephen Morse, Peter N Belhumeur, and Brian DO Anderson. Sensor and network topologies of formations with direction, bearing, and angle information between agents. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 3, pages 3064–3069. IEEE, 2003.

[15] Rajesh Doriya, Siddharth Mishra, and Swati Gupta. A brief survey and analysis of multi-robot communication and coordination. In *International Conference on Computing, Communication & Automation*, pages 1014–1021. IEEE, 2015.

[16] Reza Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. Technical report, CALIFORNIA INST OF TECH PASADENA CONTROL AND DYNAMICAL SYSTEMS, 2004.

[17] Daniel P Scharf, Fred Y Hadaegh, and Scott R Ploen. A survey of spacecraft formation flying guidance and control. part ii: control. In *American Control Conference, 2004. Proceedings of the 2004*, volume 4, pages 2976–2985. IEEE, 2004.

[18] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[19] Brian DO Anderson, Changbin Yu, Julien M Hendrickx, et al. Rigid graph control architectures for autonomous formations. *IEEE Control Systems*, 28(6), 2008.

[20] Yang Quan Chen and Zhongmin Wang. Formation control: a review and a new consideration. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3181–3186. IEEE, 2005.

[21] He Bai, Murat Arcak, and John Wen. *Cooperative control design: a systematic, passivity-based approach*. Springer Science & Business Media, 2011.

[22] Celso De La Cruz and Ricardo Carelli. Dynamic model based formation control and obstacle avoidance of multi-robot systems. *Robotica*, 26(3):345–356, 2008.

[23] Noah Cowan, O Shakerina, Rene Vidal, and Shankar Sastry. Vision-based follow-the-leader. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 2, pages 1796–1801. IEEE, 2003.

[24] Omar AA Orqueda and Rafael Fierro. Robust vision-based nonlinear formation control. In *American Control Conference, 2006*, pages 6–pp. IEEE, 2006.

[25] DA Mercado, R Castro, and Rogelio Lozano. Quadrotors flight formation control using a leader-follower approach. In *Control Conference (ECC), 2013 European*, pages 3858–3863. IEEE, 2013.

[26] Qifeng Chen, Yunhe Meng, and Jianjun Xing. Shape control of spacecraft formation using a virtual spring-damper mesh. *Chinese Journal of Aeronautics*, 29(6):1730–1739, 2016.

[27] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013.

[28] Angela Schöllig, Federico Augugliaro, Sergei Lupashin, and Raffaello D'Andrea. Synchronizing the motion of a quadrocopter to music. In *2010 IEEE International Conference on Robotics and Automation*, pages 3355–3360. IEEE, 2010.

[29] Ming Cao, Changbin Yu, and Brian DO Anderson. Formation control using range-only measurements. *Automatica*, 47(4):776–781, 2011.

[30] Laura Krick, Mireille E Broucke, and Bruce A Francis. Stabilisation of infinitesimally rigid formations of multi-robot networks. *International Journal of control*, 82(3):423–439, 2009.

[31] Hector Garcia de Marina, Ming Cao, and Bayu Jayawardhana. Controlling rigid formations of mobile agents under inconsistent measurements. *IEEE Transactions on Robotics*, 31(1):31–39, 2015.

[32] Daniel Zelazo, Antonio Franchi, Heinrich H Bülthoff, and Paolo Robuffo Giordano. Decentralized rigidity maintenance control with range measurements for multi-robot systems. *The International Journal of Robotics Research*, 34(1):105–128, 2015.

[33] Iman Shames, Adrian N Bishop, and Brian DO Anderson. Analysis of noisy bearing-only network localization. *IEEE Transactions on Automatic Control*, 58(1):247–252, 2013.

[34] Tolga Eren. Using angle of arrival (bearing) information for localization in robot networks. *Turkish Journal of Electrical Engineering & Computer Sciences*, 15(2):169–186, 2007.

[35] Adrian N Bishop, Iman Shames, and Brian DO Anderson. Stabilization of rigid formations with direction-only constraints. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 746–752. IEEE, 2011.

[36] Antonio Franchi and Paolo Robuffo Giordano. Decentralized control of parallel rigid formations with direction constraints and bearing measurements. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 5310–5317. IEEE, 2012.

[37] Antonio Franchi, Carlo Masone, Volker Grabe, Markus Ryll, Heinrich H Bülthoff, and Paolo Robuffo Giordano. Modeling and control of uav bearing formations with bilateral high-level steering. *The International Journal of Robotics Research*, 31(12):1504–1525, 2012.

[38] Audrey Lee-St John and Ileana Streinu. Angular rigidity in 3d: combinatorial characterizations and algorithms. In *CCCG*, pages 67–70. Citeseer, 2009.

[39] Shiyu Zhao, Feng Lin, Kemao Peng, Ben M Chen, and Tong H Lee. Distributed control of angle-constrained cyclic formations using bearing-only measurements. *Systems & Control Letters*, 63:12–24, 2014.

[40] Nima Moshtagh, Nathan Michael, Ali Jadbabaie, and Kostas Daniilidis. Bearing-only control laws for balanced circular formations of ground robots. *Proceedings of Robotics: Science and Systems IV*, 2008.

