# Distributed State Estimation and Control of Autonomous Quadrotor Formations Using Exclusively Onboard Resources

THÈSE N$^{O}$ 9224 (2018)

PRÉSENTÉE LE 16 NOVEMBRE 2018

À L'ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
À LA FACULTÉ DE L'ENVIRONNEMENT NATUREL, ARCHITECTURAL ET CONSTRUIT
LABORATOIRE DE SYSTÈMES ET ALGORITHMES INTELLIGENTS DISTRIBUÉS

ET

À L'INSTITUTO SUPERIOR TÉCNICO (IST) DA UNIVERSIDADE DE LISBOA

PROGRAMME DOCTORAL EN ROBOTIQUE, CONTRÔLE ET SYSTÈMES INTELLIGENTS
ET
DOUTORAMENTO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES (PhD)

PAR

Duarte DA CRUZ BAPTISTA DIAS

acceptée sur proposition du jury:

Prof. J. Santos-Victor, président du jury
Prof. A. Martinoli, Prof. P. M. U. de Almeida Lima, directeurs de thèse
Dr A. Franchi, rapporteur
Prof. J. Tasso de Figueiredo Borges de Sousa, rapporteur
Prof. R. Maria Mendes de Almeida Correia da Cunha, rapporteuse

Suisse
2019

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

# Distributed State Estimation and Control of Autonomous Quadrotor Formations Using Exclusively Onboard Resources

## Duarte da Cruz Baptista Dias

**Supervisors:** **Doctor Pedro Manuel Urbano de Almeida Lima**
**Doctor Alcherio Martinoli**

**Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering**

**Jury final classification: Pass with Distinction**

**Jury**

**Chairperson: Doctor José Alberto Rosado dos Santos Victor**, Instituto Superior Técnico, Universidade de Lisboa

**Members of the committee:**

**Doctor Alcherio Martinoli**, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédéral de Lausanne
**Doctor João Tasso de Figueiredo Borges de Sousa**, Faculdade de Engenharia, Universidade do Porto
**Doctor Rita Maria Mendes de Almeida Correia da Cunha**, Instituto Superior Técnico, Universidade de Lisboa
**Doctor Antonio Franchi**, Centre National de la Recherche Scientifique, Laboratoire d'Analyse et d'Architecture des Systèmes

**Funding institutions**

Fundação para a Ciência e Tecnologia through doctoral grant SFRH / BD / 51928 / 2012 and the ISR Strategic Funding
EPFL through discretionary funding for DISAL salary envelope

**2018**

# Acknowledgments

Firstly, I would like to thank my two supervisors, Professor Alcherio Martinoli and Professor Pedro Lima for their support and guidance during my PhD studies. I thank them for providing me with a great research environment. I am very grateful to Alcherio for helping me to develop my interest and understanding in the domains of multi agent systems and of aerial robotics, and for providing me with all the necessary support while I was developing the devices used in this work. I am very grateful to Pedro for his valuable and highly insightful feedbacks and comments that allowed me to keep improving the way that I communicate and present my work, and for his encouragements that gave me confidence in the times where I needed it the most.

I would also like to thank Professor Rodrigo Ventura for all the fruitful discussions and for all his insights and eagerness to help, that kept me motivated and helped me to think out of the box during the course of my PhD.

I am grateful to both EPFL and IST for providing a great studying and working environment, to the Foundation for Science and Technology (FCT), the Distributed Intelligent Systems and Algorithms Laboratory (DISAL), and the Institute for Systems and Robotics (ISR) at IST for funding this research. I would also like to thank my colleague Bruno Gomes from the DSOR laboratory and the managing staff of the AEIST sports pavilion at IST for providing the space for conducting flying experiments at IST. I am also grateful to Professor Miguel da Silva and my colleague Sérgio Gonçalves for helping me acquiring data to evaluate the work in this thesis, using their space at LBL in IST.

Next, I would like to thank all my colleagues from DISAL and ISR. Special thanks goes to my colleagues Henrique Silva, Filipe Rosa and Diogo Serra, that helped me conducting the experiments of this work at IST, and to my colleagues Steven Roelofsen and Chris Evans that helped me conducting the experiments of this work at EPFL. I would also like to thank the engineer Emannuel Droz from DISAL for his feedbacks, comments, and support during the development of the hardware used in this thesis. I would also like to specially thank my colleague Alicja Wasik for her constant presence and friendship during all the transitions between the IST and EPFL institutions, and to my colleague Bahar Haghighat for all our exciting discussions that helped me relax and keep a positive perspective for the future.

This work would not have been possible without the support of good friends. I would like to specially thank João Carvalho for all the discussions and encouragements, and for providing me with a familiar environment throughout my PhD studies. I would also like to thank my swimming team back in Portugal, “GESLoures/World’s Best Team”, that greatly contributed to my relaxation and motivation needed to finalize this work.

Last but not least, I thank my entire family for all their support. I am grateful to my brother and my mother for standing by me for the past 5 years. This work would not have been possible without their support. I thank them for having to put up with me in all the stressful days, and for all the sacrifices they made for me during this time.

# Abstract

The navigation of unmanned aerial vehicles operating in environments without global positioning systems, including global navigation satellite systems and motion capture systems, is a recent research topic, without much work reported in the literature. In indoor applications, particularly, small-scale vehicles are subjected to severe power and weight constraints, limiting their overall navigation capabilities. In such scenarios, multi-vehicle systems can be used in order to mitigate the impact of limited capabilities at the individual vehicle level. If, additionally, a group of vehicles has to maintain a specific spatial topology, well-established formation control algorithms can be used as long as information about mutual inter-vehicle positioning is available.

This information can be directly acquired using relative positioning systems on each vehicle. This solution enables the multi-vehicle system to reduce its dependency on absolute localization systems and explicit inter-vehicle communications. Additionally, multi-vehicle formation control can be achieved in either fully distributed or decentralized fashion, reducing the need for external and/or centralized units supervising the system. However, the aforementioned energy and weight constraints of small-scale vehicles only allow limited sensing payloads, reducing the potential sophistication of onboard relative positioning systems. For the same reason, steering a multi-vehicle system through given way-points using only onboard resources becomes a challenging task.

This thesis introduces two novel relative positioning systems for multi-vehicle formations, focusing on maximizing the number of detected team members while remaining accurate and light enough to allow their deployment on small-scale vehicles: i) a camera-based system that enables a scalable deployment on multiple vehicles; ii) an infrared-based system that provides several hardware and software enhancements with respect to systems reported in the literature using the same technology. The camera-based sensor model can be leveraged as a tool for optimizing the design parameters to meet specific accuracy requirements and allows the system to achieve highly accurate relative localization measurements using low-resolution cameras. The infrared-based system uses miniature omni-directional infrared beacons deployable in small sets on each vehicle which, together with dedicated estimation and calibration algorithms, ensures a adaptability to any 3D geometry of the carrying vehicle. Such innovative design principles result in a system which enables a direct measurement of the relative attitude, and is more flexible, lighter, and less power-hungry than state-of-the-art devices, while providing similar accuracy.

Novel formation control methods that tackle limitations arising from the exclusive use of relative positioning systems are an extra contribution of the thesis. A graph-based formation control algorithm has been extended so that sensing constraints could be taken into account when a vehicle has to observe multiple neighbors. This extension consists of enabling each vehicle to control the occupied area of the limited field of view of its sensor, while it moves to the right place in the formation. This in turn provides additional flexibility for the formation topology despite the inter-vehicle sensing limitations. A formation steering algorithm capable of providing a consistent and simultaneous motion direction to all team members has also been developed. This was achieved without requiring artificial landmarks in the environment and/or additional communication overhead between the vehicles. The proposed steering algorithm increases the reactiveness of the formation control when compared to canonical methods relying on leader vehicles.

Simulation experiments show the functionality of the developed formation control and formation steering algorithms in a four-vehicle formation, with sensing noise matching realistic conditions. Real experiments with up to three vehicles (two flying and one static) equipped with the camera-based system show that the formation control algorithm allows each vehicle to keep multiple neighbors inside the field of view of the sensor, in spite of actuation inaccuracies. Moreover, is is shown that the camera-based system can be used as the sole provider of sensory feedback for the control algorithms. Real experiments with two vehicles equipped with the infrared-based system show that the large field of view of this system enables it to track neighboring vehicles even when they move along a large portion of the sensor area of vision. Additionally, the formation steering algorithm is shown to produce a motion lag between vehicles smaller than that produced by the leader-based methods. Two different environments and two different flying platforms were used in the real experiments, showing that the developed sensors and algorithms can be deployed on different vehicles and conditions.

**Keywords:** unmanned aerial vehicles, multi-vehicle systems, formation control and formation steering, relative localization systems, camera-based system, infrared-based system, sensing constraints.

# Resumo

Veículos aéreos não tripulados a operar em espaços desprovidos de sistemas de posicionamento global, incluindo sistemas de navegação por satélite ou sistemas de captura de movimentos, é um tópico de estudo recente, com pouco trabalho reportado na literatura. Particularmente, em espaços fechados, veículos de pequena dimensão estão sujeitos a severas restrições de peso e energia, limitando as suas capacidades de navegação. Nessas situações, sistemas de múltiplos veículos podem ser usados para mitigar o impacto das capacidades reduzidas de cada veículo. Se, adicionalmente, um grupo de veículos deve manter uma formação geométrica específica, algoritmos de controlo de formação podem ser usados desde que a informação sobre o posicionamento relativo dos veículos esteja disponível.

Esta informação pode ser medida directamente através de sistemas de posicionamento relativo a bordo de cada veículo. Desta forma, o sistema de múltiplos veículos não depende das condições do espaço de operação nem dos sistemas de comunicação utilizados. Adicionalmente, com estes sensores torna-se possível a coordenação entre veículos de uma forma distribuída, reduzindo as necessidades de supervisão externa e/ou centralizada ao sistema. No entanto, as elevadas restrições energéticas e de peso dos veículos de pequena dimensão só permitem instalação a bordo de sistemas simples, limitando a potencial complexidade destes sistemas de posicionamento relativo. Isto leva a que seja difícil deslocar o sistema de múltiplos veículos entre vários objectivos utilizando apenas os recursos disponíveis a bordo.

Nesta tese são desenvolvidos novos sistemas de posicionamento relativo, tendo como objectivo maximizar o número de veículos detectados e manter os sistemas precisos e leves o suficiente para poderem ser introduzidos nos veículos: i) um sistema baseado em visão que pode ser montado a bordo de múltiplos veículos de forma escalável; ii) um sistema baseado em tecnologia infravermelha que foi desenvolvido com vários melhoramentos em termos de software e de hardware, em relação a outros sistemas da literatura que utilizam tecnologia semelhante. O modelo de sensor proposto para o sistema baseado em visão pode ser utilizado como ferramenta de optimização dos parâmetros do sistema de forma a obter um desempenho específico, e permite que o sistema atinja medidas de localização de alta precisão utilizando câmaras de baixa resolução. O sistema de tecnologia infravermelha utiliza pequenos marcadores activos e omni-direccionais, sendo que vários deles podem ser introduzidos em cada veículo em simultâneo. Este sistema, aliado a algoritmos de estimação e calibração propostos faz com que o sistema seja adaptável à geometria tridimensional do veículo alvo. Adicionalmente, as inovações propostas ao sistema permitem a aquisição medições de atitude relativa, bem como resultam num sistema mais flexível, leve, e com menos necessidades energéticas que outros sistemas na literatura.

Algoritmos de controlo de formação com o objectivo de resolver as limitações inerentes da utilização exclusiva de sistemas de posicionamento relativo consistem numa contribuição adicional desta tese. Um algoritmo de formação baseado em grafos foi estendido para que as restrições dos sensores possam ser consideradas quando o veículo tem de observar múltiplos vizinhos. Esta extensão consiste em permitir que cada veículo controle a área de visão ocupada do sensor enquanto se desloca para a sua posição na formação. Esta extensão leva a que mais formações possam ser consideradas apesar das limitações dos sensores. Um algoritmo de navegação da formação que permite o envio de uma direcção de movimento consistente a todos os veículos do grupo em simultâneo foi também desenvolvido. Os veículos são capazes de interpretar esta direcção sem comunicação adicional entre veículos ou marcadores presentes no ambiente. Este algoritmo de navegação permite o aumento da reactividade da formação quando comparado com outros métodos de navegação baseados num líder de grupo.

Experiências em simulação mostram a funcionalidade dos algoritmos de controlo e navegação de formações com um sistema de quatro veículos, em condições de ruído de sensor realista. Experiências utilizando até três veículos reais (dois a voar e um estático) equipados com o sistema baseado em visão mostram que o algoritmo de controlo de formações permite aos veículos conterem vários vizinhos na área de visão do sensor, independentemente dos erros de actuação dos veículos. É também mostrado que os algoritmos de controlo funcionam correctamente mesmo que dependam exclusivamente das medidas adquiridas pelo sistema baseado em visão. Experiências com dois veículos equipados com o sistema de tecnologia infravermelha mostram que a grande área de visão deste sistema permite a este localizar vários veículos vizinhos mesmo quando eles se movem sobre uma grande área de visão do sensor. Adicionalmente, mostra-se que o algoritmo de navegação de formações produz um atraso de movimento entre veículos menor que métodos baseados num líder de grupo. Dois ambientes e duas plataformas de voo diferentes foram utilizados para as experiências reais, mostrando que os sensores e algoritmos desenvolvidos podem ser equipados em diferentes veículos e condições.

**Palavras-chave:** Veículos aérios não tripulados, sistemas de multiplos veículos, controlo de formações, navegação de formações, sistemas de posicionamento relativo, sistemas baseados em visão, sistemas baseados em tecnologia infravermelha, restrições dos sensores.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| AOA | Angle of Arrival |
| API | Application Program Interface |
| FM | Frequency Modulation |
| FOV | Field of View |
| GNSS | Global Navigation Satellite Systems |
| ID | Identifier |
| IMU | Inertial Measurement Unit |
| IR | Infrared |
| LED | Light Emitting Diode |
| LOS | Line-of-Sight |
| MCS | Motion Capture System |
| MPU | Micro Processing Unit |
| NLOS | Non-Line-of-Sight |
| OF | Optic Flow |
| RF | Radio Frequency |
| ROI | Region of Interest |
| ROS | Robotic Operating System |
| RSS | Real Time System |

| | |
|---|---|
| RSS | Received Signal Strength |
| SLAM | Simultaneous Localization and Mapping |
| TDMA | Time Division Multiple Access |
| TDOA | Time Difference of Arrival |
| TOF | Time of Flight |
| UART | Universal Asynchronous Receiver-Transmitter |
| UAV | Unmanned Aerial Vehicle |
| UDP | User Datagram Protocol |
| USB | Universal Serial Bus |
| UWB | Ultrawide-band |

# Chapter 1

# Introduction

The use of Unmanned Aerial Vehicles (UAVs) allows for a higher penetration capability through a multitude of environments, due to their intrinsic maneuverability in 3D space and therefore higher potential in negotiating all sort of obstacles on their path. UAV systems are interesting for applications where usually the task at hand involves places considered too dangerous for the human being. Examples of such applications are construction, search and rescue, environment monitoring or aerial surveillance systems. These vehicles can fly at a certain height, encountering less obstacles on the way and benefiting from a birds eye view of the scenery. Therefore, they can naturally provide assistance to ground vehicles. For example, they can carry necessary tools (or the ground vehicles themselves) between places that are not connected from a ground navigation perspective, or provide extended coverage of the environment in order to allow an improved the task planning and therefore an improved team performance.

Usually, UAVs are deployed in outdoor environments characterized by vast open areas. However, their usage in cluttered environments (including indoor spaces) has been substantially increasing given that their design has been simplified and their control techniques made more robust, ultimately enabling smaller sized vehicles. Among the available flying platforms, quadrotors are typically chosen due to their high maneuverability in such confined spaces resulting from their ability to hover.

Coordinating multiple UAVs in formation is often considered in order to extend the overall capabilities of the multi-UAV system. For example, in the work presented in [112] (depicted in Fig. 1.1), a team of UAVs is used to increase the environment sensing and communication coverage of the entire system. As the vehicle size becomes smaller, so do their capabilities. This is especially the case for UAVs since their flying requirements impose on the vehicle design harsh power and weight constraints. In this case, the use of multi-UAV systems becomes increasingly important. Fig. 1.1b and c show two advantages of using such multi-UAV systems in terms of manipulation and sensing capabilities. Formation control is a widely studied topic, both in 2D and 3D configurations, with an extensive literature. These algorithms are based on each UAV maintaining certain relative ranges and bearings with respect to the other team members. This requires that each UAV knows the position of those team members relative to itself.

In most common approaches, the UAVs obtain the required relative localization information by shar-

[1] http://smavnet.epfl.ch/

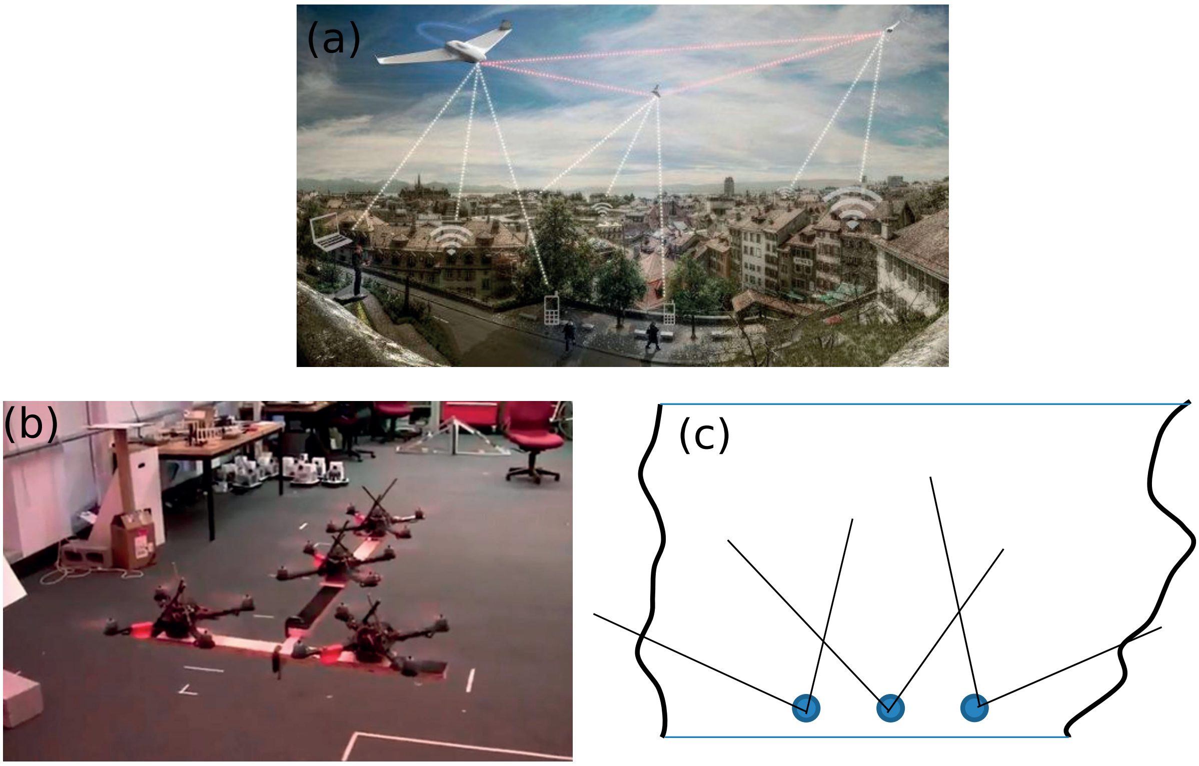


Figure 1.1: Advantages of multi-UAV systems in various scenarios. (a) Virtual mockup of SMAVNET II project, where a multi-UAV system allows communication coverage on wide outdoor scenario [112][1]. (b) UAVs in [70] organize to pick and transport heavy objects with a specific shape. (c) Virtual sketch of multiple UAVs directing their onboard cameras limited field of view so that their combined images provide full 3D environment coverage.

ing their environment positions via a communication channel. In outdoor environments, the UAVs compute their positions using Global Navigation Satellite Systems (GNSSs), such as the work in [113], depicted in Fig. 1.2a. However, for more confined spaces GNSS-based technology does not provide enough accuracy or might not be available, as for instance in indoor spaces. Typical indoor localization solutions leverages offboard Motion Capture Systems (MCSs) that externally compute the position of each UAV in the environment and communicate all the required information to their transceivers. For example, the MCS system of the GRASP laboratory in [110], shown in Fig. 1.2b, is composed of a constellation of cameras placed at different positions and orientations connected to a central processing unit that computes the position of all the UAVs in the environment. Localization information can be directly communicated to the UAVs to be used by the onboard control algorithms. In simpler cases, both localization and control algorithms are external to the UAVs, and only the final actuation commands are communicated. MCS based on different physical channels are possible: vision (the ETH flying arena [3, 89]), impulse-radio ultrawide band (the EPFL arena [84] or the system in [48]), or ultrasound [40].

However, these MCSs are costly and require a complex positioning and communication systems to allow accurate localization and control of each UAV. Additionally, by having to keep track of the sensing and control information of all UAVs, the centralized processing unit of the MCS does not scale with the number of UAVs. Moreover, MCSs can be impractical in many scenarios due to environment characteristics or lack of time or resources to set up the required infrastructure. In the past years, a large effort has been carried out to extend onboard sensing and computational capabilities for UAVs. In this way, the autonomy of these vehicles during task execution is increased, reducing the need for supervision from external planning or localization systems.

In such cases, the UAV position in the environment is usually acquired using onboard sensors capable of detecting environment landmarks, as for example in Fig. 1.2c. This landmark information can be used in model matching and Simultaneous Localization and Mapping (SLAM) algorithms to obtain the UAV position in the environment, as in [97, 100, 101]. This localization information can again be shared between UAVs using their onboard transceivers. However, acquiring an environment position and sharing it with other team members using onboard sensing and communication capabilities to obtain the necessary relative localization information can become a computationally expensive operation. Firstly, unstructured environments might limit the extraction of landmark information necessary for the localization task. Secondly, the communication between UAVs can be subjected to packet loss or latency in communication links. Finally, when using small-scale UAVs, the necessary computational resources necessary for a proper localization and communication performance might not be available. The previous points can ultimately lead to low localization update rates, which might not be feasible when high control rates are necessary to stabilize the dynamics of UAVs. This is particularly important for short range inter-UAV interactions, requiring faster reaction times.

Another interesting approach is to endow each UAV with positioning systems that directly extract the relative localization information of nearby team members by detecting features present in those UAVs' bodies. In this way, the multi-UAV system does not depend on the environment structure or the complexity of the communication system on each robot in order to function correctly. Additionally, the coordination between UAVs can be transformed into a local task for each UAV, reducing the supervision requirements from external systems. Although several technologies are used to design such onboard systems, camera-based and InfraRed (IR)-based technologies, depicted Fig. 1.3, stand out in indoor and in confined spaces. Both technologies are capable of producing highly accurate relative localization information of nearby team members which is required for coordination of multiple flying UAVs. Additionally, camera-based positioning systems have become highly mature and are simple to deploy in the vehicles, while IR-based positioning systems are capable of producing much higher sensor bandwidths with less computational power, making them an interesting solution for high speed maneuvers.

However, these positioning systems have limited capabilities. The computational complexity, in terms of image processing associated to camera-based systems, often indirectly enforces a limited Field Of View (FOV) on resource-constrained robots, such as small-scale UAVs. Lower sensor resolution and measurement frequencies could decrease the computational complexity of these systems, but that is not recommended for flying maneuvers, since high control rates are necessary to stabilize the highly dynamical system formed by these UAVs. Therefore, UAVs with this type of relative positioning systems can

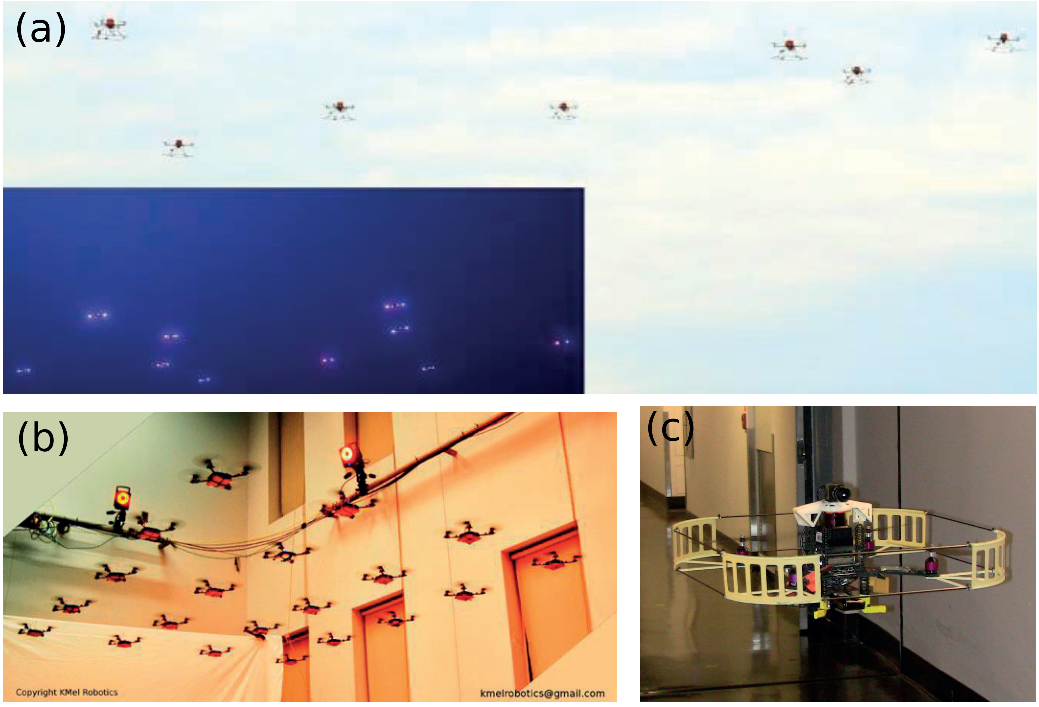


Figure 1.2: Different methodologies for acquiring relative localization information using environment localization information and communication. (a) A swarm of UAVs maintains its relative range by sharing GNSS-based measurements [113]. (b) The MCS of the GRASP laboratory is capable of externally computing the position of all UAVs in the environment and communicate to them the required control to achieve the desired formation geometry [110]. (c) The UAV in [100] computes its position in an indoor environment by comparing its onboard sensor readings to a map. Multiple UAVs with these capabilities can share the acquired localization information to allow the coordination of a multi-UAV system.

typically sense a reduced number of team members in the camera sensor, as depicted in Fig. 1.3a. This sensing limitation reduces the number of possible inter-UAV interactions, which can greatly limit the amount of geometries that can be considered for the multi-UAV system, and therefore the number of applications using these relative sensing technologies. The missing relative localization information can be provided through communication between the team members. However, as previously discussed, relying on communication systems can make the approach sensitive to packet loss or latency in communication links, given the high control rates necessary to stabilize the highly dynamical system formed by these UAVs.

Infrared-based systems, such as the ones depicted in Fig. 1.3b, provide greater FOV coverage with minimal computational requirements and fairly high sensor accuracy and bandwidths. In these systems a set of IR emitters and IR receivers are placed on each UAV. However, since IR emitters and receivers have to be in direct line of sight to allow the detection to happen, IR emission generated from a UAV's body in every direction is difficult to achieve. Although this design is simple for 2D localization, its complexity is greatly increased for the 3D case, requiring more than 100 LED emitters [92]. Additionally, the existent

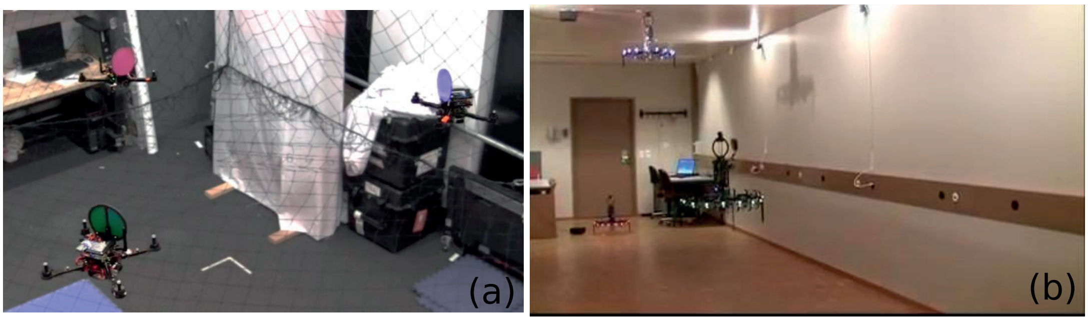


Figure 1.3: Camera-based and infrared-based relative positioning system examples. (a) In the camera-based system proposed in [109], each UAV detects the relative position of other team members related to the size and the position of their bodies observed by their onboard cameras. (b) In the infrared-based system proposed in [92], the light intensity generated by onboard IR emitters on each UAV is measured at several receivers placed on the other UAVs. These measurements are fused in order to obtain the team member's relative position.

sensor designs with this technology require the IR receivers to be placed at specific positions and orientations in the UAV, which can be incompatible or cumbersome to achieve given the complex 3D geometries of vehicles. The previous problems require the deployment of additional mechanical support structures on the UAVs, substantially increasing the vehicle weight and the sensor deployment complexity. This extra weight and complexity can become infeasible for the targeted small-scale vehicles.

Moreover, using relative positioning systems makes the UAVs to only have access to the relative localization information of its nearby team members. Without additional consensus algorithms [73] and sharing additional information between the UAVs [37] (that would increase the system complexity and lead to scalability problems), planning the motion of the multi-UAV system in the environment becomes a complicated task. Most common approaches move the system by making use of leader UAVs that decide where to move while the other UAVs follow those leaders. Other approaches motivate the direction of each UAV directly from its onboard sensors. For example, the chosen direction of motion for each UAV could be associated to the direction of the light sensed by a light sensor on board the UAV. Although these approaches are more adaptable to dynamic and unstructured environments, they lead to less reactive systems (e.g., the leader has to wait for the followers) or difficulties in specifying the desired system behaviors (e.g., in presence of multiple light sources the group might have problems in moving together, as different vehicles might want to chase different light sources in a non-optimal way).

These previous limitations of the relative positioning systems generate a challenge when considering the formation control of small-scale UAVs using exclusively their onboard resources. The goal of this thesis is to contribute in the field of aerial robotics by proposing solutions to these challenges. Specifically, this work targets the UAV sensory limitations in acquiring the position of team members in a three dimensional environment while having inherent power and weight design constraints. For this purpose, novel sensing and control solutions that enable each UAV to obtain the relative position of other team members are proposed. The solutions developed in this work are focused on reducing the limitations that these sensors bring to the tasks that involve the coordination of multiple UAVs. In the remainder of this chapter, the main contributions of this work are described along with a general overview of the thesis defining the adopted approach.

## 1.1 Main contributions

The main contribution of this thesis is the development of a set of sensor and control solutions that tackles the challenge of formation control of small-scale UAVs using exclusively onboard resources, operating in indoor environments or in environments where GNSS-based technology is not available. Specifically, this work targets vehicle sensory limitations in acquiring the position of their team members in a three dimensional environment while having inherent computational and weight design constraints. For this purpose, novel relative positioning systems that allow each UAV to obtain the relative position of other team members were proposed. These systems focused on maximizing the number of detected team members while remaining accurate and light enough to allow its deployment and its functionality for the coordination of multiple UAVs.

More specifically, this thesis contributes in this area by:

- Developing a camera-based system with the following features:

    1. The design allows its deployment in a way that is scalable with the number of UAVs.
    2. The proposed sensor model for this relative positioning system enables the implementation of a tracking system that is able to stabilize the multi-UAV system without the help of additional sensors.
    3. The proposed sensor model for this relative positioning system can be used to characterize its performance according to the chosen design, with the intent of ranking the performance of different solutions, simplifying future design choices when facing different requirements, for example in terms of FOV or maximum detection range.

- Developing an infrared-based system with the following features:

    1. Its design has a weight that is at least two times lighter than the ones reported in the literature, and it also requires less power during its operation.
    2. The novel small omni-directional IR beacon design simplified the IR emission used by the systems reported in the literature, allowing the use of several emission sources for each UAV. This work shows that multiple emission sources on each UAV enables the positioning system to acquire the attitude measurements of neighboring UAVs (this feature is not present in other described infrared-based systems).

3. The developed localization and calibration algorithms for this relative positioning system allows the placement of IR receiver in arbitrary poses on the UAV, making the placement procedure easily adapted to any 3D geometry without the need of extra supporting structures. Additionally these algorithms also allow for a simplification of the system deployment process since placement errors on the IR receivers can be corrected through calibration.

Moreover, novel formation control and steering algorithms were proposed in this work with the goal of maximizing the reactiveness of the multi-UAV team in a way that is scalable with the number of used team members. These algorithms consist of improvements made on the formation control algorithms reported in previous literature in order to tackle limitations that arise from the use of relative positioning systems. The two main improvements are on:

- Increasing the number of inter-vehicle interactions in the presence of constraints on the used relative positioning systems (mainly the FOV constraints). It was previously discussed that sensing constraints can lead to a low number of possible formation configurations and can also reduce the reactiveness of the entire system. This work proposes a control algorithm that directly controls the FOV constraints of the onboard sensor, so they will be fulfilled during formation operation. This allows each UAV to optimize the FOV of its sensor when observing multiple neighbors.

- Moving the formation with only local relative localization information. As previously discussed, while some approaches make use of additional inter-vehicle communication, others lead to less reactive systems or to difficulties in specifying the desired system motion behaviors. This work proposes a control algorithm that maintains the formation reactiveness while remaining scalable with the number of UAVs (without using communication between UAVs) and adaptable to dynamic and unstructured environments.

## 1.2 Document overview

This document presents the sensor hardware and algorithm design that were considered to accomplish the contributions stated in the previous section. Chapter 2 reviews the previous work conducted on relative positioning systems, as well as the formation control algorithms that are normally applied to multi-agent systems. In this chapter the advantages and limitations of each sensing and control strategy are discussed. In Chapter 3 the mathematical background and functional architecture that is normally implemented to allow the control of multiple UAVs is introduced. In this chapter the main concepts of formation control and inter-vehicle localization are described. In Chapter 4, the UAV software architecture considered in this work is presented. In this chapter, the main assumptions that reflect simplifications made due to UAV motion characteristics, or limitations discussed in the previous sections, are introduced. Additionally, this chapter describes the main sensing and control blocks that will play the major roles in this architecture. Chapter 5 details the design and implementation of those main blocks. Firstly, the software and hardware designs of the two relative positioning systems considered in this work (camera-based and infrared-based) are presented. Afterwards, the design of the formation control algorithms that are able to directly control the sensing constraints of the system and that are able to move the formation in the environment are introduced. Chapter 6 describes the setups that were used to test and validate the developed relative positioning systems and formation control algorithms. This chapter focus on the description of the environments and platforms, as well as on how to implement the architecture discussed in Chapter 4 in the actual experimental setup. Chapter 7 presents the main results on the performance acquired for the developed relative positioning systems and formation control algorithms. At the end of this chapter, a discussion about these results is provided, where the contributions of this thesis described in the previous section are highlighted. Chapter 8 provides a summary of the previous discussion and concludes with the presentation of some implications and future research directions that originate from this work.

## 1.3 Publications during thesis work

- D. Dias, R. Ventura, P. U. Lima, and A. Martinoli. On-board vision-based 3d relative localization system for multiple quadrotors. *In International Conference on Robotics and Automation*, pages 1181-1187, 2016.

- D. Dias, P. U. Lima, and A. Martinoli. Distributed Formation Control of Quadrotors under Limited Sensor Field of View. *In International Conference on Autonomous Agents and Multi Agent Systems*, pages 1087-1095, 2016.

# Chapter 2

# Literature Review

Formation control of a multi-UAV system is characterized by inter-vehicle interactions with the goal of achieving certain inter-vehicle geometric constraints resulting from a targeted geometric configuration. This requires each vehicle to have relative inter-vehicle localization capabilities. In this chapter, a review of the main relative inter-vehicle localization approaches is performed in Section 2.1. Focus is given to the approaches relying on onboard relative positioning systems, as discussed in Chapter 1. Afterwards, the main approaches used for formation control and formation steering towards a goal are presented in Section 2.2. Here, attention is given to how the approaches tackle constraints enforced by the discussed onboard positioning systems. In particular, a review of quadrotor formation control approaches is provided in Section 2.2.4.

## 2.1 Onboard relative inter-vehicle localization

As previously discussed, formation control of a multi-UAV system requires each vehicle to collect relative positioning information of nearby team members. In most common approaches, the relative inter-vehicle localization is performed in two stages. First, vehicles acquire their positions in the environment by relying on external systems or using self-localization techniques. Then, the team members share their self-localization information via a communication channel that enables them to compute their relative positioning. However, when external systems are non-existent or when onboard computational resources are too limited, relative inter-vehicle localization is performed using dedicated onboard relative positioning systems providing directly the needed information.

Different onboard relative positioning systems exist, employing different technologies. The most mature and most used approaches are based on camera, sound, Radio Frequency (RF), and IR technologies. The typical procedure consists on the onboard sensor perceiving incoming signals originated from the nearby team members that are being tracked. The characteristics of these signals are then analyzed in order to obtain the team member identification, range, bearing and elevation information of the tracked team members relative to the onboard sensor. If multiple features can be extracted from the received signals, the relative orientation of the team members can also be computed. These positioning systems are characterized as passive when the incoming signal does not have to be generated by the vehicles themselves, such as many camera-based approaches that rely on ambient light bouncing on the nearby vehicles to generate the necessary features. Otherwise, they are characterized as active, such as approaches based on RF and IR technologies that generate electromagnetic signals. A review of the most used techniques for onboard relative inter-vehicle localization is provided in the next sections. The main focus is on UAV applications as they are the target vehicles in this work.

### 2.1.1 Camera-based

The most used onboard relative positioning systems are camera-based since the required hardware is mature, leading to its use in a wide variety of applications, including for UAVs. In these approaches the range, bearing, elevation information of nearby team members is extracted based on the object size and position in the image. The ID information, required by some multi-vehicle coordination methods, can also be extracted by observing the geometric shape or the color of the object. The simplest methods use single blob detection techniques [37, 93, 109, 111, 118], as depicted in Fig. 2.1a. Although this results in minimum computational requirements, these systems become dependent on the lighting conditions, shadows, or partial occlusions by the 3D structures, which can compromise the accuracy of range estimations. The results can be improved by using circle matching techniques in order to fully reconstruct the observed blob [2, 25, 109]. However, these methods are mainly exploited for gathering bearing and elevation information, as it measured based is only on the position of the blob in the image, a feature that is less affected by the previously mentioned lighting problems.

Range accuracy and blob detection reliability can be improved by using approaches that rely on the detection of multiple features on the nearby team members [117]. With these approaches it is often possible to additionally extract the team members' relative orientation. In the work reported in [29], depicted in Fig. 2.1b, the UAV is able to compute the relative position and orientation of its team member using the Perspective 3 Point (P3P) method. Such method has led to high precision measurements, for large inter-vehicle ranges, as reported in [15, 62]. Relative positioning accuracy can be further improved by fusing the vehicle egomotion with the relative positioning sensor measurements [22].

It is worth noting that approaches using active beacons for the detected features, as the ones in Fig. 2.1b, will end up generating the better results. In those approaches, the relative positioning system becomes more independent from light conditions (e.g. they can still perform under dark environment or in nocturnal applications). Finally, most approaches use CMOS camera technologies, which is characterized by lower frame rates and image blur at high speeds that can deteriorate and prevent feature detection. However, recent Dynamic Vision Sensors (DVS) technologies have been studied in order to solve this issue [67]. For example, the results in [19] show that it is possible to distinguish active markers from the environment using DVS technology.

One limitation of the camera-based approaches relates to the fact that some multi-vehicle coordination methods require each vehicle to have an unique ID. Simple blob detection methods implement this capability by designing colored markers. This approach can lead to limited scalability. with the number

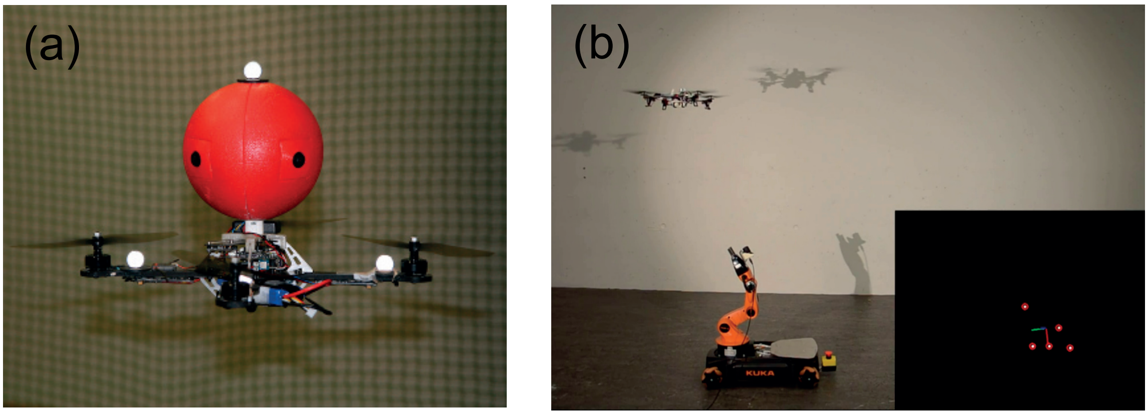


Figure 2.1: Different camera-based relative positioning sensor strategies. (a) Localization techniques based on blob detection [93]. Range is acquired from the observed blob size (using an a priori hardcoded information about the physical size of the blob), and bearing and elevation is computed from the blob position in the image. (b) Localization techniques based on multi-beacon detection [29]. This approach allows the computation of range, bearing, elevation, and orientation of the observed object, but at the expense of additional computation complexity.

of robots. Multi-feature detection methods implement this capability using different configurations of the feature 3D layout. This approach might lead to a cumbersome platform design methods since the features will have to be carefully placed on the vehicles. Additionally, the detection problem becomes combinatorial with the number of features which adds further computational complexities.

Another limitation of the camera-based approaches relates to their sensing capabilities, either on accuracy or FOV. This is especially true for the 3D case, because of the challenging sensing design, either due to the fact that the vehicle body represents an obstacle for the sensor itself, or because there is a tradeoff between the sensing area that needs to be covered, and the resolution of the sensor. This causes the existent approaches to rely on small sensor FOV. In a multi-vehicle system, this fact constrains the possible interactions between each vehicle. Inter-vehicle communication can be implemented in some cases to obtain the missing information at each vehicle. However, this makes the multi-vehicle system sensitive to packet loss or latency in communication links. This is important for UAVs since high control rates are necessary to stabilize the highly dynamical system formed by these vehicles, especially for short range inter-robot interactions that requires faster reaction times. Additionally, these approaches become sensible to sensing and actuation inaccuracies that can easily make UAV neighbors leave the FOV of the onboard camera (as in [96]). These situations might preclude the UAVs from recovering the desired configuration.

### 2.1.2 Sound-based

Another widely used relative positioning systems are the sound-based approaches. Sound waves travel in all directions and can be detected at long distances from the sound source, even in Non-Line-Of-Sight (NLOS) situations caused by obstacles or occlusions by the vehicles themselves. Additionally, the required information is computed from wave time of flight (TOF) measures, resulting in high accuracy results. Finally, microphones and sound emitters are small, lightweight, and can be easily be deployed on the UAVs without consuming too much power and with omnidirectional capabilities. This removes the previous FOV and accuracy limitations while maintaining the required infrastructure and computational complexity low.

One way to use these technology is to deploy an array of microphones scattered across the UAV and computing the angle of arrival (AOA) of sound emitter onboard nearby vehicles by measuring the Time Difference Of Arrival (TDOA) of the sound waves to each microphone, such as the work in [6–8] depicted in Fig. 2.2a. This allows the UAV to detect the bearing and elevation of nearby vehicles. Additionally, their relative positions can be acquired through triangulation and sensor fusion techniques [6]. Other approaches uses the sound system together with radio frequency (RF) synchronization in order to be able to measure the sound wave TOF [4, 11, 74, 90]. This allows the additional range estimation of nearby vehicles using the sound speed to convert the measured time to distance with high accuracies. Note that if the sound receivers are directional, as in [74] depicted in Fig. 2.2b1, it is also possible to acquire additional information about bearing and elevation of nearby vehicles.

However, sound-based approaches present several drawbacks. Firstly, multi-path interferences, echoes, and the presence of air flow disturbances may severely disturb the distance measurements between the emitter and receiver. Secondly, the system measuring frequency and bandwidth is limited by the speed of sound in the medium, which is relatively slower than electromagnetic signals. Additionally, as in [8, 74], when listening from sound from the environment, the UAVs many times have to turn off their motors in order to prevent them from contaminating the received signal with their noise. The systems are therefore effective when used in close range and in scenarios where Line-Of-Sight (LOS) and low conditions of air flow can be achieved, such as the local ultrasonic range and bearing system proposed in [90], depicted in Fig. 2.2b2. Note that low air flow conditions can not be achieved when considering quadrotors that produce air flow when generating thrust. Additionally, a relative small number of vehicles with slow dynamics are recommended (such as ground vehicles) when using these systems given the slow speed of sound. This is also a main limitation especially when highly dynamic vehicles such quadrotors, and multi-vehicle systems are considered.

### 2.1.3 Radio-signal-based

Other approaches rely on electromagnetic signals which are faster and in most cases do not receive interference from actuation systems. Due to their high wavelength characteristics, radio waves have low absorption rates through air and in solid materials, which makes them capable to travel large distances, indoors and outdoors. This makes radio waves ideal for communication, providing large ranges of operation, especially for NLOS operations. Most applications with mobile vehicles involve the use of onboard RF transceivers for communication. However, vehicle and inter-vehicle positioning is also possible with this type of technology. The advantage of using it for this purpose is that no other necessary hardware would be required. Positioning information is computed from the Received Signal Strength (RSS) values

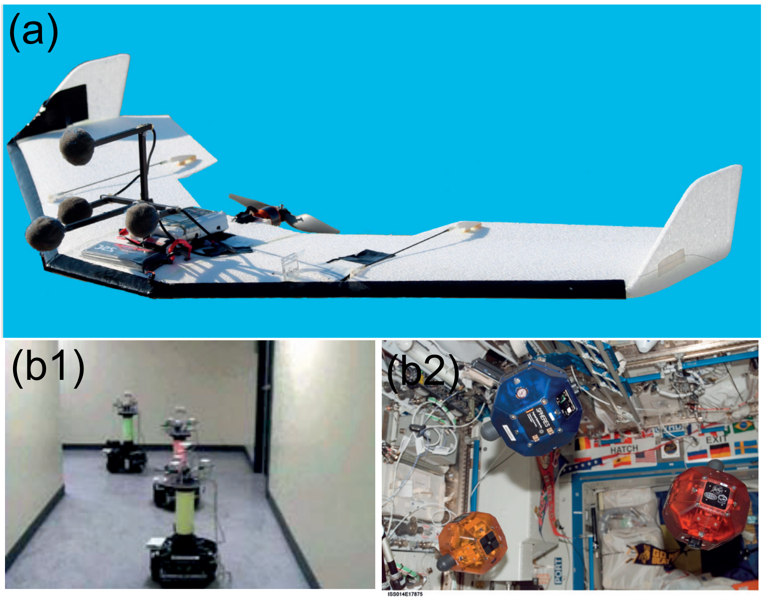


Figure 2.2: Different sound-based relative positioning sensor strategies. (a) Bearing measurements of a sound source are acquired using the TDOA of the sound source between several onboard microphones [8]. (b1 [90], b2 [74]) Range detection can be also computed by synchronizing the start of sound emission at the source with an RF signal moving at the speed of light, enabling TOF measurement from the sound source to the onboard microphones.

measured at static anchors [32,106] or mobile [53,66,79] transceiver devices tracking the RF sent by RF transceivers on board the vehicles.

The model for wave power dispersion in free space is used to obtain a relationship between the RSS and the range from the emitting source. Range-only measurements often require an additional triangulation [59] or trilateration algorithms to estimate the position between anchors with known positions. For unknown beacon positions or inter-vehicle localization multi dimensional scaling techniques can be applied [66,79]. For an extensive and a more thorough evaluation of the wireless positioning methodologies, the reader is referred to [68]. The antenna anisotropy usually needs to be considered in the previous models as most antennas have directional properties. However, some works take advantage of this fact to acquire additional orientation and bearing information. In [53], depicted in Fig. 2.3a, this factor is taken into account to allow the robots direct themselves to the targets. In [45], the antenna directionality properties are included to better estimate the position and orientation of the robot.