[41] Meysam Basiri, Adrian N Bishop, and Patric Jensfelt. Distributed control of triangular formations with angle-only constraints. *Systems & Control Letters*, 59(2):147–154, 2010.

[42] Adrian N Bishop. A very relaxed control law for bearing-only triangular formation control. *IFAC Proceedings Volumes*, 44(1):5991–5998, 2011.

[43] Adrian N Bishop, Mohammad Deghat, Brian Anderson, and Yiguang Hong. Distributed formation control with relaxed motion requirements. *International Journal of Robust and Nonlinear Control*, 25(17):3210–3230, 2015.

[44] Daniel Zelazo, Paolo Robuffo Giordano, and Antonio Franchi. Bearing-only formation control using an se (2) rigidity theory. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 6121–6126. IEEE, 2015.

[45] Fabrizio Schiano and Paolo Robuffo Giordano. Bearing rigidity maintenance for formations of quadrotor UAVs. In *ICRA 2017 - IEEE International Conference on Robotics and Automation*, pages 1467–1474, Singapore, Singapore, May 2017.

[46] Yongcan Cao and Wei Ren. Distributed coordinated tracking via a variable structure approach-part i: Consensus tracking. In *American Control Conference (ACC), 2010*, pages 4744–4749. IEEE, 2010.

[47] Meng Ji and Magnus Egerstedt. Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703, 2007.

[48] Giuseppe Notarstefano, Ketan Savla, Francesco Bullo, and Ali Jadbabaie. Maintaining limited-range connectivity among second-order agents. In *American Control Conference, 2006*, pages 6–pp. IEEE, 2006.

[49] Peng Yang, Randy A Freeman, Geoffrey J Gordon, Kevin M Lynch, Siddhartha S Srinivasa, and Rahul Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46(2):390–396, 2010.

[50] Adrian N Bishop. Distributed bearing-only formation control with four agents and a weak control law. In *Control and Automation (ICCA), 2011 9th IEEE International Conference on*, pages 30–35. IEEE, 2011.

[51] Herbert G Tanner, George J Pappas, and Vijay Kumar. Leader-to-formation stability. *IEEE Transactions on robotics and automation*, 20(3):443–455, 2004.

[52] Matthew O Jackson. A survey of network formation models: stability and efficiency. *Group formation in economics: Networks, clubs, and coalitions*, 664:11–49, 2005.

[53] Jorge Cortés. Global and robust formation-shape stabilization of relative sensing networks. *Automatica*, 45(12):2754–2762, 2009.

[54] Meng Ji, Abubakr Muhammad, and Magnus Egerstedt. Leader-based multi-agent coordination: Controllability and optimal control. In *American Control Conference, 2006*, pages 6–pp. IEEE, 2006.

[55] Mac Schwager, Nathan Michael, Vijay Kumar, and Daniela Rus. Time scales and stability in networked multi-robot systems. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3855–3862. IEEE, 2011.

[56] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[57] Wei Ren, Randal W Beard, and Ella M Atkins. A survey of consensus problems in multi-agent coordination. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1859–1864. IEEE, 2005.

[58] Peng Lin, Yingmin Jia, and Lin Li. Distributed robust h consensus control in directed networks of agents with time-delay. *Systems & control letters*, 57(8):643–653, 2008.

[59] Emmanuel Nuno, Tonatiuh Henández, Mohamed Maghenem, Antonio Loría, and Elena Panteley. Leaderless consensus-based formation control of multiple nonholonomic mobile robots with interconnecting delays. In *2019 American Control Conference (ACC)*, pages 4659–4664. IEEE, 2019.

[60] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. Multi-robot task allocation: A review of the state-of-the-art. *Cooperative Robots and Sensor Networks 2015*, pages 31–51, 2015.

[61] Xiaohan Chen and Yingmin Jia. Adaptive leader-follower formation control of non-holonomic mobile robots using active vision. *IET Control Theory & Applications*, 9(8):1302–1311, May 2015.

[62] Fabio Morbidi, Gian Luca Mariottini, and Domenico Prattichizzo. Observer design via immersion and invariance for vision-based leader–follower formation control. *Automatica*, 46(1):148–154, January 2010.

[63] Dejun Guo, Hesheng Wang, Weidong Chen, Ming Liu, Zeyang Xia, and Kam K. Leang. A unified leader-follower scheme for mobile robots with uncalibrated on-board camera. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2017.

[64] Xiaomei Liu, Shuzhi Sam Ge, and Cher-Hiang Goh. Vision-based leader–follower formation control of multiagents with visibility constraints. *IEEE Transactions on Control Systems Technology*, 27(3):1326–1333, May 2019.

[65] Joseph J LaViola. Double exponential smoothing: an alternative to kalman filter-based predictive tracking. In *Proceedings of the workshop on Virtual environments 2003*, pages 199–206. ACM, 2003.

[66] ARdrone2.0 Simulink toolbox. `https://www.mathworks.com/matlabcentral/fileexchange/43719-ar-drone-simulink-development-kit-v1-1`. Accessed: 2018-09-30.