However, radio-signal-based approaches are subjected to radio wave multipath fading, due to its low frequency in the electromagnetic spectrum. This makes RSS values measured on RF transceivers not be easily expressed with some wave propagation model. Ultrawide-Band (UWB) approaches use additional spectrum for generating scheduled RF pulses, such as the work in [39,84]. This allows a higher penetrability and the ability to compute the wave TOF by tracking and rejecting multipaths. UWB technology has been recently used with UAVs [46]. However, this technology only allows for range extraction, and the share of information (such as velocity of the UAVs) and the implementation of additional sensor fusion algorithms need to employed. For now, this technology functions better with external anchor nodes scattered over large distances [48, 63, 84], as depicted in Fig. 2.3b, to account for extremely high speed of propagation of RF waves. This anchor-based system can consist of anchors receiving UWB signals coming from simple emitters on the vehicles [84] (allowing a simpler deployment on the vehicles), or of UWB transceivers placed on both the anchors and the vehicles [48, 63]. However, even with the static anchor configuration, wall interference problems caused in NLOS situations adds problems of measurement accuracy. To deal with accuracy issues caused to RSS and TOF measurements in NLOS situations, fingerprinting [16, 32, 84] or other calibration algorithms are usually applied to construct a signal map of entire area. However, if multiple RF transceivers are placed on board the vehicles, such method can not be applied because of the limited inter-transceiver distance enforced by the size of the vehicle (especially true for small-scale UAVs).

The main limitation of radio-signal-based approaches is then the accuracy that it is possible to achieve. For the 3D case the problem is increased since the wave propagation models become even more complex. In these cases, even if fingerprinting and other calibration methods are possible, its procedure is more cumbersome and sometimes unfeasible. For example, significant changes in the environment, such as moving furniture or large equipment, could require a reconstruction of the signal map [106]. Another problem is that the main source of positioning information that is possible to be acquired is inter-vehicle range. To obtain the bearing and elevation information additional algorithms are required, as previously discussed. This requires additional communication, which makes the system sensitive to packet loss or latency in communication links. As discussed in the camera-based approaches, this can lead to serious problems for UAVs and multi-UAV systems. These reasons are behind the fact that fewer applications use this approach as a 3D relative positioning system between UAVs.

### 2.1.4 Infrared-based

Due to their shorter wavelengths (from 700 *nm* to 1 *mm*), IR wave characteristics have interesting properties contrasting to radio waves, as discussed in [56]. Firstly, it can not penetrate through walls or other opaque barriers, making IR transmissions confined to the room in which they originate. Secondly, the short wavelength compared to the IR detector size leads to spatial diversity that prevents signal distortion caused by multipath fading. Finally, IR links must employ relatively high transmit power levels and operate over a relatively limited range. This makes IR preferred over RF for short range LOS communications where a maximum inter-vehicle link bit rate is desirable. Additionally these systems leverage cheap devices and simpler signal processing complexities. Much of the effort with this technology focus on modeling channel characteristics for communication [17, 18]. However, the low inter-emitter interference, caused by the short range emissions, and the absence of multipath fading also results in a clean reception signal. This makes this technology a good candidate for reliable indoor inter-vehicle localization.

A reduced amount of work was performed in this direction, mainly on 2D localization. Typical

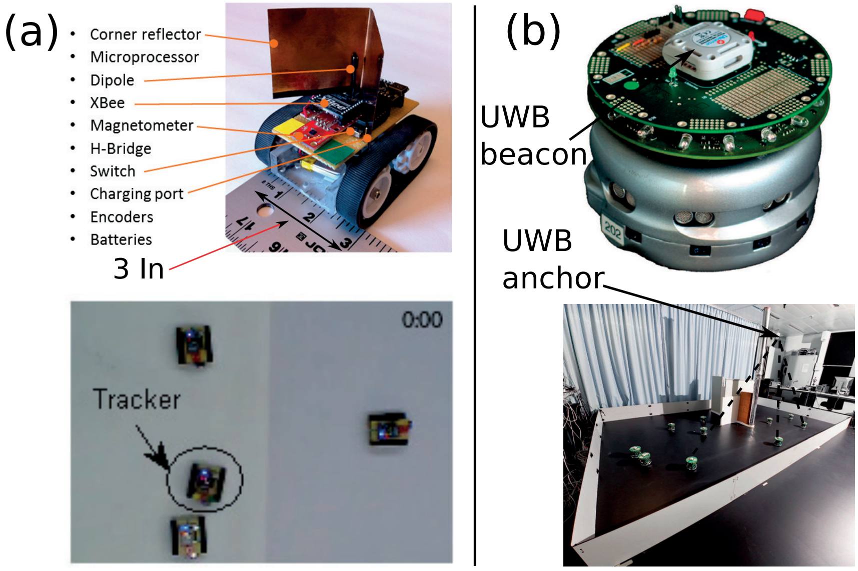


Figure 2.3: Different radio-signal-based relative positioning sensor strategies. (a) By measuring the RSS of receptors on board the robots, the range to mobile targets can be obtained using the relationship between signal decay in space and the distance from the radiation source located on board the targets. Directional transceivers can be exploited for bearing estimation [53]. (b) Ultra-wideband technology can be used to compute the distance to the radiation source using wave TOF. However these methods usually involve well separated external anchors given the high speed of these waves [84].

approaches place on board each vehicle a set of IR light emitting diodes (LEDs) that transmit IR signals. These signals are detected by the other vehicles with a set of onboard IR photodiodes playing the role of IR receivers. Signal generation and detection is performed using a sequence of techniques adopted from RF technology. The work developed with the Moorebots platform [60, 85] uses a frequency modulation (FM) approach, where a 455KHz carrier frequency is transmitted using the IR LEDs. Upon arrival to the IR receivers, the signal is filtered with a band-pass filter in the designated carrier frequency. This ensures the rejection of possible IR interference caused by the environment and other devices usually present in indoor environments. The filtered signal intensity is taken as a RSS value. These values can be used to localize the IR emission source, considered to be the position of the nearby vehicle. The range of the IR emission source can be computed by again interpreting the received RSS values using the model describing the wave power dispersion in free space. Due to the IR receiver strong directionality, the AOA of the IR signal describing the bearing of the IR emission source can be computed. This can be done by fusing the RSS values of a set of discrete IR receivers placed at different positions and orientations on the vehicle. In [60, 85] only four IR receivers were placed at each robot. In [86] an upgraded expansion board was developed for the Khepera III robots, depicted in Fig. 2.4a, with an increased carrier frequency of 10.7Mhz and placing at each robot eight receivers. This setup was able to achieve higher localization accuracy and communication speeds. Additionally, in [91] the received signal passes through a cascade of amplifiers to achieve greater Signal to Noise Ratio (SNR) and therefore achieve localization ranges.

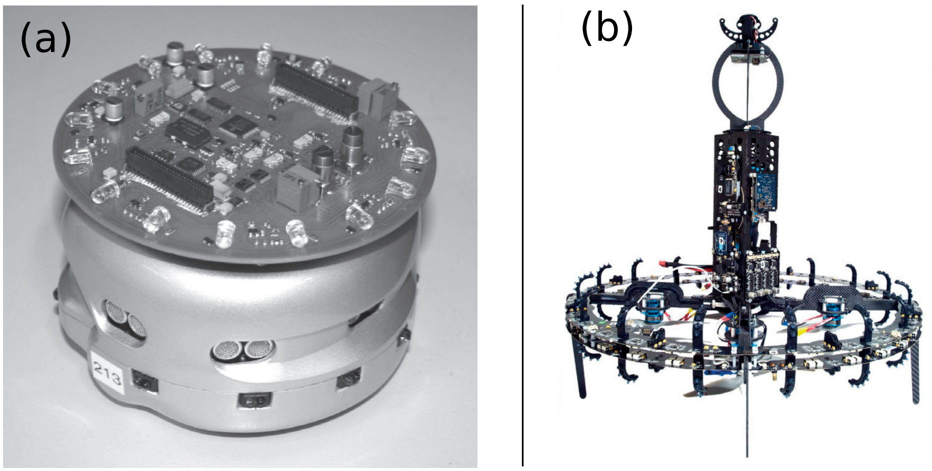


Figure 2.4: IR-based relative positioning sensor strategies. Directional receivers collect the IR light coming from a previously established and calibrated set of IR emitters deployed on the mobile vehicles. The range to the neighboring vehicles is obtained using the relationship between signal decay in space and the distance from the radiation source. The relative bearing of the vehicles is obtained by comparing the RSS values of different onboard IR receivers. (a) Planar board for 2D range and bearing estimation of nearby vehicles [86]. (b) Extension of IR technology to 3D space [92].

In these previous contributions, the emitter module was obtained by placing several directional IR LEDs on board each vehicle. The position and orientations of these LEDs were chosen to optimize emission homogeneity. However, emission intensity irregularities still produced significant errors in range computation, as shown in [86]. In spite of the previous issue, the sensor design is still simple for 2D inter-vehicle localization. For the 3D case, the design complexity is greatly increased. The work in [92], depicted in Fig. 2.4b, extended the homogeneous omni-directional emission intensity to the 3D case using a system of more than 100 IR LEDs. Also in this work, IR receivers are required to be placed at specific positions and orientations on the robot, which can be incompatible or cumbersome to achieve given the complexity of 3D robot geometries. These problems require the deployment of additional structure on the vehicles, substantially increasing the vehicle weight and the sensor deployment complexity.

Finally, in order to allow local communication that can scale with the number of vehicles, IR-based systems traditionally implement a communication protocol directly using the IR channel. An example of such protocol is the Carrier Sense Multiple Access (CSMA) with collision avoidance algorithm implemented in [86]. However, most recent works [91, 92] have also reintroduced a coupled RF channel to increase inter-vehicle communication throughput (for both LOS and NLOS situations). Additionally, this RF channel can also act as a global synchronization signal that simplify IR signal processing for the emission and reception, and thus allowing the positioning system to achieve larger measuring frequencies. However, this removes the local communication and synchronization characteristics of [86], which can lead to problems of scalability when increasing the number of vehicles.

## 2.2 Formation control

Multi-agent formation control consists of ensuring a set of geometrical constraints, such as inter-agent ranges and attitudes. A formation is employed whenever the target application requires the agents to achieve a specific geometric structure. Formation control methods are responsible to control the desired inter-agent geometric constraints that are required to achieve the desired formation structures. Quite often, these methods also include a formation steering component in order to move the group of agents towards a desired location in the environment. This implementation has to take into account the sensing and actuation constraints when real vehicles are considered. Next, a description of the most used formation control and steering methods is presented. The sensing and actuation constraints of real vehicles are discussed, with a specific focus on the previous relative position systems limitations and on the dynamic constraints of the quadrotor vehicles.

### 2.2.1 Formation control methods

Multi-agent formation control consists of ensuring a set of geometrical constraints, such as the inter-agent ranges and attitudes among team members. Initial works [5, 88] approached this problem by defining a set of control behaviors for each agent. If the agents were too close to each other or to an obstacle they would experience a repulsion force towards each other or the obstacle, in order to avoid collisions. On the other hand, if the agents were to far from each other they would experience an attraction force towards each other, in order to guarantee the cohesion of the multi-agent system. Additionally, the agents would try to align their velocities (in speed and in direction) in order to increase the maneuverability of the formation (or flock) in critical operations.

Several works [28, 65] implemented these behaviors through potential fields, function of the inter-agent ranges and attitudes with a minimum on the desired values. The designed controllers act in the multi-agent system in order to achieve that minimum value, which is equivalent to solving an optimization problem subject to the agent's motion dynamics. The simplest approaches are based on gradient descent algorithms, using direct feedback of the potential field gradient as control law. By building the overall system potential through a summation of potential functions defined for each isolated inter-agent constraint, decentralized controller design comes naturally. From the computed gradients, control laws for each agent become only dependent on the state of each agent and the states of the agents that are directly interacting with it. Stability and convergence of the proposed controllers are proven using non-linear and Lyapunov theory [61]. In [65], a user-defined potential function describes each inter-agent distance, with a minimum on the desired value. Agents sufficiently close to each other are attracted (if too far) or repelled (if too close), according to a decentralized controller using direct feedback from the potential field gradient as control law for the agent's acceleration (shown in Fig. 2.5).

Other approaches [27, 55, 77], denoted as graph-based, express the gradients using tools from graph theory, exploring the consensus problem. These approaches define each agent as a node in a graph with a certain position in an n-dimensional space. The desired formation geometry is defined as set of inter-

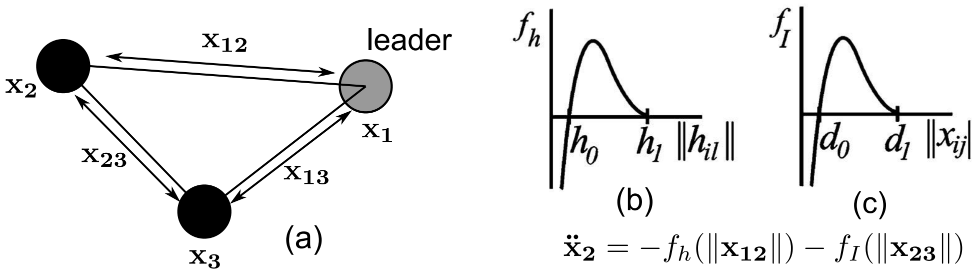


Figure 2.5: Example of potential field derived control forces in a formation. (a) configuration example; (b) potential gradient as in function of distance with respect to a leader; (c) the same gradient with respect to a follower. Distances above and below $d_0/h_0$ will respectively, attract and repel the robots. [65]

agent position biases to be satisfied, described by the edges of the graph. The consensus equation is then used to stabilize the multi-agent system to satisfy the desired biases. The controller design is simplified by using powerful analysis tools, based on the spectral properties of the connectivity, incidence, and graph-Laplacian matrices.

### 2.2.2 Formation steering methods

Typically, the formation moves in the environment towards some specified goal, while maintaining its geometric configuration. Several approaches exist to define the goal and move the multi-agent system, as described in [9], and depicted in Fig. 2.6. The first approach consists of leader-following, where a set of agents is designated as leaders while the rest of the agents are designated as followers. The leaders move according to predefined trajectories or moving directions given by their onboard sensors or by computational units external to the formation [21, 24, 30, 35]. The followers follow the leaders while maintaining the formation. The leaders can be defined as virtual elements common to all agents [33, 76]. This avoids bottlenecks and reject disturbances caused by sensor measurements, but it requires consistency on the leader perception by all the agents in the team. This is possible using communications, as for example in [33], which exploits a consensus problem for linear systems to achieve estimation of the formation center. In [76] each agent gives its position and temperature measurements to an external computational unit, that gives back to the agents the new position of the virtual leaders, computed from the received measurements in order to move towards a heat source. Stability of both formation movement and formation control is proven using Lyapunov theory, relating the maximum velocity of the leader with the designed controller [24, 75, 76].

The second approach consists involves the definition of a virtual structure representing the whole multi-agent system. In this approach, the desired kinematics of the virtual structure is first define, and then it is translated into the desired kinematics for each agent. Finally, control laws on board each agent are implemented in order to track the desired agent kinematics. In [28, 115] the entire path of the multi-agent system is defined by a set of virtual leaders to be followed by each real agent individually. The kinematics of each leader represent the desired kinematics of its associated agent, which is followed using control laws implemented on the agent.

Leader-follower approaches with physical leader agents are mostly used when high control rates are necessary to stabilize highly dynamical systems. Under such conditions, the consistency on the virtual leader perception by all the agents is sensitive to packet loss or latency in communication links. Physical leaders are also used when the agents have access to inter-agent localization only relative to their local frames, where no common reference frame exists to define virtual leaders, or virtual structures. An agent can control its relative inter-vehicle localization to a physical leader if it can sense the leader, or in other words, the agent is one sensing hop away from the leader. When the number of team members increases, the number of sensing hops between an agent and a leader tends to increase. In these cases, the agent is only capable of indirectly controlling its relative inter-vehicle localization to a leader by following other agents that can sense the leader. The increase of the number of sensing hops between an agent and the leader leads to an increase of the motion delay between the leader and the followers. This delay generates a distortion in the desired formation geometry, defined by the maximum number sensing hops between two agents. Although there are works assuring that these distortions are bounded ( [82, 107]), they assume the agents are within the area of detection of the sensor on board their neighbors. And for that to happen, the leader speed has to be limited, reducing the reactiveness of the entire formation. Note that, when the virtual structure approach can be applied these distortions can be substantially reduced since the motion of each agents is defined to move the multi-agent system as an single structure.

Behavior-based approaches add a control factor associated to the location of the task objective. In [5] the agents feel a force in the direction of a target goal, with an adjustable value. In [35] a potential field is defined for the distance between the agent and the goal. The agents use a gradient-descent approach to optimize their potential value. In [103] the group of agents inside an odor plume are able to sense the wind direction and move in formation correspondingly. While leader-follower and virtual structure approaches can be better controlled since trajectories are built for the problem, behavior-based approaches are more flexible in unpredictable environments, since simple behaviors (avoid obstacles and other robots, maintain distance to neighbors, move closer to the objective, etc.) typically hold.

Finally, the motion of the multi-agent system through obstacles has to be considered. Most works make each agent feel a repulsive force in the opposite direction from the obstacle, as in [34]. Formation deadlocks can appear when considering non-convex obstacles, since they can generate the presence of local minima in the used potential field functions. In [34] this problem is solved by having the agents keep track of the previous leader positions, and using them to move out form a current formation deadlock situation.

### 2.2.3 Actuation and sensing constraints

When the agents are implemented as real vehicles (e.g., differential drive vehicles, quadrotors, etc.), it must be pointed out that most of them have non-holonomic kinematics, meaning that their instantaneous moving directions are constrained by the vehicle and actuation configuration (see Fig. 2.7c). Therefore, the previous controllers can no longer be implemented as a direct feedback from the gradient descent

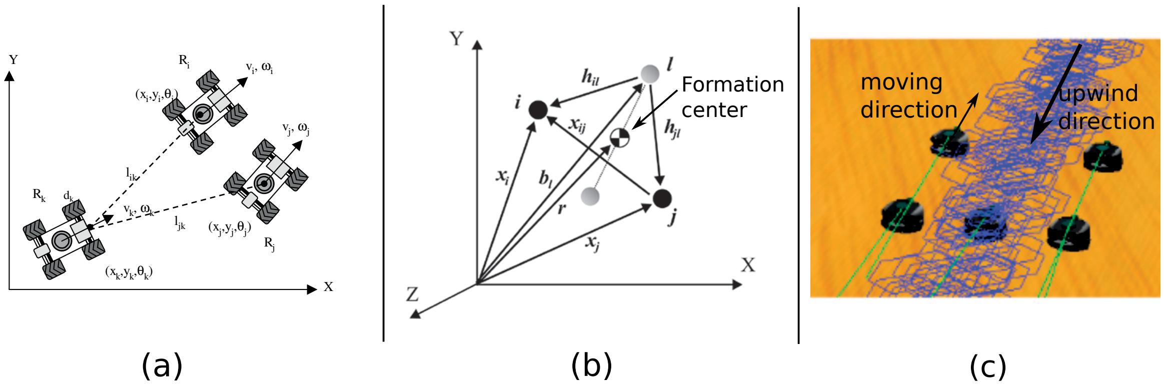


Figure 2.6: List of the several formation steering methods. (a) Leader-follower approach in [35], where a follower agent ($R_k$) track the movement of a set of leader agents ($R_i$, $R_j$); (b) Virtual leader approach in [76], where real agents (dark circles) are influenced by the movement of virtual agents (light circles and formation center);(c) Behavior-based approach in [103], where a formation of vehicles follow the upwind direction of a plume.

methods. The simplest approach to solve this issue is to adapt the controllers generated by direct feedback from the gradient descent to the constrained agent dynamics [31,42]. In [34], an holonomic point relative to the vehicle center, created from the vehicle dynamics, is used instead of the vehicle center in the formation control algorithms. Other approaches directly design the control methods solely resorting to nonlinear control theory [24, 28]. For UAVs, the previous methods have also to be adapted to deal with second order systems [41, 87] since their actuation is based on thrust.

Additionally, as discussed in previous sections, agents relying on onboard relative positioning sensors to extract the required information from the other team members, are dependent on the limitations of those sensors. These limitations are usually constraints either in maximum range and FOV and occasional occlusions by obstacles or by the members of the team. These problems are more predominant in the 3D case, because of the challenging sensing design, either due to the fact that the agent body represents an obstacle for the sensor itself, or because there is a compromise between the sensing area that needs to be covered and the resolution of the sensor. Fig. 2.7a and b illustrate the main issues behind these problems. One way to solve the problem is using communication. For example, in [37, 109], onboard cameras extract accurate relative bearing information, which is then used in a formation control algorithm. Given the limited FOV of the cameras, the information required from the nearby team members that are not directly observed is provided through communication. Also, in [31], occlusion problems are compensated by a communication network, where agents share their local measurements. To avoid too much communication overhead, information flow is limited to a maximum number of communication hops.

In real scenarios, when considering agents with fast dynamics, such as the quadrotors, high control rates are necessary to stabilize the resulting systems. In these cases, approaches become sensitive to packet loss or latency in communication links. This is particularly important for short range inter-agent

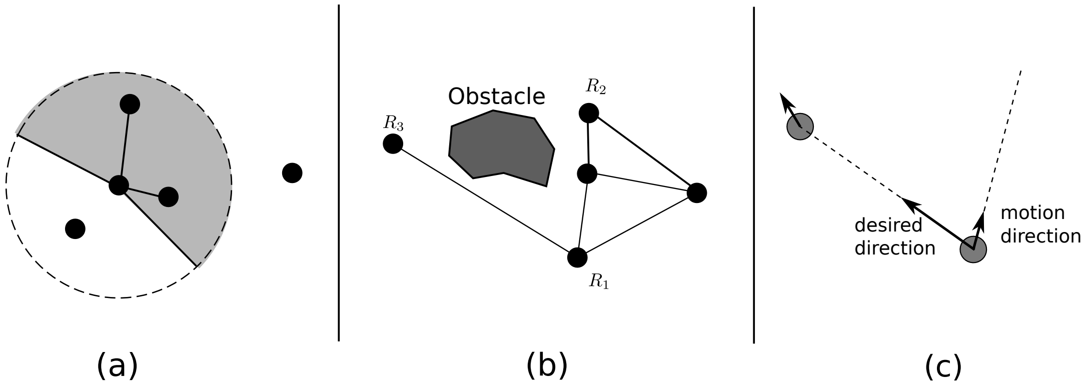


Figure 2.7: Coordination issues caused by the agents' actuation and sensing constraints. (a) Sensor range and FOV limitations (dashed circle and gray area respectively) can prevent the agent from sensing other team members, which can lead to formation braking/dissolving; (b) Occlusions can also prevent the sensing of other team members (e.g., seen from $R_2$, $R_1$ is occluded by another team member and $R_3$ by an obstacle); (c) Agent non-holonomicity requires the adaptation of the control algorithms to the agent-specific kinematics.

interactions, requiring even faster reaction times. Therefore, there is an interest in keeping communication to a minimum. In [105], the proposed framework uses both range and bearing information collected by each agent to reduce the amount of required observed team members per agent. However, if the required team members are not observed, communication is still used to recover the missing information. Sensor constraints can be directly introduced into the agents' control laws so that they can always be met, thus avoiding communication. This was done mostly for ground vehicles, and for sensor range constraints, using potential field algorithms that include specialized terms to guarantee that neighbor behavior will not compromise this type of constraint [54, 55, 83]. FOV constraints have also been considered, but just for tracking a single team member inside the sensor FOV [80, 81, 114]. This enforces the multi-agent to be bounded to a limited number of inter-agent connections, compromising the number of available formation geometries. Additionally, less connections means reduced system reactivity.

Also, many of the sensors onboard the agents in real scenarios can only provide inter-agent localization in the relative frame of the agent. In these situations, control algorithms must rely solely on measurements that are locally available. Several approaches use the inter-agent range. For example, the work in [31] adapts the previous graph-based approaches by converting the desired inter-agent position biases expressed in an absolute frame into range biases, so they can be used under relative agent frame assumptions. In that work, the authors prove the stability of the multi-agent system provided that enough inter-agent range are controlled, resulting into rigid control graphs (the reader is referred to [78] for more details about graph rigidity). However, proving convergence to a desired configuration is more difficult as the nonlinearities of the problem generate local minima in the used potential field functions (similar to the obstacle avoidance problem in Section 2.2.2). Other approaches use bearing measurements, such as in [37], although in this case the agents need to communicate additional information and a formation scale control needs to be added.

Finally, in order to be robust to the noise and the discrete availability of measurements provided by the onboard sensors, some approaches try to plan the trajectory according to some time of horizon [10, 43, 94]. In these approaches, each agent plans its next moves up to some time in the future, given the current information that it has about its goal and the localization of the other team members. The problem of these methodologies is that they require the use of extensive computational resources for this planning which limits the possible control rates that can be achieved. In case of quadrotor vehicles, the available resources on each vehicle makes it difficult to achieve high enough control rates necessary to cope with the dynamics of these system.

### 2.2.4 Quadrotor formation control

A popular approach for quadrotor formation control considered in the literature is to generate trajectories for the individual controllers to follow. These approaches require the existence of external systems for tracking and closely monitoring the control of the quadrotors. This allows quadrotors to carry minimal sensing and computational capabilities but makes them dependent on high reliable communication systems. In [110], trajectories are generated and managed in real time according to some time of horizon. The robots communicate their trajectories (built from polynomials) and recompute them after some time of horizon, given the communicated information from the nearby team members and the desired shape vectors. In the paper, the authors show that with the trajectory generation algorithm and the chosen communication scheme, the trajectories will become synchronized. In [3], trajectories are built off-line, in a centralized fashion, and then transmitted to the onboard trajectory controllers of the quadrotors. The trajectories are tested by feasibility checkers, based on the predicted acceleration necessary for the maneuvers. Also in [20] trajectory synchronization between robots is achieved by controlling the path following speed of each vehicle according to path tracking errors between neighbors.

Other approaches do not assume the presence of external systems and focus on the use of onboard relative positioning sensors to acquire the necessary inter-vehicle localization. In these approaches, the control laws are more reactive, producing less smooth trajectories. In [37, 98, 109] convergence of a team of quadrotors into a target configuration is achieved using mostly bearing information and communication. In that particular work, bearing information is preferred since it can be extracted with great accuracy from off-the-shelf cameras, as discussed on previous sections. Other works use the range, bearing, elevation, and the attitude of the nearby team members in order to achieve the convergence of the team of quadrotors to the target configuration [72]. In these works there is no need for any agreement on an absolute coordinate frame as before. Formation steering is performed either through teleoperation of the dynamics of a virtual struture [37, 98] or using a leader guiding the formation [96, 109]. The virtual structure approach, a consensus on the desired virtual structure dynamics as to be achieved using additional communication [37] or absolute localization in the environment [98] when relative inter-vehicle localization is used. Therefore, this approach becomes significantly more complex to achieve than the leader-follower approach.

# Chapter 3

# Theoretical Background

In this chapter the control and estimation concepts most relevant for the thesis concerning UAVs and more specifically quadrotor vehicles are presented. The dynamics of the considered quadrotor is first described in Section 3.2, followed by the proposed models used for single vehicle control and localization in Sections 3.3 and 3.4, based on the existing literature. Then, the inter-vehicle localization and formation control problems are formulated in Sections 3.5 and 3.6 respectively. Here and henceforth. the term UAV will refer to quadrotors, since these are the unmanned aerial vehicles used in the thesis.

## 3.1 Notation

Before presenting the basic definitions required in this work it is important to first define the notation used throughout this document. Most presented terms represent three dimensional position and attitude of objects in a three dimensional coordinate systems. This coordinate system is named as frame and denoted by the letter **I**. The main frame is the absolute or environment frame to which all objects relate to, and is denoted as $\mathbf{I}_{\mathscr{W}}$. It is also possible to refer to other local frames. For example, the body frame of object $i$ is denoted as $\mathbf{I}_{\mathscr{B}_i}$. The axes of each frame are represented by an additional subscript on the respective frame notation. For example, the environment frame axes are defined as $\mathbf{I}_{\mathscr{W}} = (\mathbf{I}_{\mathscr{W}_x}, \mathbf{I}_{\mathscr{W}_y}, \mathbf{I}_{\mathscr{W}_z})$, and the body frame of object $i$ axes are defined as $\mathbf{I}_{\mathscr{B}_i} = (\mathbf{I}_{\mathscr{B}_{ix}}, \mathbf{I}_{\mathscr{B}_{iy}}, \mathbf{I}_{\mathscr{B}_{iz}})$.

Vector variables expressed in a frame have the superscript of the frame, and the subscript identify the corresponding vector. For example, the position and the velocity of an object $i$ expressed in $\mathbf{I}_{\mathscr{W}}$ (defined in the next section) are denoted as $\mathbf{x}_i^{\mathscr{W}}$ and $\mathbf{v}_i^{\mathscr{W}}$ respectively. A second object $j$ position expressed in $\mathbf{I}_{\mathscr{W}}$ is denoted as $\mathbf{x}_j^{\mathscr{W}}$. However, the same variable expressed in $\mathbf{I}_{\mathscr{B}_i}$ is denoted as $\mathbf{x}_j^{\mathscr{B}_i}$. Although scalar variables do not require the frame superscript as they are invariant between frames, they can have superscripts to simplify the notation. An example of such a variable is the light energy of an IR beacon $b$ collected at an IR receiver $r$, $E_b^r$, presented in Section 5.1.2.

Variables that relate multiple objects are referred as inter-object variables. These variables include in their subscript the index of the objects. Inter-object vectors can be expressed in different frames. When the frame is one objects', the vector is classified as a relative variable. An example is the relative inter-object position of objects $i$ and $j$ expressed in object $i$'s frame $\mathbf{x}_{ij}^{\mathscr{B}_i}$. Note that $\mathbf{x}_{ij}^{\mathscr{B}_i} = \mathbf{x}_{j}^{\mathscr{B}_i}$. If the variable is expressed in the absolute frame then it is classified as an absolute variable.

The time dimension is also denoted in a similar way as before. The time of an event expressed in the absolute time-line is denoted as $\tau^*$. The same time measured in the object $i$'s time-line is $\tau^i$. The subscript of the time variable describes the event. For example, $\tau_{e_{jm}}^i$, presented in Section 5.1.2.1, refers to the emission time of marker $j$'s IR beacon $m$ expressed in UAV $i$'s time-line. The difference between two time instances $a$ and $b$ measured in the absolute time-line, $\tau_b^* - \tau_a^*$ (and measured in the object $i$'s time-line $\tau_b^i - \tau_a^i$) is denoted as $\tau_{ab}$. A generic time instance (not tight to any specific event) has no subscript, and therefore it is only denoted as $\tau^{\mathscr{X}}$, where $X$ identifies the time-line in which the time instance is expressed. To simplify the notation, the generic time instance expressed in the absolute time-line is simply denoted as $\tau$ (without the $*$ superscript). Any variable can be expressed in function of a time instance. For example, the position of object $i$ in function of a generic time instance is denoted as $\mathbf{x}_i^{\mathscr{W}}(\tau)$. In this work, time discretization is conducted by considering a sequence of time instances separated by a period of time $\Delta t$. A variable $X$ expressed in discrete time always refers to a specific time instance $k$ of that sequence. In this case the variable is denoted as $X(\tau_{start} + k\Delta t)$ or simply $X(k)$, where $\tau_{start}$ is the time instance when the discrete time sequence starts. Additional variables are denoted with the previously defined rules.

## 3.2 Quadrotor dynamics

Each quadrotor $i$ is has an absolute localization in the environment defined by the three dimensional position and attitude of its body frame, $\mathbf{I}_{\mathscr{B}_i}$, expressed in environment frame, $\mathbf{I}_{\mathscr{W}}$, as shown in Fig. 3.1a. The respective quadrotor position and attitude information form a quadrotor pose. The quadrotor position $\mathbf{x}_i^{\mathscr{W}}$ is defined by the three dimensional coordinates $\mathbf{x}_i^{\mathscr{W}} = (x_i^{\mathscr{W}}, y_i^{\mathscr{W}}, z_i^{\mathscr{W}})^T$ referred to the $\mathbf{I}_{\mathscr{W}}$'s origin. The quadrotor attitude is defined by the rotation between $\mathbf{I}_{\mathscr{W}}$ and $\mathbf{I}_{\mathscr{B}_i}$. In this work, rotations are expressed in their extrinsic form. This means that three Euler angles are defined to rotate a frame $\mathbf{I}_{\mathscr{X}}$ to a frame $\mathbf{I}_{\mathscr{Y}}$. Additionally, each Euler angle $(\phi, \theta, \psi)$ is responsible for rotations about a single axis of $\mathbf{I}_{\mathscr{X}}$ ($\mathbf{I}_{\mathscr{X}_x}$, $\mathbf{I}_{\mathscr{X}_y}$, $\mathbf{I}_{\mathscr{X}_z}$, respectively). In the case of the rotation between $\mathbf{I}_{\mathscr{W}}$ and $\mathbf{I}_{\mathscr{B}_i}$, the Euler angles are denoted $\phi_i$, $\theta_i$, and $\psi_i$ and each one is responsible for rotations about a single axis of this static absolute frame ($\mathbf{I}_{\mathscr{W}_x}$, $\mathbf{I}_{\mathscr{W}_y}$, and $\mathbf{I}_{\mathscr{W}_z}$, respectively). The rotations for each axis are defined by the matrices:

$$\mathbf{R}_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}, \quad \mathbf{R}_\theta = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad \mathbf{R}_\psi = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The full rotation matrix representing an attitude can be described as a product combination between those matrices, generating a rotation matrix in $SO(3)$. The combination defines the desired order of single axis rotations, and can be chosen at will. In this work, the order was chosen to be, $x$ axis, $y$ axis, and $z$ axis, and the respective rotation matrix is (note that a different ordering would lead to a different rotation description):

$$\mathbf{R} = \mathbf{R}_\psi \mathbf{R}_\theta \mathbf{R}_\phi = \begin{bmatrix} \cos(\psi)\cos(\theta) & \cos(\psi)\sin(\theta)\sin(\phi) - \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\phi) + \cos(\psi)\sin(\theta)\cos(\phi) \\ \sin(\psi)\cos(\theta) & \cos(\psi)\cos(\phi) + \sin(\psi)\sin(\theta)\sin(\phi) & \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi) \\ -\sin(\theta) & \cos(\theta)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix}.$$

Therefore, the attitude of UAV $i$ in the environment is defined by $\mathbf{R}_i^{\mathscr{W}} = \mathbf{R}_{\psi_i}\mathbf{R}_{\theta_i}\mathbf{R}_{\phi_i}$. In this work, the front of each frame is defined by the positive $x$ axis of that frame. The front of each quadrotor is considered to be coincident with the front of its body frame, consisting of its $x$ axis $\mathbf{I}_{\mathscr{B}_{ix}}$, depicted in Fig. 3.1b.

Quadrotor $i$ moves in 3D space using four propellers. The propellers rotate on the $xy$ plane of the $\mathbf{I}_{\mathscr{B}_i}$, generating thrust in the $\mathbf{I}_{\mathscr{B}_{iz}}$ direction. The thrust for each propeller $m$, $F_{im}$, is modeled as:

$$F_{im} = b_{im}\omega_{im}^2,$$

where $\omega_{im}$ is the angular velocity intensity of quadrotor $i$'s propeller $m$, and $b_{im}$ the respective thrust factor. As shown in Fig. 3.1b, propellers 1, 2, 3 and 4 are, respectively, in the front, right, rear, and left with respect to $\mathbf{I}_{\mathscr{B}_{ix}}$ direction. The total thrust of quadrotor $i$, $F_i$, is simply the sum of the thrusts generated by the four propellers:

$$F_i = b_{i1}\omega_{i1}^2 + b_{i2}\omega_{i2}^2 + b_{i3}\omega_{i3}^2 + b_{i4}\omega_{i4}^2. \tag{3.1}$$

From the previous definitions, the model of the quadrotor linear dynamics expressed in $\mathbf{I}_{\mathscr{W}}$ is as follows:

$$m\begin{bmatrix} \ddot{x}_i^{\mathscr{W}}(t) \\ \ddot{y}_i^{\mathscr{W}}(t) \\ \ddot{z}_i^{\mathscr{W}}(t) \end{bmatrix} = \mathbf{R}_i^{\mathscr{W}}(t)\begin{bmatrix} 0 \\ 0 \\ F_i(t) \end{bmatrix} - m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \tag{3.2}$$

where $R_i^{\mathscr{W}}(t)$ and $F_i(t)$ are respectively, the rotation matrix defining the UAV $i$ attitude and the applied thrust at time $t$, $m$ is the total mass of the UAV, and $g$ is the gravity acceleration. Also, each propeller generates torque in the opposite direction of its angular velocity. Since the propellers rotate on the $xy$ plane of the $\mathbf{I}_{\mathscr{B}_i}$, the torque is applied in the $\mathbf{I}_{\mathscr{B}_{iz}}$ axis. The torque generated by each propeller can be defined in the center of the UAV and its intensity, $T_{im_z}$, is defined by:

$$T_{im_z} = d_{im}\omega_{im}^2,$$

where $d_{im}$ is the drag factor of UAV $i$'s propeller $m$, considering a vehicle with symmetric geometry. To prevent the UAV from spinning about the $\mathbf{I}_{\mathscr{B}_{iz}}$ axis, propellers 1 and 3 rotate in the opposite direction of propellers 2 and 4, as described in Fig. 3.1b. The total torque about $\mathbf{I}_{\mathscr{B}_{iz}}$ is:

$$T_{i_z} = -d_{i1}\omega_{i1}^2 + d_{i2}\omega_{i2}^2 - d_{i3}\omega_{i3}^2 + d_{i4}\omega_{i4}^2. \tag{3.3}$$

Note that although the propellers rotate in opposite directions, they all generate thrust in the $\mathbf{I}_{\mathscr{B}_{iz}}$ direc-

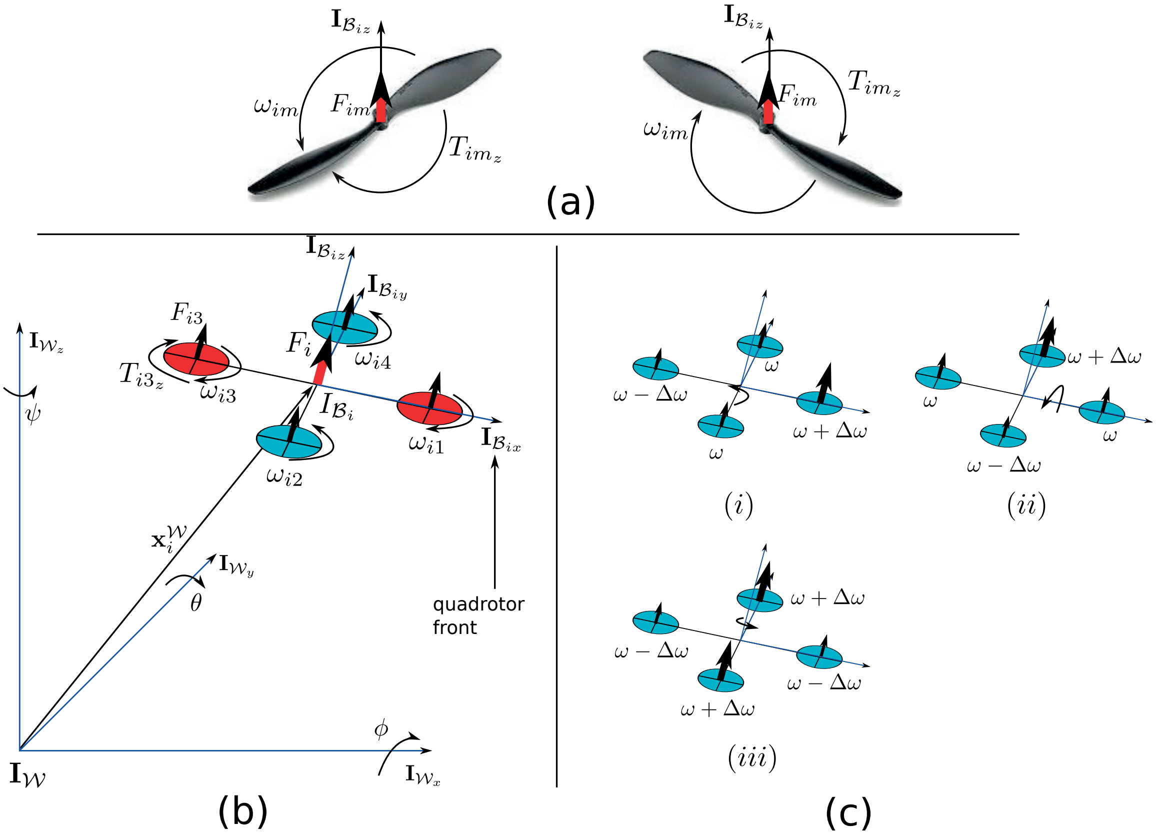


Figure 3.1: Description of the quadrotor physical model. (a) Different propeller manufacturing. Different propellers rotate in opposite directions but still generate thrust in the same direction. (b) Definition of body, $\mathbf{I}_{\mathscr{B}_i}$, and environment, $\mathbf{I}_{\mathscr{W}}$, frames. Propeller thrust is applied orthogonally to the UAV $xy$ plane. Propellers 1 and 3 (red) rotate in opposite direction of propellers 2 and 4 (blue). (c) Cases for torque generation over the body axes. Rotations around $\mathbf{I}_{\mathscr{B}_{ix}}$ (i) or $\mathbf{I}_{\mathscr{B}_{iy}}$ (ii) occur during differences in generated thrust for propellers in the orthogonal axis. Rotations around $\mathbf{I}_{\mathscr{B}_{iz}}$ (iii) occur during differences in generated thrust on each axis.

tion. Therefore the propellers 1 and 3 are manufactured differently than propellers 2 and 4, as illustrated in Figs. 3.1a. Finally, as illustrated in Fig. 3.1c, if the propellers of the same axis spin with different velocities, different thrusts are applied on the axis edges, generating a torque of intensity:

$$T_{i_x} = l(b_{i4}\omega_{i4}^2 - b_{i2}\omega_{i2}^2), \qquad T_{i_y} = l(b_{i1}\omega_{i1}^2 - b_{i3}\omega_{i3}^2), \tag{3.4}$$

where $l$ is the length of the arm between the center of the vehicle and the propeller, and $T_{i_x}$ and $T_{i_y}$ are the generated torques about $\mathbf{I}_{\mathscr{B}_{ix}}$ and $\mathbf{I}_{\mathscr{B}_{iy}}$ respectively. Thus, the attitude dynamics expressed in $\mathbf{I}_{\mathscr{B}_i}$ can be derived:

$$\mathbf{J}_i \begin{bmatrix} \dot{\omega}_{i_x}(t) \\ \dot{\omega}_{i_y}(t) \\ \dot{\omega}_{i_z}(t) \end{bmatrix} = \begin{bmatrix} T_{i_x}(t) \\ T_{i_y}(t) \\ T_{i_z}(t) \end{bmatrix} - \begin{bmatrix} \omega_{i_x}(t) \\ \omega_{i_y}(t) \\ \omega_{i_z}(t) \end{bmatrix} \times \mathbf{J}_i \begin{bmatrix} \omega_{i_x}(t) \\ \omega_{i_y}(t) \\ \omega_{i_z}(t) \end{bmatrix}, \tag{3.5}$$

where $(\omega_{i_x}(t),\omega_{i_y}(t),\omega_{i_z}(t))^T$ and $(T_{i_x}(t),T_{i_y}(t),T_{i_z}(t))^T$ are respectively, the angular velocities and the applied torques about each body axis at time $t$, and $\mathbf{J}_i$ is a 3x3 matrix defining the moment of inertia of the UAV. The second term of the right hand side of (3.5) describes the gyroscopic effects, responsible for conserving angular momentum in the absence of external forces. The attitude expressed in $\mathbf{I}_{\mathscr{W}}$ can be described in terms of the angular velocities on the body axes as follows [23]:

$$\dot{\mathbf{R}}_i^{\mathscr{W}}(t) = \mathbf{R}_i^{\mathscr{W}}(t)\hat{\mathbf{w}}_i(t), \qquad \hat{\mathbf{w}}(t) = \begin{bmatrix} 0 & \omega_{i_z}(t) & \omega_{i_y}(t) \\ \omega_{i_z}(t) & 0 & -\omega_{i_x}(t) \\ -\omega_{i_y}(t) & \omega_{i_x}(t) & 0 \end{bmatrix}.$$

The model for the quadrotor dynamics expressed in the environment frame $\mathbf{I}_{\mathscr{W}}$ considered in this work can now be completely defined:

$$\begin{cases} \dot{\mathbf{x}}_i^{\mathscr{W}}(t) = \mathbf{v}_i^{\mathscr{W}}(t) \\ m\dot{\mathbf{v}}_i^{\mathscr{W}}(t) = F(t)\mathbf{R}_i^{\mathscr{W}}(t)\mathbf{I}_{\mathscr{W}_z} - mg\mathbf{I}_{\mathscr{W}_z} \\ \dot{\mathbf{R}}_i^{\mathscr{W}}(t) = \mathbf{R}_i^{\mathscr{W}}(t)\hat{\mathbf{w}}_i \\ \mathbf{J}_i\dot{\mathbf{w}}_i(t) = \mathbf{T}_i - \mathbf{w}_i(t) \times \mathbf{J}_i\mathbf{w}_i(t) \end{cases}, \tag{3.6}$$

where $\mathbf{x}_i^{\mathscr{W}} = (x_i^{\mathscr{W}}, y_i^{\mathscr{W}}, z_i^{\mathscr{W}})^T$ is $\mathbf{I}_{\mathscr{B}_i}$'s absolute position, $\mathbf{v}_i^{\mathscr{W}} = (v_{i_x}^{\mathscr{W}}, v_{i_y}^{\mathscr{W}}, v_{i_z}^{\mathscr{W}})^T$ is the linear velocity, and $\mathbf{w}_i = (\omega_{i_x}, \omega_{i_y}, \omega_{i_z})^T$ is the angular velocity.

## 3.3 Single quadrotor control

Considering the previous quadrotor dynamics, it is possible to provide a controller for its movement in the environment. The UAV states to be controlled are the position $\mathbf{x}_i^{\mathscr{W}} = (x_i^{\mathscr{W}}, y_i^{\mathscr{W}}, z_i^{\mathscr{W}})$, velocity $\mathbf{v}_i^{\mathscr{W}} = (v_{i_x}^{\mathscr{W}}, v_{i_y}^{\mathscr{W}}, v_{i_z}^{\mathscr{W}})$, and the attitude $(\phi_i, \theta_i, \psi_i)$. From (3.6), one can see that the attitude dynamics is completely separated from the position dynamics. This allows the separation of the control problem into two decoupled parts: one for the attitude and another for position. The position can be further divided into height control and horizontal position control. Attitude and height control are simpler as they can be directly controlled through the generated torques and thrusts of the vehicle (example of such controllers in [13]). Horizontal position control is more complex since the vehicle is under-actuated (it has to turn its propellers to the desired direction). One simple way is to use the propeller thrust $F_i$ to control movement in the $\mathbf{I}_{\mathscr{W}_z}$ axis, and the $(\phi_i, \theta_i)$ angles to control movement in the $\mathbf{I}_{\mathscr{W}_y}$ and $\mathbf{I}_{\mathscr{W}_x}$ axes respectively, as in [20,71]. Since a large amount of study is already done for attitude control with Inertial Measurement Unit (IMU) sensors [95,104,108], this work considers that the UAV Euler angles are known and controlled with an already working attitude controller. This controller is usually implemented by the auto-pilots of available commercial UAVs.