[67] Fadri Furrer, Michael Burri, Markus Achtelik, and Roland Siegwart. *Robot Operating System (ROS): The Complete Reference (Volume 1)*, chapter RotorS—A Modular Gazebo MAV Simulator Framework, pages 595–625. Springer International Publishing, Cham, 2016.

[68] Lorenz Meier, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision. *Autonomous Robots*, 33(1-2):21–39, 2012.

[69] Matthew L Robinson, Eric T Baumgartner, Kevin M Nickels, and Todd E Litwin. Hybrid image plane/stereo (hips) manipulation for robotic space applications. *Autonomous Robots*, 23(2):83–96, 2007.

[70] Alexandre Santos Brandão and Mário Sarcinelli-Filho. On the guidance of multiple uav using a centralized formation control scheme and delaunay triangulation. *Journal of Intelligent & Robotic Systems*, 84(1-4):397–413, 2016.

[71] Arthur C Sanderson and Lee E Weiss. Adaptive visual servo control of robots. In *Robot vision*, pages 107–116. Springer, 1983.

[72] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[73] F. Chaumette. Visual servoing. In K. Ikeuchi, editor, *Computer Vision: A Reference Guide*, pages 869–874. Springer, 2014.

[74] Francois Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control*, pages 66–78. Springer, 1998.

[75] L. Teixeira, F. Maffra, M. Moos, and M. Chli. Vi-rpe: Visual-inertial relative pose estimation for aerial vehicles. *IEEE Robotics and Automation Letters*, 3(4):2770–2777, October 2018.

[76] Mark Bastourous, François Guérin, Frédéric Guinand, and Eric Lemains. Decentralized high level controller for formation flight control of uavs. To appear in the 2020 6th International Conference on Mechatronics and Robotics Engineering. Barcelona, Spain. February 12-15, 2020.

[77] Omar Tahri and Francois Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics*, 21(6):1116–1127, 2005.

[78] Peter A Karasev, Miguel Moises Serrano, Patricio A Vela, and Allen Tannenbaum. Depth invariant visual servoing. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4992–4998. IEEE, 2011.

[79] Olivier Kermorgant and François Chaumette. Dealing with constraints in sensor-based robot control. *IEEE Transactions on Robotics*, 30(1):244–257, 2013.

[80] Kazuo Sugihara and Ichiro Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *Journal of Robotic Systems*, 13(3):127–139, 1996.

[81] Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, January 1999.

[82] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita. Distributed memoryless point convergence algorithm for mobile robots with limited visibility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, October 1999.

[83] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theoretical Computer Science*, 407(1-3):412–447, November 2008.

[84] Shantanu Das, Paola Flocchini, Nicola Santoro, and Masafumi Yamashita. Forming sequences of geometric patterns with oblivious mobile robots. *Distributed Computing*, 28(2):131–145, April 2015.

[85] T. Balch and R. C. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.

[86] Antonio Franchi, Giuseppe Oriolo, and Paolo Stegagno. Mutual localization in a multi-robot system with anonymous relative position measures. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, October 2009.

[87] B. Fidan, V. Gazi, S. Zhai, N. Cen, and E. Karataş. Single-view distance-estimation-based formation control of robotic swarms. *IEEE Transactions on Industrial Electronics*, 60(12):5781–5791, December 2013.

[88] Gazebo: http://gazebosim.org, last visit: 04/2019.

[89] Ros: http://www.ros.org, last visit: 04/2019.

[90] Antoine Dutot, Frédéric Guinand, Damien Olivier, and Yoann Pigné. Graphstream: A tool for bridging the gap between complex systems and dynamic graphs. In Aziz Alaoui and Cyrille Bertelle, editors, *Proceedings of Emergent Properties in Natural and Artificial Complex Systems. Satellite Conference within the 4th European Conference on Complex Systems (ECCS'2007), October 4-5. Dresden, Germany.*, pages 63–72, 2007.

[91] Viktor Walter, Nicolas Staub, Antonio Franchi, and Martin Saska. Uvdar system for visual relative localization with application to leader–follower formations of multirotor uavs. *IEEE Robotics and Automation Letters*, 4(3):2637–2644, 2019.

[92] Samir Bouabdallah and Roland Siegwart. Full control of a quadrotor. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–158. Ieee, 2007.

[93] Samir Bouabdallah. Design and control of quadrotors with application to autonomous flying. Technical report, Epfl, 2007.

[94] Gabriele Perozzi. A toolbox for quadrotors: from aerodynamic science to control theory. 2018.

[95] Alex M Andrew. Multiple view geometry in computer vision. *Kybernetes*, 2001.

[96] Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016.

[97] Giuseppe Silano, Pasquale Oppido, and Luigi Iannelli. Software-in-the-loop simulation for improving flight control system design: a quadrotor case study. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 466–471. IEEE, 2019.

[98] Michail Kalaitzakis, Sabrina Carroll, Anand Ambrosi, Camden Whitehead, and Nikolaos Vitzilaios. Experimental comparison of fiducial markers for pose estimation. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 781–789. IEEE, 2020.