The previous auto-pilots implement the lower level control of the UAV by producing the desired (subscript 'd') acceleration inputs. The UAV acceleration is computed from the linear model, in (3.6), as follows:

$$\begin{bmatrix} a^{\mathcal{W}}_{i_x,d}(t) \\ a^{\mathcal{W}}_{i_y,d}(t) \\ a^{\mathcal{W}}_{i_z,d}(t) \end{bmatrix} = \frac{F_i(t)}{m} \begin{bmatrix} \sin(\psi_i)\sin(\phi_i)+\cos(\psi_i)\sin(\theta_i)\cos(\phi_i) \\ \sin(\psi_i)\sin(\theta_i)\cos(\phi_i)-\cos(\psi_i)\sin(\phi_i) \\ \cos(\theta_i)\cos(\phi_i) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \begin{bmatrix} f^{\mathcal{W}}_{i_x}(t) \\ f^{\mathcal{W}}_{i_y}(t) \\ f^{\mathcal{W}}_{i_z}(t) \end{bmatrix}, \tag{3.7}$$

where $\mathbf{a}^{\mathcal{W}}_{i,d} = (a^{\mathcal{W}}_{i_x,d}, a^{\mathcal{W}}_{i_y,d}, a^{\mathcal{W}}_{i_z,d})^T$ corresponds to the desired acceleration expressed in the environment frame. Also, $\mathbf{f}^{\mathcal{W}} = (f^{\mathcal{W}}_{i_x}, f^{\mathcal{W}}_{i_y}, f^{\mathcal{W}}_{i_z})^T$ are the components of the disturbance vector characterizing unmodeled aerodynamics, parameter identification errors and attitude estimation errors, following the lines of [71]. Most common indoor control methods consider simple vehicle dynamics and disregard aerodynamic effects, given the low vehicle velocities and the lack of powerful wind gusts [52], so many times this quantity is set to zero. Solving Eq. (3.7) in order to obtain $\mathbf{a}^{\mathcal{W}}_{i_d}$ as in function of the actual quadrotor inputs $(F_{i,d}, \phi_{i,d}, \theta_{i,d}, \psi_{i,d})^T$ is not straightforward. The most simple way to get around the problem is to consider the quadrotor in a quasi-hovering situation, consisting of small angle displacements for $\phi_i$ and $\theta_i$ [49, 71, 102]. This allows the following approximations: $cX \approx 1$ and $sX \approx X$, $X = \phi_i$ or $\theta_i$. With this assumption, Eq. (3.7) can be redefined as follows:

$$\begin{cases} \frac{F_{i,d}}{m} = g + a^{\mathcal{W}}_{i_z,d} - f^{\mathcal{W}}_{i_z} \\ \begin{bmatrix} \phi_{i,d} \\ \theta_{i,d} \end{bmatrix} = \left(\frac{F_{i,d}}{m}\right)^{-1} \begin{bmatrix} \sin(\psi_{i,d}) & -\cos(\psi_{i,d}) \\ \cos(\psi_{i,d}) & \sin(\psi_{i,d}) \end{bmatrix} \begin{bmatrix} a^{\mathcal{W}}_{i_x,d} \\ a^{\mathcal{W}}_{i_y,d} \end{bmatrix} - \begin{bmatrix} f^{\mathcal{W}}_{i_x} \\ f^{\mathcal{W}}_{i_y} \end{bmatrix} \end{cases} . \tag{3.8}$$

The desired $\phi_i$ and $\theta_i$ values are passed to the working attitude controller as reference values. Note that the $\psi_i$ angle is not required for controlling the acceleration. Therefore it is possible to control it independently. Most of the existent auto-pilots, such as the one used in [1], provide an independent controller for this angle, usually on the $z$ axis angular velocity $\omega_{i_z}$.

The torques applied on each UAV axis $(T_{i_x}, T_{i_y}, T_{i_z})$ are computed in order for the UAV to reach the previous angular control values. The desired thrust $F_{i,d}$ is passed to the propeller speed controller as a reference value. Using Eqs. (3.1), (3.3) and (3.4) with the previously computed $(F_{i,d}, T_{i_x}, T_{i_y}, T_{i_z})$, the autopilot is able to compute the appropriate propeller speeds. The disturbance vector can be estimated by fusing positioning and actuation information through time, as described in the next sections.

Other approaches [14, 64, 69] deal with Eq. (3.7) in a different way. They consist of turning the UAV to the desired thrust direction considering the true UAV rotation, which involves more complex computations. The propeller thrust is computed from the projection of the desired force vector into the propeller axis. The $\theta_i$ and $\phi_i$ angles are computed in order to lead the propeller axis to the desired force direction satisfying the desired $\psi_i$ angle. Since single UAV control is not the main scope of this work, the first simpler approach is adopted.

Using the previously described auto-pilot, it is possible to define a control algorithm that moves the quadrotor in the environment. This is done by first computing the desired accelerations which are then given to the UAV auto-pilot. These desired accelerations can be computed for example taking a desired

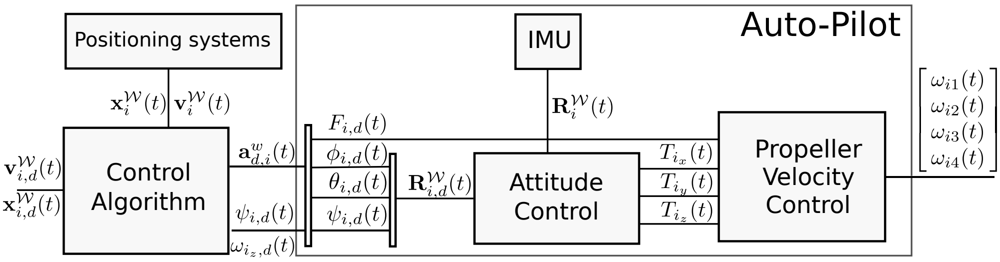


Figure 3.2: Sketch of the considered control scheme for UAV *i*. The control algorithm uses the control law in Eq. (3.10).

position and velocity as a goal. Here, position and velocity errors are considered as:

$$\mathbf{e}_{p_i}^{\mathscr{W}} = \mathbf{x}_{i,d}^{\mathscr{W}} - \mathbf{x}_i^{\mathscr{W}}, \qquad \mathbf{e}_{v_i}^{\mathscr{W}} = \mathbf{v}_{i,d}^{\mathscr{W}} - \mathbf{v}_i^{\mathscr{W}}, \tag{3.9}$$

where $\mathbf{x}_{i,d}^{\mathscr{W}}$ and $\mathbf{v}_{i,d}^{\mathscr{W}}$ are respectively, the desired 3D position and velocity of the UAV. Since the goal is for the quadrotor to achieve a certain position, the desired velocity is set to zero. Previous work proved that it is possible to control the UAV to any configuration in 3D space by defining $\mathbf{a}_{i,d}^{\mathscr{W}}$ values through position and velocity Proportional, Integral and Derivative (PID) controllers on the respective errors. The desired UAV acceleration, $\mathbf{a}_{i,d}^{\mathscr{W}}$, is then defined as follows:

$$\mathbf{a}_{i,d}^{\mathscr{W}} = \mathbf{K}_p \mathbf{e}_{p_i}^{\mathscr{W}} + \mathbf{K}_v \mathbf{e}_{v_i}^{\mathscr{W}}, \tag{3.10}$$

where $\mathbf{K}_p$ and $\mathbf{K}_v$ are diagonal matrices, with gains $(k_{px}, k_{py}, k_{pz})$ and $(k_{vx}, k_{vy}, k_{vz})$ respectively, to allow different control responses in height and horizontal dimensions. The desired acceleration can now be passed to the previous described auto-pilots running on similar equations to Eq. (3.8). The position and the velocity can be acquired by localization methods using onboard absolute positioning systems [38, 50, 51, 113]. Here, ultrasonic/laser sensors can be used to measure UAV height, while Optic-Flow (OF) sensors measure the UAV velocity, and camera or Laser Range Finder (LRF) positioning systems acquire the horizontal position of the UAV. As it will be seen in this work, it is also possible to control one UAV with respect to another UAV using the information provided by inter-vehicle localization methods that use onboard relative positioning systems, such as the ones described in the following sections. The block diagram of the chosen control scheme is shown in Fig. 3.2.

Note that, although $\theta_i$ and $\phi_i$ angles are always obtained with respect to the direction of gravity, the $\psi_i$ angle is extracted using the onboard magnetometer of the IMU sensors. These magnetometers can produce high error measurements when in presence of close metallic objects, which can be common in indoor environments. Measurements of $\psi_i$ can also be obtained with UAV onboard absolute positioning systems (e.g., a LRF), but in many situations and for simpler UAVs they are not available. In these cases, $\psi_i$ measurements can become too noisy for this angle be controlled to its desired value $\psi_{i,d}$. However, some auto-pilots are still able to provide $\psi_i$ control using angular velocity around the $z$ axis, $\omega_{i_z,d}$, as previously described, since this quantity can be measured by the more accurate gyroscopes from the IMU sensor. In case $\psi_i$ measurements are not reliable, the controller defined in (3.8) has to be adapted, as it will be further discussed in the next chapter.

## 3.4 UAV self-localization

The previous control laws require the knowledge of the quadrotor pose in the environment. The attitude is provided by the onboard IMU. A positioning system is needed to provide the position. The used positioning systems for quadrotors typically assume that the quadrotor moves according to a constant speed model. This model can be achieved by using the linear model of Eq. (3.6), expanded in Eq. (3.2), with a simple discretized inertial navigation model, similarly to [69]. The UAV $i$ position and velocity $(\mathbf{x}_i^{\mathscr{W}}, \mathbf{v}_i^{\mathscr{W}})$ are assumed to evolve in time as follows:

$$\begin{bmatrix} \mathbf{x}_i^{\mathscr{W}}(k+1) \\ \mathbf{v}_i^{\mathscr{W}}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I_{3x3}} & \Delta t \mathbf{I_{3x3}} \\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^{\mathscr{W}}(k) \\ \mathbf{v}_i^{\mathscr{W}}(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2}{2} \mathbf{I_{3x3}} \\ \Delta t \mathbf{I_{3x3}} \end{bmatrix} \mathbf{a}_i^{\mathscr{W}}(k+1), \tag{3.11}$$

where $\mathbf{I_{ixj}}$ and $\mathbf{0_{ixj}}$ are respectively, identity and zero matrices with ixj dimension, $\Delta t$ is the chosen time step for discretization, and $\mathbf{a}_i^{\mathscr{W}}(k) = (a_{i_x}^{\mathscr{W}}(k), a_{i_y}^{\mathscr{W}}(k), a_{i_z}^{\mathscr{W}}(k))^T$ is the acceleration applied to the UAV at step $k$, expressed in the environment frame, $\mathbf{I}_w$. From Eq. (3.7), it is possible to write the acceleration as follows (note that the current UAV attitude, $\mathbf{R}_i^{\mathscr{W}}(k)$, is assumed known at every time step):

$$\mathbf{a}_i^{\mathscr{W}}(k+1) = \frac{F_i(k+1)}{m} \mathbf{R}_i^{\mathscr{W}}(k) \mathbf{I}_{\mathscr{W}_z} + \xi_i^{\mathscr{W}}(k+1) - g \mathbf{I}_{\mathscr{W}_z} + \mathbf{f}_i^{\mathscr{W}}(k), \tag{3.12}$$

where $\xi_i^{\mathscr{W}}(k) = (\xi_{i_x}^{\mathscr{W}}(k), \xi_{i_y}^{\mathscr{W}}(k), \xi_{i_z}^{\mathscr{W}}(k))^T$ is an additional stochastic term added to account for actuation and attitude estimation errors. This term is modeled as a zero mean white Gaussian signal, with a diagonal covariance matrix, $Q_{\xi_i^{\mathscr{W}}}$. The parameter $\mathbf{f} = (f_x^{\mathscr{W}}, f_y^{\mathscr{W}}, f_z^{\mathscr{W}})^T$ characterizes important perturbations, such as errors in model parameter identification and the unmodeled aerodynamics, already present in Eq. (3.7). The dynamics of $\mathbf{f_i^{\mathscr{W}}}$ is defined as a slow time varying quantity disturbance:

$$\mathbf{f}_i^{\mathscr{W}}(k+1) = \mathbf{f}_i^{\mathscr{W}}(k) + \gamma_i^{\mathscr{W}}(k+1), \tag{3.13}$$

where $\gamma_i^{\mathscr{W}}(k) = (\gamma_{i_x}^{\mathscr{W}}(k), \gamma_{i_y}^{\mathscr{W}}(k), \gamma_{i_z}^{\mathscr{W}}(k))^T$ is the term defining the uncertainty on the included disturbance vector, in order to model the time varying behavior. This term is also modeled as a zero mean white Gaussian signal, with a diagonal covariance matrix, $Q_{\gamma_i^{\mathscr{W}}}$. This disturbance vector is to be estimated alongside the vehicle position and velocity. For indoor environments or experiments where perturbations seem to be negligible, this term can be put aside. Merging expressions (3.11) to (3.13), a motion model for the UAV can be derived as follows:

$$\begin{bmatrix} \mathbf{x}_i^{\mathcal{W}}(k+1) \\ \mathbf{v}_i^{\mathcal{W}}(k+1) \\ \mathbf{f}_i^{\mathcal{W}}(k+1) \end{bmatrix} = \mathbf{A} \begin{bmatrix} \mathbf{x}_i^{\mathcal{W}}(k) \\ \mathbf{v}_i^{\mathcal{W}}(k) \\ \mathbf{f}_i^{\mathcal{W}}(k) \end{bmatrix} + \mathbf{B}\left(F_i(k+1)\mathbf{R}_i^{\mathcal{W}}(k)\mathbf{I}_{\mathcal{W}_z} - g\mathbf{I}_{\mathcal{W}_z}\right) + \mathbf{W} \begin{bmatrix} \xi_i^{\mathcal{W}}(k+1) \\ \gamma_i^{\mathcal{W}}(k+1) \end{bmatrix} \tag{3.14}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{I_{3x3}} & \Delta t\mathbf{I_{3x3}} & \frac{\Delta t^2}{2}\mathbf{I_{3x3}} \\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}} & \Delta t\mathbf{I_{3x3}} \\ \mathbf{0_{3x3}} & \mathbf{0_{3x3}} & \mathbf{I_{3x3}} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{\Delta t^2}{2}\mathbf{I_{3x3}} \\ \Delta t\mathbf{I_{3x3}} \\ \mathbf{0_{3x3}} \end{bmatrix}, \quad \mathbf{W} = \begin{bmatrix} \frac{\Delta t^2}{2}\mathbf{I_{3x3}} & \mathbf{0_{3x3}} \\ \Delta t\mathbf{I_{3x3}} & \mathbf{0_{3x3}} \\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}} \end{bmatrix}.$$

Note that no correlation between $\gamma_i^{\mathcal{W}}(k)$ and $\xi_i^{\mathcal{W}}(k)$ is considered. To deal with model uncertainty, height and horizontal position measurements are assumed available using the previously described absolute positioning sensors (e.g., LRF or camera-based systems). The respective measurement model is:

$$\mathbf{x}_{o_i}^{\mathcal{W}}(k) = \begin{bmatrix} \mathbf{I_{3x3}} & \mathbf{0_{3x3}} & \mathbf{0_{3x3}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^{\mathcal{W}}(k) \\ \mathbf{v}_i^{\mathcal{W}}(k) \\ \mathbf{f}_i^{\mathcal{W}}(k) \end{bmatrix} + \eta_{i1}^{\mathcal{W}}, \tag{3.15}$$

where $\mathbf{x}_{o_i}^{\mathcal{W}}$ is the measured UAV 3D position, and $\eta_{i1}^{\mathcal{W}}$ is an uncorrelated zero mean white Gaussian signal modeling sensor uncertainty, with a covariance matrix of $R_{\eta_{i1}^{\mathcal{W}}}$. The UAV velocity can also be determined using the previously described OF sensors. These camera-based sensors are usually set up to capture the movement of ground or ceiling features in the sensor frame. The UAV velocity expressed in the world frame is then achieved by transforming the received measurements using the current UAV attitude. Given that quadrotor vehicles are assumed to be in a quasi hovering conditions, $\phi_i$ and $\theta_i$ angles can be consider small, and only the rotation about the $z$ axis effectively matters for the transformation. This allows the UAV to measure its horizontal velocity. The respective measurement model then becomes:

$$\mathbf{v}_{o_{ih}}^{\mathcal{W}}(k) = (\mathbf{R}_{\psi_i})^{-1} \begin{bmatrix} \mathbf{0_{1x3}} & 1 & 0 & 0 & \mathbf{0_{1x3}} \\ \mathbf{0_{1x3}} & 0 & 1 & 0 & \mathbf{0_{1x3}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^{\mathcal{W}}(k) \\ \mathbf{v}_i^{\mathcal{W}}(k) \\ \mathbf{f}_i^{\mathcal{W}}(k) \end{bmatrix} + \eta_{i2}^{\mathcal{W}}, \tag{3.16}$$

where $\mathbf{v}_{o_{ih}}^{\mathcal{W}}$ is the measured UAV horizontal velocity of the vehicle, and $\eta_{i2}^{w}$ is an uncorrelated zero mean white Gaussian signal modeling sensor uncertainty, with covariance matrix of $R_{\eta_{i2}^{\mathcal{W}}}$.

A Kalman Filter [58] is used with the described motion and measurement models, to estimate the vehicle states, $(\hat{\mathbf{x}}_i^{\mathcal{W}}(k), \hat{\mathbf{v}}_i^{\mathcal{W}}(k), \hat{\mathbf{f}}_i^{\mathcal{W}}(k))$. Matrix $Q_{\xi_i^{\mathcal{W}}}$ coefficients are according to the uncertainty on the propeller speed and the maximum perturbations experienced on the accelerometer and attitude measurements. Matrix $Q_{\gamma_i^{\mathcal{W}}}$ coefficients define how fast the disturbance vector, $\mathbf{f}_i^{\mathcal{W}}$, can change. Matrices $R_{\eta_{i1}^{\mathcal{W}}}$ and $R_\eta^{\mathcal{W}}$ coefficients are selected according to the observed sensor perturbations. Refer to Appendix A for details on the implementation of this filter.

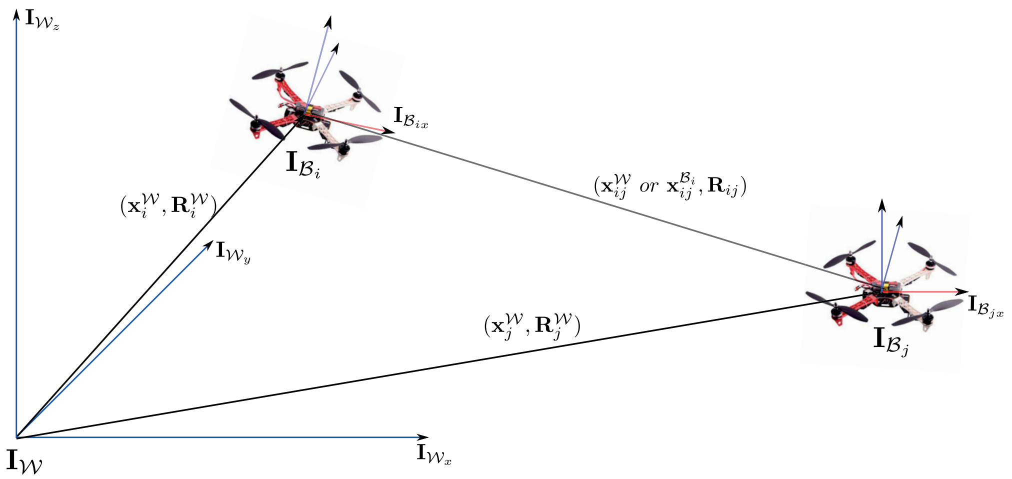

$\mathbf{I}_{\mathcal{W}_z}$
$\mathbf{I}_{\mathcal{B}_{ix}}$
$\mathbf{I}_{\mathcal{B}_i}$
$(\mathbf{x}_{ij}^{\mathcal{W}}$ *or* $\mathbf{x}_{ij}^{\mathcal{B}_i}, \mathbf{R}_{ij})$
$(\mathbf{x}_i^{\mathcal{W}}, \mathbf{R}_i^{\mathcal{W}})$
$\mathbf{I}_{\mathcal{W}_y}$
$\mathbf{I}_{\mathcal{B}_{jx}}$
$(\mathbf{x}_j^{\mathcal{W}}, \mathbf{R}_j^{\mathcal{W}})$
$\mathbf{I}_{\mathcal{B}_j}$
$\mathbf{I}_{\mathcal{W}}$
$\mathbf{I}_{\mathcal{W}_x}$

Figure 3.3: Definition of the environment absolute frame and the UAV body frame. The UAV absolute localization refers to its three dimensional position and attitude defined in the absolute frame. The inter-vehicle localization refers to the differences between the individual positions and attitudes of each UAV, either described in the absolute frame ($\mathbf{x}_{ij}^{\mathscr{W}}$) or in the local frame of each UAV ($\mathbf{x}_{ij}^{\mathscr{B}_i}$).

## 3.5 Inter-vehicle localization

When coordinating multiple UAVs, their inter-vehicle localization becomes important. For this it is relevant to recall the absolute pose of UAV $i$, as three dimensional position $\mathbf{x}_i^{\mathscr{W}}$ and attitude $\mathbf{R}_i^{\mathscr{W}}$ of $I_{\mathscr{B}_i}$ with respect to $I_{\mathscr{W}}$, as shown in Fig. 3.3. The absolute inter-vehicle position between UAVs $i$ and $j$ is obtained by computing the differences between the individual absolute positions:

$$\mathbf{x}_{ij}^{\mathscr{W}} = \mathbf{x}_j^{\mathscr{W}} - \mathbf{x}_i^{\mathscr{W}}.$$

It is also possible to define the relative inter-vehicle position by expressing $\mathbf{x}_{ij}^{\mathscr{W}}$ in UAV $i$'s local frame, $\mathbf{x}_{ij}^{\mathscr{B}_i} = (\mathbf{R}_i^{\mathscr{W}})^{-1}\mathbf{x}_{ij}^{\mathscr{W}}$. The inter-vehicle attitude between UAVs $i$ and $j$ ($\mathbf{R}_{ij}$ or $\mathbf{R}_j^{\mathscr{B}_i}$) is defined by the rotation between $I_{\mathscr{B}_i}$ and $I_{\mathscr{B}_j}$, and it is always a relative quantity. The three Euler angles that define this rotation are denoted as $\phi_{ij}$, $\theta_{ij}$, $\psi_{ij}$ and each one is responsible for rotations about a single axis of $\mathbf{I}_{\mathscr{B}_i}$ ($\mathbf{I}_{\mathscr{B}_{ix}}$, $\mathbf{I}_{\mathscr{B}_{iy}}$, and $\mathbf{I}_{\mathscr{B}_{iz}}$, respectively). Note that $\mathbf{R}_{ij} = \mathbf{R}_{\psi_{ij}}\mathbf{R}_{\theta_{ij}}\mathbf{R}_{\phi_{ij}}$. The computation of $\mathbf{R}_{ij}$ from the absolute attitudes of UAVs $i$ and $j$ is as follows:

$$\mathbf{R}_{ij} = (\mathbf{R}_i^{\mathscr{W}})^{-1}\mathbf{R}_j^{\mathscr{W}}.$$

Since in this work $\phi_i$ and $\theta_i$ angles are assumed to be small, $\mathbf{R}_{ij}$ can sometimes be represented only by $\psi_{ij}$ (or the matrix $\mathbf{R}_{\psi_{ij}}$). Under these conditions $\psi_{ij}$ can be computed as $\psi_j - \psi_i$. The set ($\mathbf{x}_{ij}^{\mathscr{B}_i}$, $\mathbf{R}_{ij}$), depicted in Fig. 3.3, defines the relative inter-vehicle pose of UAV $j$ in UAV $i$ local frame which consists of the three dimensional position and attitude of $I_{\mathscr{B}_j}$ expressed in $I_{\mathscr{B}_i}$. The absolute inter-vehicle localization can be acquired by having the UAVs sharing between each other (through a communication system) their absolute positions acquired using the estimation algorithm discussed in Section 3.4. The relative inter-vehicle localization can be acquired by transforming the absolute inter-vehicle localization to the UAV body frame. However, this information can also be acquired by measuring it directly using onboard relative positioning systems, as it will be discussed in further sections.

## 3.6 Formation control

A formation is employed whenever the target application requires a multi-agent system to achieve a specific geometric configuration. Formation control algorithms are responsible to control the desired inter-agent geometric constraints that are required to achieve the desired geometric configuration. This work implements these algorithms for UAVs using a graph-based approach similar to the ones in [27, 55, 77], discussed in Section 2.2.2. These approaches define a group of $N$ UAVs that are required to maintain at any time a certain geometric configuration, defined by a full network graph $\mathscr{G} := (\mathscr{V}, \mathscr{E})$. In this graph, $\mathscr{V}$ is the set of $N$ nodes, each one representing a UAV, described by its absolute pose and velocity in the environment ($\mathbf{x}_i^{\mathscr{W}}$, $\mathbf{R}_i^{\mathscr{W}}$, $\mathbf{v}_i^{\mathscr{W}}$). As discussed in Section 3.3, each quadrotor is considered to have an onboard auto-pilot and an IMU sensor, which allows for the control of the UAV thrust together with the estimation and control of its attitude. Only the control of ($\phi_i$,$\theta_i$) together with the thrust control are needed to move the UAV on the three Cartesian axes, leaving the control of $\psi_i$ free to be used for independent purposes (as it will be discussed later). Therefore, in practice, quadrotor control can be seen as a combination of 3D position control, with a double integrator model, and $\psi_i$ controller with a single integrator dynamics:

$$\ddot{\mathbf{x}}_i^{\mathscr{W}} = \mathbf{a}_{i_d}^{\mathscr{W}}, \qquad \dot{\psi}_i = \omega_{i_z,d}, \tag{3.17}$$

where $\mathbf{a}_{i,d}^{\mathscr{W}}$ and $\omega_{i_z,d}$ are the desired control inputs for the UAV 3D position and the $z$ axis angular velocity respectively, given to the auto-pilots presented in Section 3.3 and depicted in the block diagram of Fig. 3.2. Since the previous control is assumed to operate with the quadrotor in quasi hovering conditions, the UAV $\phi_i$ and $\theta_i$ angles can be assumed to be small. This allows the representation of the quadrotor attitude $\mathbf{R}_i^{\mathscr{W}}$ simply by $\psi_i$.

The set of all $N(N-1)/2$ edges of $\mathscr{G}$ is represented by $\mathscr{E}$, where each edge represents the inter-vehicle localization information between two UAVs, namely pose ($\mathbf{x}_{ij}^{\mathscr{W}}$,$\mathbf{R}_{ij}$) and, as derived quantity, velocity $\mathbf{v}_{ij}^{\mathscr{W}}$. Note that since $\phi_i$ and $\theta_i$ angles are assumed to be small, $\mathbf{R}_{ij}$ can be represented by the difference between the angles around the $z$ axes of each UAV $\psi_{ij} = \psi_j - \psi_i$ (or the matrix $\mathbf{R}_{\psi_{ij}}$). To control $\mathscr{G}$, the UAVs need to collect information represented by a subset of the previous edges. The information that can be obtained at any time is described by a sensing graph, defined by $\mathscr{G}_S := (\mathscr{V}, \mathscr{E}_S)$. The edge set $\mathscr{E}_S$ represents the edges of $\mathscr{G}$ for which the inter-vehicle localization information between the respective UAVs can be acquired. The edge $\mathscr{E}_{S_{ij}}$ is directional, pointing from UAV $i$ to UAV $j$ if only UAV $i$ can acquire this information. If both UAVs can acquire this information, the edge $\mathscr{E}_{S_{ij}}$ is bidirectional, as shown in Fig. 3.4.

A target geometric configuration is defined by a set of desired inter-vehicle poses ($\mathbf{x}_{ij,d}^{\mathscr{W}}$,$\mathbf{R}_{\psi_{ij,d}}$), and

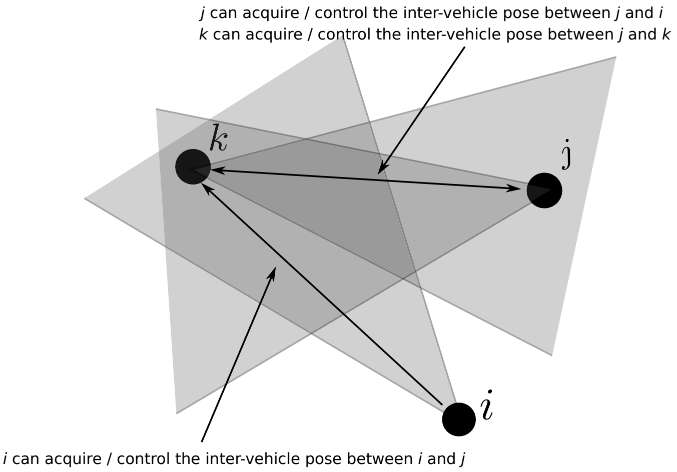

*j* can acquire / control the inter-vehicle pose between *j* and *i*
*k* can acquire / control the inter-vehicle pose between *j* and *k*
*k*
*j*
*i*
*i* can acquire / control the inter-vehicle pose between *i* and *j*

Figure 3.4: Definition of the sensing and control graphs used in the considered formation control algorithms. Each UAV represents a node in the graph, and each edge an (unidirectional or bidirectional) inter-vehicle interaction.

a zero inter-vehicle velocity. To control the geometric configuration, a formation control graph $\mathscr{G}_F := (\mathscr{V}, \mathscr{E}_F)$ is defined, where each edge in $\mathscr{E}_F$ corresponds to a controlled inter-vehicle position. The edge $\mathscr{E}_{F_{ij}}$ is directional, pointing from UAV $i$ to UAV $j$ if only UAV $i$ is controlling the respective inter-vehicle position. If both UAVs are controlling the inter-vehicle position, the edge $\mathscr{E}_{F_{ij}}$ is bidirectional, as shown in Fig. 3.4. The desired geometric configuration is controllable if the following two conditions are verified: $\mathscr{G}_F$ is rigid, so that it has an unique solution (the reader is referred to [78] for more details about graph rigidity); $\mathscr{G}_S$ must contain enough information to enable the control of all edges in $\mathscr{G}_F$. The graphs $\mathscr{G}_S$ and $\mathscr{G}_F$ can be fixed and defined a priori, or could be adapted for time varying graph configurations. The definition of sensing and control graphs is similar to the work presented in [26]. Note that UAV $j$ is considered to be a neighbor of UAV $i$ if it belongs to any edge in $\mathscr{E}_S$ or $\mathscr{E}_F$ that also contains UAV $i$.

As discussed in Section 2.2.1, the graph-based approaches define the desired geometric configuration as a set of inter-agent position biases to be satisfied, described in $\mathscr{E}_F$. The consensus equation is then used to stabilize the multi-agent system while satisfying the desired geometric configuration. The consensus equation can be easily defined when using the absolute inter-vehicle localization. Its definition for each UAV $i$ is as follows:

$$\mathbf{a}_{i,d}^{\mathscr{W}} = k_p \sum_{j=1}^{N} \mathbf{L}_{ij}(\mathbf{x}_{ij}^{\mathscr{W}} - \mathbf{x}_{ij,d}^{\mathscr{W}}) + k_v \sum_{j=1}^{N} \mathbf{L}_{ij}\mathbf{v}_{ij}^{\mathscr{W}} \tag{3.18}$$

where $\mathbf{a}_{i,d}^{\mathscr{W}} = (\mathbf{a}_{i_x,d}^{\mathscr{W}}, \mathbf{a}_{i_y,d}^{\mathscr{W}}, \mathbf{a}_{i_z,d}^{\mathscr{W}})$ is the desired acceleration, $\mathbf{x}_{ij}^{\mathscr{W}}$ and $\mathbf{x}_{ij,d}^{\mathscr{W}}$ are respectively the current and desired inter-vehicle positions, and $\mathbf{v}_{ij}^{\mathscr{W}}$ is the current inter-vehicle velocity. All the previous quantities are expressed in the world frame. The scalars $k_p$ and $k_v$ are gain parameters for the position and the velocity components of the controller. The scalar $\mathbf{L}_{ij}$ is a control weight for the inter-vehicle localization between UAVs $i$ and $j$, describing how strongly the UAVs control it. The weighting information can be elegantly represented using a Laplacian matrix, $\mathbf{L}$, where the element $\mathbf{L}_{ij}$ defines the control weight between neighbors $i$ and $j$. This matrix is positive definite, the sum of the elements of each line add up to zero, and $\mathbf{L}_{ii} = -\sum_{j \neq i} \mathbf{L}_{ij}$. If $\mathbf{L}_{ij} = 0$, the inter-vehicle position between UAV $i$ and $j$ is not directly controlled. In this work, $\mathbf{L}$ is constant, since $\mathscr{G}_F$ is assumed constant. In the previous control law presented in Eq. (3.18), the position component aims at achieving the formation requirements, while the velocity component stabilizes the second order dynamic system. In [41, 87] the previous control law implemented at each UAV $i$ is proven to stabilize the multi-agent system to the desired geometric configuration.

A separate controller using the $z$ axis angle control input $\omega_{i_z,d}$ defined in Eq. (3.17) can be considered to control the inter-vehicle attitude $\mathbf{R}_{\psi_{ij}}$ to the desired values. This provides a $z$ axis angle controller for each UAV. Angles about the UAV's $x$ and $y$ axes are controlled by the position controller to move the UAV horizontally, as described in Section 3.3, and are kept close to zero values. Note that the attitude controller can be applied independently from the previously described position controller. An implementation of this controller is presented in Section 4.2.2.

# Chapter 4

# Quadrotor Relative State Estimation and Formation Control

This work tackles the problem of formation control of multiple UAVs using exclusively their onboard resources. In this chapter, the approach used to solve this problem is presented, making use of the concepts that were introduced in the previous chapters. In Section 4.1 the general assumptions made in this work, based on the target UAVs and environments that were considered. are presented. Then, Section 4.2 proceeds with the presentation of the overall algorithm architecture on each UAV. Section 4.2.1 provides a general description of the estimation algorithms on board each UAV, enabling it to acquire the localization information required for the control algorithms. Finally, this chapter concludes in Section 4.2.2 with the general description of the formation control algorithms that were considered on board the UAVs.

## 4.1 Assumptions

Since small-scale UAVs are considered in this work, the algorithm framework on each UAV was developed in order to allow its implementation on UAVs with a small amount of sensory and computational capabilities. The following assumptions are then considered when developing this approach:

- The UAVs are assumed to be capable only of acquiring UAV height and horizontal velocity measurements in the UAV body frame, using height and OF sensors.
- The $\psi_i$ angle measurements are assumed to be too noisy to be considered. However, control of the UAV about the $z$ axis angular velocity $\omega_{i_z,d}$ can still be provided, as discussed in Section 3.3.
- Although no further absolute localization capabilities are assumed on board each UAV, it is assumed that it possesses an onboard relative positioning system capable of acquiring relative inter-vehicle localization information from its neighbors.
- Given the fast dynamics of the UAVs, high control rates are necessary to stabilize the resulting systems. In these cases, approaches become sensitive to packet loss or latency in communication

links. This is particularly important for short range inter-agent interactions, requiring even faster reaction times. These ranges are considered in this work since they are common in small-scale UAVs and indoor environments. Therefore, there is an interest in keeping communication to a minimum. This work assumes that there is no communication between UAVs necessary to perform the formation control algorithms. A communication link can be establish between each robot and a centralized system (e.g., an external system) for high level planning, such as moving the formation in the environment. However, the possibility of adding inter-vehicle communication to improve the quality of the interactions is allowed.

- Additionally, since the considered UAVs are quadrotors, they are assumed to be most of the time in the near-hovering condition. Small inclinations (small $\phi_i$ and $\theta_i$ values) are provided in order to move the UAV horizontally. This assumption is realistic since usually the quadrotors do not need to tilt more than a few degrees to achieve good reaction times. This assumption was already considered in Section 3.6. Since $\phi_i$ and $\theta_i$ values are always considered small, the attitude of UAV $i$ ($\mathbf{R}_i^{\mathscr{W}}$) can be simply described by $\psi_i$ (or matrix $\mathbf{R}_{\psi_i}$). Additionally, the inter-vehicle attitude $\mathbf{R}_{ij}$ can be represented in this work by $\psi_{ij} = \psi_j - \psi_i$ (or matrix $\mathbf{R}_{\psi_{ij}}$).

The first two assumptions require the change of the estimation and control algorithms to another frame different form the environment frame (since the later is not observable from the UAV). Note that, as discussed in Section 3.3, the $\phi_i$ and $\theta_i$ angles of the UAV in the environment can still be measured. Therefore, the UAV is able to observe the direction of the horizontal plane and the vertical axis of the environment. Based on this fact, it is possible to define a new (local) frame of reference for the quadrotors, denoted as $\mathbf{I}_{\mathscr{L}_i}$, such as the one depicted in Fig. 4.1. Its origin is the same as the origin of the UAV frame, the $z$ axis is aligned with the $z$ axis of the environment frame $I_{\mathscr{W}}$ (vertical), and its $xy$ plane parallel to the ground. In the remainder of this work, this frame will be named as *flying frame*; it has an attitude of $(\phi, \theta, \psi) = (0, 0, \psi_i)$ with respect to the environment frame $I_{\mathscr{W}}$. With this frame all quadrotors have the same local vertical axis. Additionally, it is possible to eliminate perturbations on the measured ($\mathbf{x}_{ij}^{\mathscr{B}_i}$, $\mathbf{R}_{ij}$) caused by UAV $i$'s $\phi_i$ and $\theta_i$ inclinations necessary for horizontal movements.

The control and estimation algorithms on board each UAV are only related to quantities expressed in the flying frame. Therefore, the previous quadrotor control described by Eq. 3.8 has to be rewritten in order to consider the control of desired accelerations in this frame, $\mathbf{a}_{i,d}^{\mathscr{L}_i}$. The transformation between $\mathbf{a}_{i,d}^{\mathscr{L}_i}$ and the acceleration expressed in the environment frame is $\mathbf{a}_{i,d}^{\mathscr{L}_i} = (\mathbf{R}_{\psi_i})^{-1}\mathbf{a}_{i,d}^{\mathscr{W}}$. The $\phi_i$ and $\theta_i$ rotations are discarded here since the UAV is assumed to be most of the time in the near-hovering condition. Under these conditions, Eq. 3.8 can be adapted for the control in the flying frame as follows:

$$\begin{cases} \frac{F_{i,d}}{m} = g + a_{i_z,d}^{\mathscr{L}_i} - f_{i_z}^{\mathscr{L}_i} \\ \begin{bmatrix} \phi_{i,d} \\ \theta_{i,d} \end{bmatrix} = \left(\frac{F_{i,d}}{m}\right)^{-1} \begin{bmatrix} a_{i_x,d}^{\mathscr{L}_i} \\ a_{i_y,d}^{\mathscr{L}_i} \end{bmatrix} - \begin{bmatrix} f_{i_x}^{\mathscr{L}_i} \\ f_{i_y}^{\mathscr{L}_i} \end{bmatrix} \end{cases} . \tag{4.1}$$

Additionally, given that only height and OF sensors are assumed available on board the UAVs, only their height and horizontal velocity can be measured. These quantities can be expressed in the UAV flying frame, and their dynamics in this frame can be taken from the discretized motion model described in Eq. (3.14), as follows:

$$\begin{bmatrix} z_i^{\mathcal{L}_i}(k+1)\\ \mathbf{v}_i^{\mathcal{L}_i}(k+1)\\ \mathbf{f}_i^{\mathcal{L}_i}(k+1)\end{bmatrix} = \mathbf{A}\begin{bmatrix} z_i^{\mathcal{L}_i}(k)\\ \mathbf{v}_i^{\mathcal{L}_i}(k)\\ \mathbf{f}_i^{\mathcal{L}_i}(k)\end{bmatrix} + \mathbf{B}\left(F_i(k+1)\mathbf{R}_i^{\mathcal{L}_i}\mathbf{I}_{\mathcal{L}_{iz}} - g\mathbf{I}_{\mathcal{L}_{iz}}\right) + \mathbf{W}\begin{bmatrix}\xi_i^{\mathcal{L}_i}(k+1)\\ \gamma_i^{\mathcal{L}_i}(k+1)\end{bmatrix} \tag{4.2}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0\ \ 0\ \ \Delta t & 0\ \ 0\ \ \frac{\Delta t^2}{2}\\ \mathbf{0_{3x1}} & \mathbf{I_{3x3}} & \Delta t\mathbf{I_{3x3}}\\ \mathbf{0_{3x1}} & \mathbf{0_{3x3}} & \mathbf{I_{3x3}}\end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} 0\ \ 0\ \ \frac{\Delta t^2}{2}\\ \Delta t\mathbf{I_{3x3}}\\ \mathbf{0_{3x3}}\end{bmatrix}, \qquad \mathbf{W} = \begin{bmatrix} \frac{\Delta t^2}{2}\mathbf{I_{1x3}} & \mathbf{0_{1x3}}\\ \Delta t\mathbf{I_{3x3}} & \mathbf{0_{3x3}}\\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}}\end{bmatrix}.$$

Note that $\mathbf{R}_i^{\mathcal{L}_i}$ is described by the Euler angles $(\phi,\theta,\psi) = (\phi_i,\theta_i,0)$ and $\mathbf{I}_{\mathcal{L}_{iz}} = \mathbf{I}_{\mathcal{W}_z}$. Additionally, note that $\mathbf{v}_i^{\mathcal{L}_i}$, $\mathbf{f}_i^{\mathcal{L}_i}$, $\xi_i^{\mathcal{L}_i}$ and $\gamma_i^{\mathcal{L}_i}$ are now all expressed in the flying frame. The respective measurement models for these quantities can be reduced from the models in Eqs. (3.15) and (3.16) and transformed to the flying frame as follows:

$$z_{o_i}^{\mathcal{L}_i}(k) = \begin{bmatrix} 0 & 0 & 1 & \mathbf{0_{1x3}} & \mathbf{0_{1x3}}\end{bmatrix}\begin{bmatrix} z_i^{\mathcal{L}_i}(k)\\ \mathbf{v}_i^{\mathcal{L}_i}(k)\\ \mathbf{f}_i^{\mathcal{L}_i}(k)\end{bmatrix} + \eta_{i1}^{\mathcal{L}_i}, \qquad \mathbf{v}_{o_{ih}}^{\mathcal{L}_i}(k) = \begin{bmatrix} \mathbf{0_{1x3}} & 1 & 0 & 0 & \mathbf{0_{1x3}}\\ \mathbf{0_{1x3}} & 0 & 1 & 0 & \mathbf{0_{1x3}}\end{bmatrix}\begin{bmatrix} \mathbf{x}_i^{\mathcal{L}_i}(k)\\ \mathbf{v}_i^{\mathcal{L}_i}(k)\\ \mathbf{f}_i^{\mathcal{L}_i}(k)\end{bmatrix} + \eta_{i2}^{\mathcal{L}_i} \tag{4.3}$$

where $z_{o_i}^{\mathcal{L}_i}$ and $\mathbf{v}_{o_{ih}}^{\mathcal{L}_i}$ are, respectively, the UAV height and horizontal velocity expressed in the flying frame, and $\eta_{i1}^{\mathcal{L}_i}$ and $\eta_{i2}^{\mathcal{L}_i}$ are uncorrelated zero mean white Gaussian signals modeling sensor uncertainties, with covariance matrices of $R_{\eta_{i1}^{\mathcal{L}_i}}$ and $R_{\eta_{i2}^{\mathcal{L}_i}}$. Eqs. (4.2) and (4.3) describe the UAV motion and measurement models used for the UAV self-localization in this work.

The onboard relative positioning sensor of the UAV is considered to have a certain FOV, centered in the horizontal $xy$ plane of the flying frame, considered to be the sensor direction, $\mathbf{r}_{s_i}^{\mathcal{L}_i}$. The FOV is represented by its horizontal, $\theta_h$, and vertical, $\theta_v$, components, defined with respect to the previous described horizontal plane, and a vertical plane formed by $\mathbf{r}_{s_i}^{\mathcal{L}}$ and the $z$ axis of the flying frame. For the previous FOV assumption to be accurate the UAV has to move with small inclinations (small $\phi_i$ and $\theta_i$ values). Additionally, the deployed sensors have to be close to the UAV center. These conditions are usually met for quadrotors. In case the UAV and the sensors are tilted, the measurements acquired in the sensor frame can be transformed into the flying frame, using the $\phi_i$ and $\theta_i$ values acquired from the onboard IMU sensor. The relative positioning sensors are assumed to measure either part or the entire relative position $\mathbf{x}_{ij}^{\mathcal{L}_i}$ and attitude $\mathbf{R}_{ij}$ of UAV $j$.

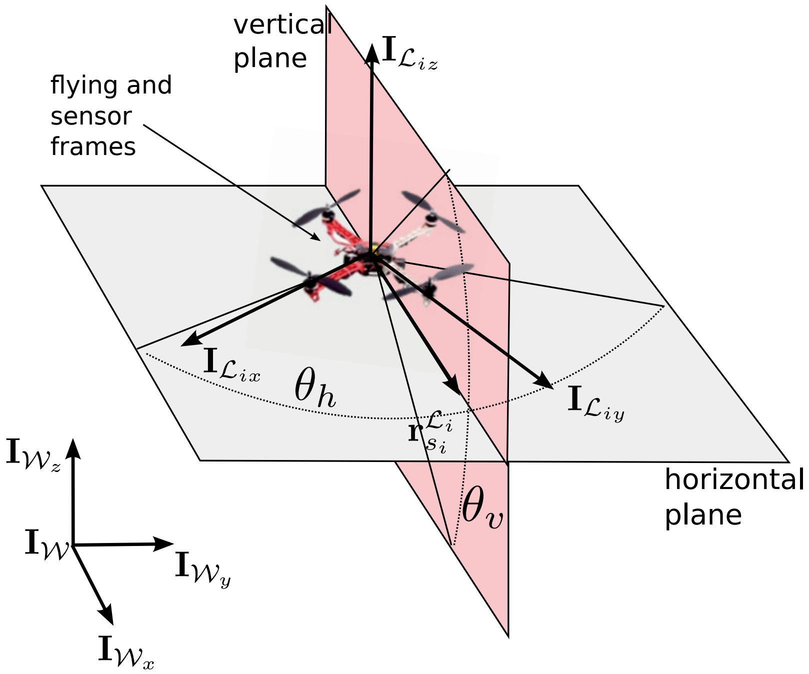

vertical plane
$\mathbf{I}_{\mathcal{L}_{iz}}$
flying and sensor frames
$\mathbf{I}_{\mathcal{L}_{ix}}$
$\theta_h$
$\mathbf{r}_{s_i}^{\mathcal{L}_i}$
$\mathbf{I}_{\mathcal{L}_{iy}}$
horizontal plane
$\theta_v$
$\mathbf{I}_{\mathcal{W}_z}$
$\mathbf{I}_{\mathcal{W}}$
$\mathbf{I}_{\mathcal{W}_y}$
$\mathbf{I}_{\mathcal{W}_x}$

Figure 4.1: Definition of the quadrotor local frame, named as flying frame. The absolute 3D position of this frame coincides with the absolute 3D position of the UAV. The attitude of this frame with respect to the environment frame is described by the three Euler angles $(\phi, \theta, \psi) = (0, 0, \psi_i)$. The relative inter-vehicle localization measurements acquired by the UAV onboard relative positioning sensor are defined in the UAV's flying frame.

## 4.2 Overall architecture

The overall algorithm architecture onboard each UAV is depicted in Fig. 4.2. The *Self Pose Estimator* is responsible to provide UAV *i* with self localization estimations using the tools presented in Section 3.4. As discussed in Section 4.1, the small-scale UAVs considered in this work are assumed to only be capable of acquiring UAV height and horizontal velocity measurements in the UAV flying frame. This makes the UAV only capable of having height and velocity estimations ($\hat{z}_i^{\mathcal{L}_i}$,$\hat{\mathbf{v}}_i^{\mathcal{L}_i}$) for self localization, using the models described by Eqs. (4.2) and (4.3).

The *Neighbor Relative Pose Estimator* is responsible to provide the UAV *i* with relative inter-vehicle localization measurements of its neighbors using the relative positioning system on board the UAV. These measurements are then given as inputs to the *Formation Controller.* The estimation algorithms used in this task are described in more detail in Section 4.2.1.

The *Formation Controller* implements the formation control algorithm on board each UAV *i*, based on the tools presented in Section 3.6. The outputs from this controller consist of a desired acceleration expressed in the UAV *i* flying frame, and an additional angular velocity about the *z* axis of the flying frame. The importance of controlling the the angular velocity about the *z* axis of the flying frame is

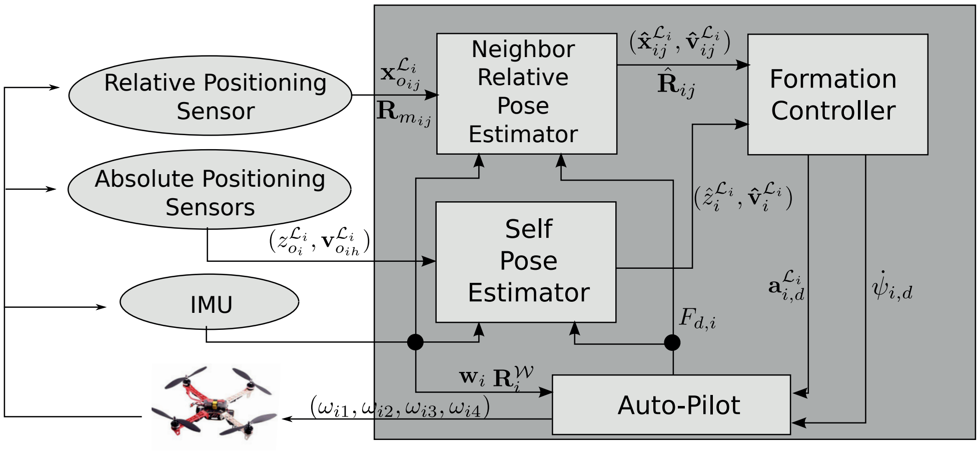

Relative Positioning Sensor
Absolute Positioning Sensors
IMU
$\mathbf{x}_{o_{ij}}^{\mathcal{L}_i}$
$\mathbf{R}_{m_{ij}}$
Neighbor Relative Pose Estimator
$(\hat{\mathbf{x}}_{ij}^{\mathcal{L}_i}, \hat{\mathbf{v}}_{ij}^{\mathcal{L}_i})$
$\hat{\mathbf{R}}_{ij}$
Formation Controller
$(z_{o_i}^{\mathcal{L}_i}, \mathbf{v}_{o_{ih}}^{\mathcal{L}_i})$
Self Pose Estimator
$(\hat{z}_i^{\mathcal{L}_i}, \hat{\mathbf{v}}_i^{\mathcal{L}_i})$
$\mathbf{a}_{i,d}^{\mathcal{L}_i}$
$\dot{\psi}_{i,d}$
$F_{d,i}$
$\mathbf{w}_i\ \mathbf{R}_i^{\mathcal{W}}$
$(\omega_{i1}, \omega_{i2}, \omega_{i3}, \omega_{i4})$
Auto-Pilot

Figure 4.2: UAV onboard formation control and estimation functional architecture. The onboard positioning systems are composed of the onboard sensors and respective estimators. They provide the necessary self-vehicle and inter-vehicle localization measurements required by the onboard formation controller. The formation controller provides desired UAV acceleration and rotations in its flying frame, which are carried out by the onboard auto-pilot.

further discussed Section 5.2. These inputs are transformed to the desired UAV thrust, $\phi_{i,d}$, $\theta_{i,d}$, and $\psi_{i,d}$ control inputs using an algorithm described by Eq. (4.1), and then given to the UAV onboard auto-pilot. The *Auto-Pilot* controls the propeller speeds in order to achieve the desired UAV control inputs, as discussed in Section. 3.3. The control algorithms used in this task are described in more detail in Section 4.2.2.

The *Formation Controller* is also responsible to steer the formation in the environment. This task is achieved by using an external system that is capable of communicating with the UAVs high level motions commands in the environment, given that the UAVs are considered not to have absolute localization capabilities. The external system can be either a teleoperator or an external autonomous system. The functional architecture is depicted in Fig. 4.3. In this work, two approaches are considered to move the formation.

The first approach consists of a leader-follower approach with a team-member chosen as a physical leader, previously described in Section 2.2.2. This approach is included by default in the control algorithms presented in Section 4.2.2. In this approach, only the leader's local frame is necessary for the movement. While the leader moves, the formation controller on board each follower will keep the formation in the desired configuration, triggering the UAV movement towards the desired leader direction. However, as discussed in Section 2.2.2, this approach will generate motion delay between the leader and the followers, which tends to increase as the number of team-members increases. This delay generates a distortion in the desired geometric configuration, defined by the maximum number sensing hops between two agents.

A second approach is a virtual structure approach described in Section 2.2.2, which is used to min-

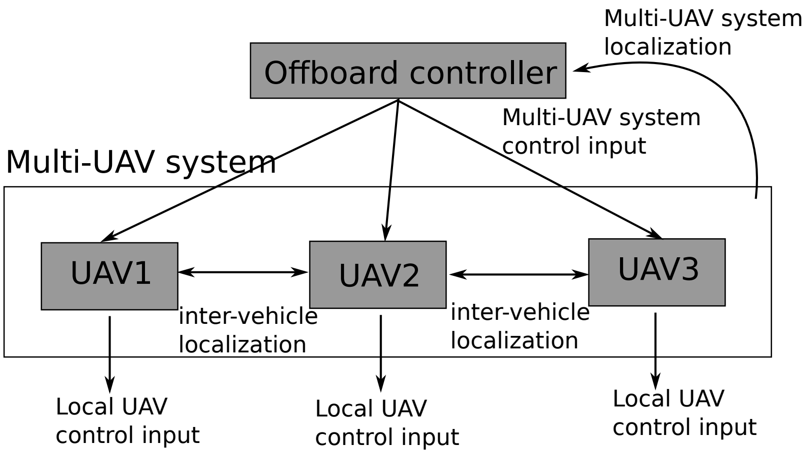


Figure 4.3: Conceptual architecture for the offboard multi-UAV system controller. The same multi-UAV system motion control input is given through communication to all UAVs of the system. The UAVs translate this input into motion in their individual local frames, according to their currently inter-vehicle localization measurements. The offboard controller generates the multi-UAV system motion control inputs in order to steer the system towards a desired goal in the environment.

imize the previous geometric configuration distortions. The external system acquires information about the formation center localization in the environment, which defines the formation position in the environment. Afterwards, the external system issues desired formation motion commands to all team-members of the multi-UAV system simultaneously. Since all team-members receive the movement commands at the same time, the previous geometric configuration distortion discussed in the leader-follower approach can be substantially reduced. The formation controller that maintains the desired inter-vehicle geometric constraints is still controlled on each UAV, in parallel to the steering controller. Therefore, in cases where the communication between the external system and only some UAVs fails, their onboard formation controllers will still make the UAVs move in the desired direction using their onboard relative positioning sensors as sensory feedback; eventually, if the communication fails for all UAVs, this approach would degenerate to the previous leader-follower approach.

As discussed in Chapter 3, accurate and reliable offboard localization structures, such as MCSs, are expensive, difficult to mount, and impractical in many scenarios due to environment characteristics or lack of time or resources to set up the required infrastructure. Additionally, when using a teleoperator as the external system, individual UAV localization in the environment is not possible, and only a rough estimate of the formation center can be acquired. In order to accommodate the previous limitations and to simplify the deployment of the external system, only an approximated estimation of the formation center in the environment is assumed available to the external system. No individual UAV localization in the environment is available at the external system, or at any of the UAVs. This translates into problems when moving the formation with a virtual structure approach, because finding a common reference frame observable by the UAVs on which the virtual structure can be defined becomes cumbersome, as discussed in Section 2.2.2. Given that no communication between UAVs exist and that the individual poses of each UAV are unknown by the external system and by the UAVs, converting the desired virtual structure motion commands to the control inputs on the local frames of each UAV becomes impossible using the current literature. In this work, a novel algorithm will allow this conversion by making use of the formation desired geometry configuration. The details of this algorithm are presented in Section 5.2.2.

### 4.2.1 Neighbor relative pose estimator

The *Neighbor Relative Pose Estimator* is responsible to provide the UAV $i$ with relative inter-vehicle localization measurements of its neighbors using the relative positioning system on board the UAV. This thesis focuses on the acquisition of these measurements. As discussed in Chapter 1, the relative positioning systems used in this work are camera-based and infrared-based. The specific hardware implementation and the adaptation of the previously discussed relative inter-vehicle localization algorithms to each positioning system are described in detail in Section 5.1.

Provided the measurements of the relative positioning system onboard UAV $i$, relative inter-vehicle localization estimates for each observed UAV $j$ can be acquired using an estimator based on the Kalman Filter framework, as done before for the UAV self-localization. The states to be estimated are the relative position $\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}$, velocity $\hat{\mathbf{v}}_{ij}^{\mathscr{L}_i}$ and attitude $\hat{\mathbf{R}}_{ij}$ of each observed UAV $j$, defined in the UAV $i$'s flying frame. The relative attitude is also defined by the relative $\phi_{ij}$, $\theta_{ij}$ and $\psi_{ij}$ Euler angles. The estimation algorithm tracks each observed UAV with an independent estimation process.

An estimator for $\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}$, $\hat{\mathbf{v}}_{ij}^{\mathscr{L}_i}$ and $\hat{\psi}_{ij}$ is implemented as follows. The states are propagated using a motion model based on a constant speed model described by the egomotion of UAV $i$. This model is discretized, similarly to what was considered in Section 3.4. Between these time intervals, the UAV $i$ egomotion is computed using the desired vertical thrust commands currently issued to the UAV autopilots combined with the current attitude measurements provided by the onboard IMU sensor, $\phi_i$ and $\theta_i$, to predict the linear acceleration felt in the UAV flying frame, $\mathbf{a}_i^{\mathscr{L}_i} = (a_{i_x}^{\mathscr{L}_i}, a_{i_y}^{\mathscr{L}_i}, a_{i_z}^{\mathscr{L}_i})^T$, at that time interval. This is done using again the translational part of the quadrotor dynamics defined in Eq. (3.6), with the difference that the generated acceleration has an inverse impact on the estimated relative position. For example, if the UAV moves forward towards a team member, their inter-vehicle range will decrease. Additionally, one needs to also take into account the UAV self-rotations about the $z$ axis that are not compensated in the flying frame. For example, although when one quadroror rotates around its $z$ axis the absolute inter-vehicle positions between it and another team-member remain the same, the relative bearing of the team mate changes in the rotating UAV flying frame. With this information, it is possible to define the motion model for the observed UAV $j$ in UAV $i$'s flying frame:

$$\begin{aligned}\begin{bmatrix}\mathbf{x}_{ij}^{\mathscr{L}_i}(k+1)\\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k+1)\\ \psi_{ij}^{\mathscr{L}_i}(k+1)\end{bmatrix} &= \begin{bmatrix}\mathbf{I_{3x3}} & \Delta t\mathbf{I_{3x3}} & \mathbf{0_{3x1}}\\ \mathbf{0_{3x3}} & \mathbf{I_{3x3}} & \mathbf{0_{3x1}}\\ \mathbf{0_{1x3}} & \mathbf{0_{1x3}} & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_{ij}^{\mathscr{L}_i}(k)\\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k)\\ \psi_{ij}^{\mathscr{L}_i}(k)\end{bmatrix} - \begin{bmatrix}\frac{\Delta t^2}{2}\mathbf{I_{3x3}}\\ \Delta t\mathbf{I_{3x3}}\\ \mathbf{0_{1x3}}\end{bmatrix}\mathbf{a}_i^{\mathscr{L}_i}(k)\\ &+ \begin{bmatrix}\mathbf{x}_{ij}^{\mathscr{L}_i}(k)\times\Omega_i^{\mathscr{L}_i}(k)\\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k)\times\Omega_i^{\mathscr{L}_i}(k)\\ \Omega_i^{\mathscr{L}_i}\Delta t\end{bmatrix} + \xi_{ij}^{\mathscr{L}_i}(k)\end{aligned}, \tag{4.4}$$

where $\Omega_i^{\mathscr{L}_i}$ is the vector $(0,0,\dot{\psi}_i)^T$, which provides a rotation about the $\mathscr{L}_i$'s $z$ axis of $\mathbf{x}_j^{\mathscr{L}_i}$ and $\hat{\mathbf{v}}_j^{\mathscr{L}_i}$ for a non-zero $\dot{\psi}_i$. This term uses directly the angular velocity around the $z$ axis of the UAV $\omega_{i_z}$, which makes the model only valid when UAV $i$ is flying with small $\phi_i$ and $\theta_i$ values (otherwise a transformation would have to be applied in order to convert the angular velocity to the true angle rate about the $\mathscr{L}_i$'s $z$ axis). But in this work it is assumed that those conditions are met. Note that $\mathbf{a}_i^{\mathscr{L}_i}$ refers to the acceleration felt in the flying frame. The last term, $\xi_{ij}^{\mathscr{L}_i}$, is the movement noise simply considered as a zero mean Gaussian distribution with a covariance profile that tries to encapsulate the noise of all the previous terms combined. This last term is also used to adapt to the situations where the observed UAV is moving by its own in an unknown pattern.

The measurements acquired by the relative positioning system on UAV $i$ are used to update state estimates. The measurement models used for the estimator are as follows:

$$\begin{bmatrix}\mathbf{x}_{o_{ij}}^{\mathscr{L}_i}(k)\\ \psi_{o_{ij}}(k)\end{bmatrix} = \begin{bmatrix}\mathbf{I_{3x3}} & \mathbf{0_{3x3}} & 0\\ \mathbf{0_{1x3}} & \mathbf{0_{1x3}} & 1\end{bmatrix}\begin{bmatrix}\mathbf{x}_{ij}^{\mathscr{L}_i}(k)\\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k)\\ \psi_{ij}(k)\end{bmatrix} + \eta_{ij}^{\mathscr{L}_i}, \tag{4.5}$$

where $\mathbf{x}_{o_{ij}}^{\mathscr{L}_i}$ and $\psi_{o_{ij}}$ are, respectively, the relative inter-vehicle position and attitude between UAVs $i$ and $j$ measured by the relative positioning system on UAV $i$, and $\eta_{ij}^{\mathscr{L}_i} = (\eta_{ij_x}^{\mathscr{L}_i}, \eta_{ij_y}^{\mathscr{L}_i}, \eta_{ij_z}^{\mathscr{L}_i}, \eta_{\psi_{ij}}^{\mathscr{L}_i})^T$ is the term defining the uncertainty of the relative positioning sensor for each observed UAV $j$, model by a zero mean Gaussian distribution. The above models can be used to implement a Kalman Filter that acquires the estimates of $\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}$, $\hat{\mathbf{v}}_{ij}^{\mathscr{L}_i}$ and $\hat{\psi}_{ij}$. Refer to Appendix A for details on the implementation of this filter.

Estimates for $\phi_{ij}$ and $\theta_{ij}$ are directly taken from the relative positioning sensor measurements without any additional filtering, as they are considered accurate enough:

$$\begin{bmatrix}\hat{\phi}_{ij} & \hat{\theta}_{ij}\end{bmatrix} = \begin{bmatrix}\phi_{o_{ij}} & \theta_{o_{ij}}\end{bmatrix}. \tag{4.6}$$

This estimator allows UAV $i$ to acquire the relative inter-vehicle localization of its neighbors, using a version of the algorithms described by Eqs. (4.4), (4.5) and (4.6). Finally, it is worth noting that in this work the relative positioning system is able to track the ID of each observed UAV.

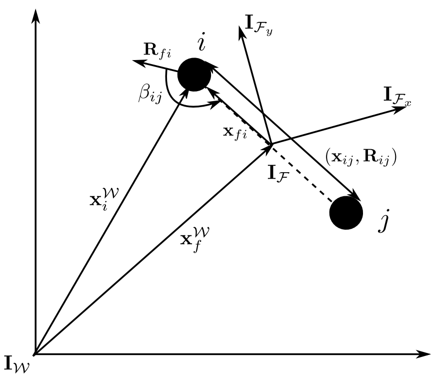

$\mathbf{R}_{fi}$
$i$
$\mathbf{I}_{\mathcal{F}_y}$
$\beta_{ij}$
$\mathbf{I}_{\mathcal{F}_x}$
$\mathbf{x}_{fi}$
$(\mathbf{x}_{ij}, \mathbf{R}_{ij})$
$\mathbf{I}_{\mathcal{F}}$
$\mathbf{x}_i^{\mathcal{W}}$
$j$
$\mathbf{x}_f^{\mathcal{W}}$
$\mathbf{I}_{\mathcal{W}}$

Figure 4.4: Definition of the target geometric configuration for the formation. Note the definition of the formation frame, $\mathbf{I}_{\mathscr{F}}$. For simplification, the third dimension is omitted. Note how the UAV pose is expressed in this frame $\mathbf{I}_{\mathscr{F}}$, as well as in the absolute frame $\mathbf{I}_{\mathscr{W}}$.

### 4.2.2 Formation controller

The *Formation Controller* implements the formation control algorithm on board each UAV $i$, based on the tools presented in Section 3.6. Recall from that section that a target geometric configuration is defined by a set of desired inter-vehicle poses $(\mathbf{x}_{ij,d}^{\mathscr{W}}, \mathbf{R}_{\psi_{ij,d}})$, and a zero inter-vehicle velocity. In this work, these inter-vehicle constraints are generated making use of a *formation frame*, denoted as $\mathbf{I}_{\mathscr{F}}$, and depicted in Fig. 4.4. Similar to the previously defined flying frame, the formation frame's $z$ axis is aligned with the absolute frame. Therefore, the formation frame has an attitude with respect to the environment frame $I_{\mathscr{W}}$ ($\mathbf{R}_{\mathscr{F}}^{\mathscr{W}}$) described by the three Euler angles $(\phi, \theta, \psi) = (0, 0, \psi_f)$. The frame $x$ axis $\mathbf{I}_{\mathscr{F}_x}$ defines the formation attitude. The frame position with respect to $I_{\mathscr{W}}$ ($\mathbf{x}_{\mathscr{F}}^{\mathscr{W}}$) coincides with the geometric center of the formation. The desired position of each UAV $i$ in the formation frame $(\mathbf{x}_{fi,d}^{\mathscr{F}}, \mathbf{R}_{\psi_{fi,d}})$ is first defined. The desired inter-vehicle poses $(\mathbf{x}_{ij,d}^{\mathscr{F}}, \mathbf{R}_{\psi_{ij,d}})$ can then be computed. These desired values can be transformed at any instant of time to the absolute frame using the formation frame pose as follows:

$$(\mathbf{x}_{ij,d}^{\mathscr{W}}, \mathbf{R}_{\psi_{ij,d}}) = (\mathbf{x}_{\mathscr{F}}^{\mathscr{W}} + \mathbf{R}_{\psi_f}^{\mathscr{W}} \mathbf{x}_{ij,d}^{\mathscr{F}}, \mathbf{R}_{\psi_{ij,d}}).$$

In Section 3.6, the formation controller that managed the desired inter-vehicle positioning constraints was defined in terms of absolute inter-vehicle localization information Eq. (3.18). However, this information is not available on board the UAV, and the relative inter-vehicle localization measured by the onboard relative positioning systems has to be used instead. This information is expressed in each UAV local frame, making it difficult to obtain a common frame on which the desired inter-vehicle geometric constraints $\mathbf{x}_{ij,d}^{\mathscr{W}}$ can be defined, as discussed in Section 2.2.2. Therefore, this work is based on range and bearing control laws, such as the one proposed in [31]. In this case the desired inter-vehicle range is used, which is independent of the frame considered for the control law. Since the $z$ axis of the flying frame is the same for all UAVs, this work decouples the formation control into horizontal and vertical components. For this reason, the previous absolute inter-vehicle localization information is divided into horizontal components, $\mathbf{x}^{\mathscr{W}}_{ij_h} = (x^{\mathscr{W}}_{ij}, y^{\mathscr{W}}_{ij})^T$ and $\mathbf{v}^{\mathscr{W}}_{ij_h} = (v^{\mathscr{W}}_{ij_x}, v^{\mathscr{W}}_{ij_y})^T$, and vertical components, $z^{\mathscr{W}}_{ij}$ and $v^{\mathscr{W}}_{ij_z}$. The horizontal position component can be transformed into polar coordinates $(e_{ij_h}, \mathbf{r}^{\mathscr{W}}_{ij_h})$, where $e_{ij_h}$ is the range between UAVs $i$ and $j$, and $\mathbf{r}^{\mathscr{W}}_{ij_h}$ is a unitary vector defining the bearing between UAVs $i$ and $j$. Each UAV $i$ controls the inter-vehicle position of its neighbors as follows. The height component $z^{\mathscr{W}}_{ij}$ is controlled with a simple consensus equation, similarly to [27, 87]:

$$\mathbf{a}^{\mathscr{W}}_{i_z,d} = k_p \sum_{j=1}^{N} \mathbf{L}_{ij} \left( z^{\mathscr{W}}_{ij,d} - z^{\mathscr{W}}_{ij} \right) + k_v \sum_{j=1}^{N} \mathbf{L}_{ij} v^{\mathscr{W}}_{ij_z}, \tag{4.7}$$

where $\mathbf{a}^{\mathscr{W}}_{i_z,d}$ is the desired vertical acceleration for UAV $i$, $k_p$ and $k_v$ are gain parameters for the position and the velocity components of the controller. To control the horizontal position component $(e_{ij_h}, \mathbf{r}^{\mathscr{W}}_{ij_h})$, the range and bearing controller proposed in [31] is used, but extended for a double integrator case and simplified to the case of holonomic vehicles:

$$\mathbf{a}^{\mathscr{W}}_{i_h,d} = k_p \sum_{j=1}^{N} \mathbf{r}^{\mathscr{W}}_{ij_h} \mathbf{L}_{ij} \left( e_{ij_h} - e_{ij_h,d} \right) + k_v \sum_{j=1}^{N} \mathbf{L}_{ij} \mathbf{v}^{\mathscr{W}}_{ij_h}, \tag{4.8}$$

where $\mathbf{a}^{\mathscr{W}}_{i_h,d} = (\mathbf{a}^{\mathscr{W}}_{i_x,d}, \mathbf{a}^{\mathscr{W}}_{i_y,d})$ is the desired horizontal acceleration for UAV $i$ and $e_{ij_h,d}$ is the desired horizontal range between UAV $i$ and $j$, obtained from $||(x^{\mathscr{W}}_{ij_d}, y^{\mathscr{W}}_{ij_d})||$. Inter-vehicle positions controlled in this way allow the edges representing them to be added into $\mathscr{G}_F$. To control these edges, the respective inter-vehicle localization information is needed, which means these control edges must also belong to $\mathscr{E}_S$.

Note that the previous control law can now be expressed in the UAV $i$ flying frame, as discussed in Section 4.1, by multiplying both sides of the Eqs. (4.7) and (4.8) with $\mathbf{R}_{\psi_i}$. Note that this transformation only alters Eq. (4.8). Eq. (4.7) remains the same since the vertical axis remains unchanged during rotations about the $z$ axis. This transformation results in the following control laws in the UAV $i$ flying frame:

$$\mathbf{a}^{\mathscr{L}_i}_{i_z,d} = k_p \sum_{j=1}^{N} \mathbf{L}_{ij} \left( z^{\mathscr{L}_i}_{ij,d} - z^{\mathscr{L}_i}_{ij} \right) + k_v \sum_{j=1}^{N} \mathbf{L}_{ij} v^{\mathscr{L}_i}_{ij_z}, \tag{4.9}$$

$$\mathbf{a}^{\mathscr{L}_i}_{i_h,d} = k_p \sum_{j=1}^{N} \mathbf{r}^{\mathscr{L}_i}_{ij_h} \mathbf{L}_{ij} \left( e_{ij_h} - e_{ij_h,d} \right) + k_v \sum_{j=1}^{N} \mathbf{L}_{ij} \mathbf{v}^{\mathscr{L}_i}_{ij_h}. \tag{4.10}$$

Note that $(\mathbf{z}^{\mathscr{L}_i}_i, \mathbf{v}^{\mathscr{L}_i}_{i_z}, \mathbf{a}^{\mathscr{L}_i}_{i_z}) = (\mathbf{z}^{\mathscr{W}}_i, \mathbf{v}^{\mathscr{W}}_{i_z}, \mathbf{a}^{\mathscr{W}}_{i_z})$. This allows the previous formation control algorithm to be applied using solely the relative inter-vehicle localization information and control inputs in the UAV $i$ flying frame.

The separate inter-vehicle attitude controller referred in the end of Section 3.6 can now be implemented. The attitude constraints of the formation can be defined at random, or they are defined according to the FOV constraints present in the onboard relative positioning systems, as further explained in Section 5.2.1. In this work, the attitude controller for each UAV $i$ uses the relative bearing angle measurements of neighboring UAVs acquired by the onboard relative positioning system. Since the UAVs are assumed to always have small $\phi_i$ and $\theta_i$ values, only the relative horizontal bearing angle is considered $\beta_{ij}$, defined as the angle formed between the $x$ axis of UAV $i$'s flying frame and the horizontal component of the UAV $j$'s relative position expressed in the UAV $i$'s flying frame, $\mathbf{x}^{\mathscr{L}_i}_{ij_h}$, as depicted in Fig. 4.4.

To control $\beta_{ij}$ to its desired value ($\beta_{ij,d}$) a consensus equation similarly to Eq. (3.18) is used, as follows:

$$\omega_{i_z,d} = k_\psi \sum_{j=1}^{N} c_{\psi_{ij}} \mathbf{L}_{ij} \left(\beta_{ij,d} - \beta_{ij}\right), \tag{4.11}$$

where $k_\psi$ is a control gain and $c_{\psi_{ij}}$ is a parameter that is either 0 or 1 used to chose the neighbors that will effectively be used for the attitude controller, as it will be further explained in Section 5.2.1. The desired bearing for each neighbor $\beta_{ij,d}$ is defined from the desired positions of UAVs $i$ and $j$ defined in the formation frame ($\mathbf{x}^{\mathscr{F}}_{fi_h,d}$ and $\mathbf{x}^{\mathscr{F}}_{fj_h,d}$) and the desired attitude of UAV $i$ defined in the formation frame ($\psi_{fi,d}$). Note that this controller can already be applied in the UAV flying frame, and no further transformations are need.

Finally, in order to move the UAV or to stabilize it in the environment, Eqs. (4.10) and (4.9) are changed in order to have the UAV controlling its height and horizontal velocity in the environment, using measurements provided by the previously referred height and OF sensors, as follows:

$$\acute{\mathbf{a}}^{\mathscr{L}_i}_{i_z,d} = \mathbf{a}^{\mathscr{L}_i}_{i_z,d} + k_{p_1}(z^{\mathscr{L}_i}_{i,d} - z^{\mathscr{L}_i}_{i}) + k_{v_1}(v^{\mathscr{L}_i}_{i_z,d} - v^{\mathscr{L}_i}_{i_z}), \tag{4.12}$$

$$\acute{\mathbf{a}}^{\mathscr{L}_i}_{i_h,d} = \mathbf{a}^{\mathscr{L}_i}_{i_h,d} + k_{v_1}(\mathbf{v}^{\mathscr{L}_i}_{i_h,d} - \mathbf{v}^{\mathscr{L}_i}_{i_h}), \tag{4.13}$$

where $z^{\mathscr{L}_i}_{i,d}$ and $z^{\mathscr{L}_i}_{i}$ are respectively the desired and current UAV $i$'s height expressed in the absolute frame, $\mathbf{v}^{\mathscr{L}_i}_{i_h,d}$ and $\mathbf{v}^{\mathscr{L}_i}_{i_h}$ the desired and current velocity, and $k_{p_1}$ and $k_{v_1}$ are control gains for the position and velocity components respectively. To keep the formation into place, $(\mathbf{v}^{\mathscr{L}_i}_{i_h,d}, v^{\mathscr{L}_i}_{i_z,d})$ is set to zero, as previously discussed, and $z^{\mathscr{L}_i}_{i,d}$ for each UAV is selected in order to respect the desired inter-vehicle height constraints $z^{\mathscr{W}}_{ij,d}$. Note that with no extra command provided, the formation will just remain in the same place. To move the formation in the previously described leader-follower approach, one of the UAVs is selected as the leader, and an external system is used to send non-zero $(\mathbf{v}^{\mathscr{L}_i}_{i_h,d}, v^{\mathscr{L}_i}_{i_z,d})$ commands to it. As previously discussed in this chapter, Section 5.2.2 presents an algorithm to steer the formation through the environment with less motion delays between the leader and the followers. This novel steering algorithm uses an external system that does not need individual UAV positions or extra inter-vehicle communication. The formation controller described in Eqs. (4.12) and (4.13) will be used as a baseline formation control algorithm in this work. On top of this algorithm, multiple enhancements will be added in order to tackle several problems that arise when conducting formation control algorithms using relative sensing as sensory feedback. These enhancements are presented in Section 5.2.

Finally, this work considers the edges in $\mathscr{E}_S$ and $\mathscr{E}_F$ to be bidirectional. For $\mathscr{E}_S$ edges, this concretely means that the UAVs forming any given edge can mutually sense each other. For $\mathscr{E}_F$ edges, this concretely means that both UAVs forming the edge actively participate on the control of the mutual range. These bidirectionality assumptions are made to reduce the range instability between UAVs, caused by possible delays in the vehicle perception-to-action loop. These assumptions lead to the consideration of formation geometric configurations that place the UAVs on the convex hull of the target formation shape. UAVs can also be placed inside the convex hull, at different heights to avoid occlusions of the onboard sensors. Additionally, unidirectional edges can still be considered with the previous described algorithm, as long as the overall formation rigidity is kept. These bidirectionality assumptions are further discussed in Section 7.4.

# Chapter 5

# System Design and Implementation

This chapter provides a description of the relative positioning systems used in this work, as well as how the formation control algorithms presented in the last chapter can be used with these systems. The camera-based and IR-based positioning systems are first described in terms of the specific hardware and localization algorithms required. Here, the computational complexity, FOV, and design complexity problems characterizing these technologies are introduced and design solutions to tackle some of them are proposed. The remaining limitations are listed and tackled in the formation control problem. Finally, a method on how to move the multi-UAV system in formation through the environment is described, based on the UAV onboard relative and absolute positioning systems assumed available.

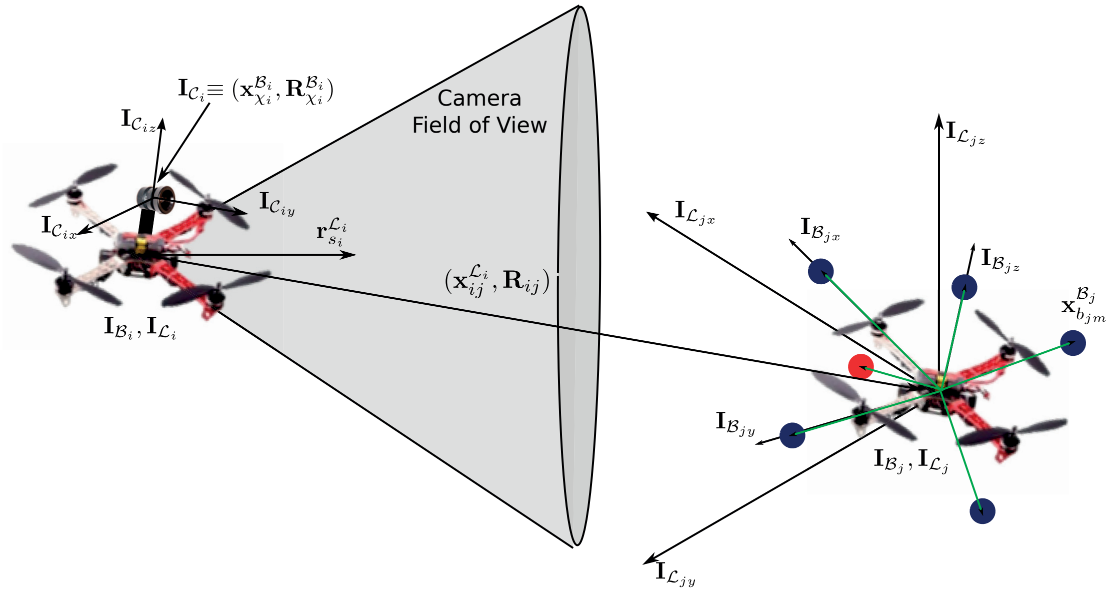
## 5.1 Relative positioning systems

The goal of relative positioning systems is to enable each UAV $i$ to compute neighbor UAV $j$ pose with respect to its flying frame $(\mathbf{x}_{ij}^{\mathscr{L}_i}, \mathbf{R}_{ij})$, as discussed in the previous chapter.

### 5.1.1 Camera-based system

A camera-based relative positioning system was selected since the required hardware is mature, leading to its use in a wide variety of applications. This technology is particularly of interest for estimation and control of UAVs since it can provide measurements with high accuracy, necessary to tackle UAV dynamics. As previously discussed in Section 2.1.1, multiple active beacon approaches, as in [29], are typically used to maximize sensor accuracy and to make the system more robust to environment conditions and blob detection imperfections. This work considers such an approach, equipping each UAV with a multi-beacon marker and an onboard camera, as conceptually depicted in Fig. 5.1. This enables each UAV $i$ to compute the relative inter-vehicle localization of each neighbor $j$, $(\mathbf{x}_{ij}^{\mathscr{L}_i}, \mathbf{R}_{ij})$, along with its relative velocity $\mathbf{v}_{ij}^{\mathscr{L}_i}$. An RGB camera is used instead of an IR camera in order to still allow extraction of environment features from the same images if required by other algorithms.

As previously discussed in Section 2.1.1, this technology is subjected to two main problems. The



$\mathbf{I}_{\mathcal{C}_i} \equiv (\mathbf{x}^{\mathcal{B}_i}_{\chi i}, \mathbf{R}^{\mathcal{B}_i}_{\chi i})$
$\mathbf{I}_{\mathcal{C}_{iz}}$
$\mathbf{I}_{\mathcal{C}_{ix}}$
$\mathbf{I}_{\mathcal{C}_{iy}}$
$\mathbf{r}^{\mathcal{L}_i}_{s_i}$
Camera
Field of View
$(\mathbf{x}^{\mathcal{L}_i}_{ij}, \mathbf{R}_{ij})$
$\mathbf{I}_{\mathcal{B}_i}, \mathbf{I}_{\mathcal{L}_i}$
$\mathbf{I}_{\mathcal{L}_{jz}}$
$\mathbf{I}_{\mathcal{L}_{jx}}$
$\mathbf{I}_{\mathcal{B}_{jx}}$
$\mathbf{I}_{\mathcal{B}_{jz}}$
$\mathbf{x}^{\mathcal{B}_j}_{b_{jm}}$
$\mathbf{I}_{\mathcal{B}_{jy}}$
$\mathbf{I}_{\mathcal{B}_j}, \mathbf{I}_{\mathcal{L}_j}$
$\mathbf{I}_{\mathcal{L}_{jy}}$

Figure 5.1: Conceptual diagram of the proposed camera-based positioning system. Multiple active localization beacons (blue circles) inserted on UAV $j$ are observed by the neighbor UAV $i$'s camera, allowing UAV $i$ to localize UAV $j$ in its flying frame according to the chosen beacon 3D layout. A pulsating ID beacon (red circle) is used to indicate the UAV $j$ ID. The UAV camera FOV is placed in the center of the UAV local frame as the camera is assumed to be near this center. Note the difference between the UAV $i$'s camera ($\mathbf{I}_{\mathscr{C}_i}$), body/marker ($\mathbf{I}_{\mathscr{B}_i}$), and flying ($\mathbf{I}_{\mathscr{L}_i}$) frames.

first one relates to the sensor limitations either on accuracy or FOV. This is especially true for the 3D case, because of the challenging sensing design, either due to the fact that the vehicle body represents an obstacle for the sensor itself, or because there is a tradeoff between the sensing area that needs to be covered and the resolution of the sensor. This causes the existing approaches for UAVs to rely on small sensor FOV, in order to achieve high measurement accuracies while maintaining a low computational power capable to run on board the UAVs. In a multi-vehicle system, this fact constrains the possible interactions between neighbors. In this work, the FOV is kept within reasonable values while low camera pixel resolutions are adopted to allow the implementation of the system on board the UAV. Here, the measurement noise is studied and modeled in order to acquire stable relative position and velocity estimations in these conditions. Additionally, from the studied sensor model, a system performance metric is defined, characterizing the system performance according to the chosen design. This allows mapping the performance of different solutions and, therefore, potentially simplifying future design choices when facing different requirements, for example in terms of FOV or maximum range.

The second problem is that multi-vehicle coordination methods usually require each vehicle to have an unique ID. Current approaches implement this capability either by using different colored features, which can lead to limited scalability with the number of vehicles, or using different configurations of the beacon 3D layout, which lead to a cumbersome platform design and adds combinatorial complexity for the marker detection and localization. In this work, this problem is tackled by using a pulsating beacon to avoid using different configurations of the beacon 3D layout, as depicted in Fig. 5.1.

The next sections provide the details about the developed relative inter-vehicle localization algorithm for this relative positioning system. First, the algorithm used to extract relative neighbor positions and attitudes is described. Then, the tracking algorithm implemented to obtain relative neighbor localization estimates together with the relative velocity is introduced. The noise of the measurement model used in the tracking algorithm is then evaluated. Here, a metric based on the maximum desired measurement noise in function of the neighbor range from the UAV is discussed, which allows the mapping of desired performances across different design choices. Finally, the hardware design is described, taking special attention to the visibility constraints of the system.

#### 5.1.1.1 3D relative pose and ID extraction

The proposed camera-based positioning system defines the pose of each UAV $i$ by means of a multi-beacon marker with the same frame as the UAV body frame, as depicted in Fig. 5.1. In this way, the marker or the UAV localization are the same (frame $\mathbf{I}_{\mathscr{B}_i}$). The marker is composed by a set of $N_b$ beacons. Each beacon $m$ ($b_{im}$) is placed at a certain position in the marker frame, $\mathbf{x}^{\mathscr{B}_i}_{b_{im}}$. Additionally, an onboard RGB camera ($\chi_i$) is placed with a certain pose in the marker frame ($\mathbf{x}^{\mathscr{B}_i}_{\chi_i}, \mathbf{R}^{\mathscr{B}_i}_{\chi_i}$). The camera frame ($\mathbf{I}_{\mathscr{C}_i}$) is defined by the camera's 3D attitude. The $x$ axis of this frame ($\mathbf{I}_{\mathscr{C}_{ix}}$) corresponds to the front of the camera. The camera has an horizontal FOV, $\theta_h$, and a vertical FOV, $\theta_v$, which is centered in the camera front direction. As discussed in Section 4.2.1, since the UAV does not tilt much while flying it is possible to represent the camera for each UAV $i$ as a virtual sensor centered at the UAV position, with its front axis, $\mathbf{r}^{\mathscr{L}_i}_{s_i}$, defined from the projection of the camera front in the UAV $xy$ plane of its flying frame. The virtual sensor FOV has the same parameters as the actual camera but it is centered in the new sensor front direction, as depicted in Fig. 5.1. The 3D beacon layout of the marker and the camera pose in the marker frame are the same for all UAVs, in order to avoid additional computational complexity in the localization algorithm.

Two types of beacons are considered in this system, each one with different RGB lighting properties to separate their detection processes, as discussed in the following sessions and depicted in Fig. 5.1. On the one hand, the localization beacons are used to obtain the relative localization of UAV $i$ relative to the other neighbors. On the other hand, the ID beacon is used to identify UAV $i$ from the other team members. There are several advantages obtained by introducing this separation. Firstly, the relative inter-vehicle localization accuracy and bandwidth are not affected by a temporary loss of the pulsating ID beacon. Secondly, the beacon ID is not generated using different geometrical configurations, which would require a careful choice of marker positions to prevent possible ID misclassification, and would increase the computational complexity of the used classification algorithms with the number of IDs. Finally, as discussed later, this additional beacon also allows for additional filtering capabilities, providing a way to confirm the existence of the observed marker. However, the initial ID extraction phase takes some time, which increases as the time used between marker pulses increases to allow for possible additional IDs.

UAV $i$ is able to compute UAV $j$'s pose in its flying frame ($\mathbf{x}^{\mathscr{L}_i}_{ij}, \mathbf{R}_{ij}$) by processing the images

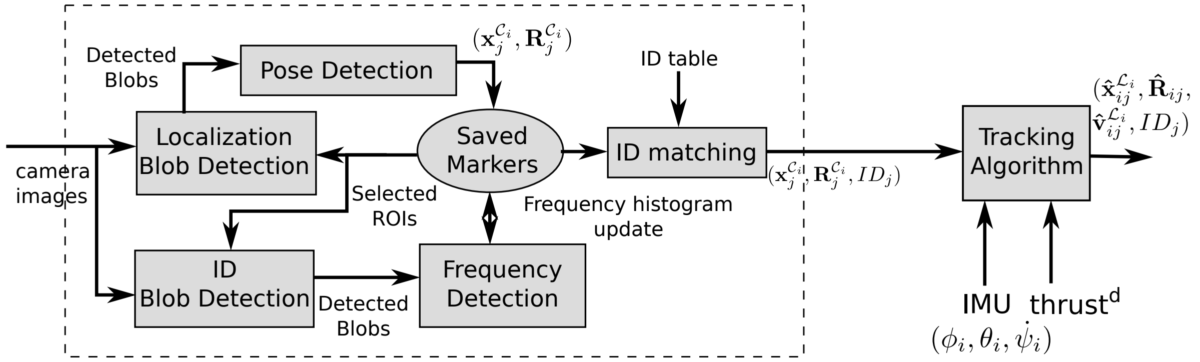

Detected Blobs
Pose Detection
$(\mathbf{x}_j^{\mathcal{C}_i}, \mathbf{R}_j^{\mathcal{C}_i})$
ID table
Localization Blob Detection
Saved Markers
ID matching
camera images
Selected ROIs
Frequency histogram update
ID Blob Detection
Detected Blobs
Frequency Detection
$(\mathbf{x}_j^{\mathcal{C}_i}, \mathbf{R}_j^{\mathcal{C}_i}, ID_j)$
Tracking Algorithm
$(\hat{\mathbf{x}}_{ij}^{\mathcal{L}_i}, \hat{\mathbf{R}}_{ij}, \hat{\mathbf{v}}_{ij}^{\mathcal{L}_i}, ID_j)$
IMU
thrust$^d$
$(\phi_i, \theta_i, \dot{\psi}_i)$

Figure 5.2: Camera-based positioning system overall software architecture and information flow, for a specific UAV $i$. Sensor measurements are extracted from the camera image and fed into a tracking algorithm for velocity estimation and measurement noise filtering. The block "Saved Markers" saves the relative 3D pose and pulsating beacon state of potential markers observed on each image. This allows to compute the pulsating frequency of the ID beacon across images for ID extraction, and to reduce the algorithm complexity on each image through the establishment of a Region Of Interest (ROI) on the image for each marker.

acquired by the onboard camera, according to the algorithm depicted in Fig. 5.2. Initially, a blob detection algorithm is applied to the image in order to identify potential localization beacons. Blobs with a size smaller than a certain threshold $\sigma_{c1}$ are discarded. With the resulting blob set, all combinations of $(N_b - 1)$ blobs configurations are tested as a potential UAV $j$ multi-beacon marker (here the ID beacon is discarded). Since the analysis of all blob configurations is a combinatorial problem, two additional pruning algorithms are performed to discard in advance wrong configurations:

- The first pruning algorithm is based on the relationship between the beacon size and the maximum distance between two localization beacons in the marker. This allows to define a maximum blob spread of a specific blob configuration $\sigma_{c2}$ according to the average pixel size of the observed blobs.

- The second pruning algorithm is based on the relationship between the observed blob pixel sizes and the distance between the localization beacons and the camera (beacons that are further away will have a smaller projection). This allows to define a maximum blob pixel size spread $\sigma_{c3}$ according to the maximum distance between localization beacons in the marker.

The tunning of $\sigma_{c1}$, $\sigma_{c2}$, and $\sigma_{c3}$ is further discussed in Section 7.1.1.1.

For each remaining blob configuration $j$, an association between the observed blobs with the respective localization beacons of the marker is performed using the 3D beacon layout of the marker defined a priori. All association combinations within each blob configuration are attempted. For each combination, three of the associations are fed into the perspective three points (P3P) algorithm, described in [62]. This algorithm is able to obtain four possible solutions for the marker pose in the camera frame $(\mathbf{x}_j^{\mathscr{C}_i}, \mathbf{R}_j^{\mathscr{C}_i})$ justifying the three associations given to the algorithm. The other associations are used to select the right solution and to evaluate its correctness, by comparing the distance error in pixels between the 2D positions of each blob in the image and the 3D projection of the localization beacon associated to it in the image, computed using the transformation of the solution that is being tested. Configurations with an error larger than a threshold $\sigma_{c4}$ are not considered. Valid configurations are saved with a local temporary identifier while the ID extraction is not concluded. Their extracted 3D poses are used to define Region Of Interest (ROI) for the marker in the image and predict the blob positions for a more efficient detection in the next images. This ROI is centered at the predicted projected marker center. The ROI's size is dynamically computed by assuming the maximum image blob spread that a marker would generate when projected to the camera at the predicted distance between the marker and the camera. Finally, for saved blob configurations, only three blob associations are required for the detection, as the P3P solutions are validated by an additional beacon projection from the previous image.

The saved blob configurations are then subjected to an ID extraction process, using the pulsating ID beacon. A new blob detection algorithm is launched for each saved blob configuration to detect blobs with the lighting properties chosen for these beacons. Since at this stage there is already a marker pose estimate for this configuration, the blob detection algorithm is done in a smaller ROI. This ROI has its center around the predicted pulsating beacon projection position in the image, and its size is computed from the predicted distance between the marker and the camera. Again, a threshold on the blob size $\sigma_{c5}$ is used to remove blobs that are too small. This threshold is still relevant since, although most of the ROI will comprehend the UAV under predefined lighting properties, some portions can also include the environment and possible clutter. The tuning of $\sigma_{c4}$ and $\sigma_{c5}$ is further discussed in Section 7.1.1.1.

The marker ID is associated to an unique ID beacon pulsating frequency. To detect these frequencies, the time between two beacon pulses is recorded and used to create a pulsating frequency histogram. This histogram is used to compute an average pulsating frequency, which is then matched with a potential corresponding marker ID. Different times between two beacon pulses result in different recorded pulsating frequencies. The average pulsating frequency is computed considering only the recorded pulsating frequencies with more than five appearances in the histogram. Detecting the marker ID independently from the marker localization algorithm has the advantage that any false positive generated by the pose extraction will most likely be filtered out, as it is unlikely for it to emit an unique frequency with enough relevance for it to be detected as a marker with a valid ID. Additionally, using a pulsating frequency histogram avoids losing the correct marker ID when clutter affects the individual pulsating frequency measurements. The marker ID initialization takes the time needed to observe at least one relevant pulsating frequency; after this period, the ID information is always available.

#### 5.1.1.2 Tracking algorithm

Each acquired measurement of marker $j$ is forwarded to a tracker. This tracker keeps a list of observed markers, each described by its ID, its estimated position and velocity, and its attitude in the UAV $i$'s flying frame $(\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}, \hat{\mathbf{v}}_{ij}^{\mathscr{L}_i}, \hat{\mathbf{R}}_{ij})$. The filter tracks each marker independently at discrete time intervals of $\Delta t$ seconds, with a version of the algorithm described in Section 4.2.1. The estimates $(\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}, \hat{\mathbf{v}}_{ij}^{\mathscr{L}_i})$ are predicted using a model adapted from Eq. (4.4) as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{ij}^{\mathscr{L}_i}(k+1) \\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I_3} \; \Delta t \mathbf{I_3} \\ \mathbf{0_3} \;\; \mathbf{I_3} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{ij}^{\mathscr{L}_i}(k) \\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k) \end{bmatrix} - \begin{bmatrix} \frac{\Delta t^2}{2}\mathbf{I_3} \\ \Delta t \mathbf{I_3} \end{bmatrix} \mathbf{a}_i^{\mathscr{L}_i}(k) \\ + \begin{bmatrix} \mathbf{x}_{ij}^{\mathscr{L}_i}(k) \times \Omega_i(k) \\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k) \times \Omega_i(k) \end{bmatrix} + \xi_{ij}^{\mathscr{L}_i}(k) \end{aligned} , \tag{5.1}$$

where $\mathbf{a}_i^{\mathscr{L}_i}(k)$ is the UAV $i$'s egomotion described in the UAV flying frame, which can be computed as described in the Section 4.2.1. For each measurement of marker $j$ generated by the previous described image processing algorithm, the tracker updates the 3D relative inter-vehicle pose and velocity estimations of the respective marker on the tracking list. First, the measurement is converted from the camera to the flying frame, using the known camera pose in UAV $i$'s body frame $(\mathbf{x}_{\chi_i}^{\mathscr{B}_i}, \mathbf{R}_{\chi_i}^{\mathscr{B}_i})$, and the current $\phi_i$ and $\theta_i$ values acquired by the UAV $i$'s IMU sensor. Then the estimates $(\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}, \hat{\mathbf{v}}_{ij}^{\mathscr{L}_i})$ are updated with the following measurement model:

$$\begin{bmatrix} \mathbf{x}_{ij_o}^{\mathscr{L}_i}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{I_{3x3}} & \mathbf{0_{3x3}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{ij}^{\mathscr{L}_i}(k) \\ \mathbf{v}_{ij}^{\mathscr{L}_i}(k) \end{bmatrix} + \eta_{ij}^{\mathscr{L}_i},, \tag{5.2}$$

where $\eta_{ij}$ is the term defining the uncertainty of the relative positioning sensor measurements for each observed position of marker $j$ relative to the UAV $i$'s flying frame. This noise term has a high influence on the relative velocity estimations, especially for low resolution cameras, since small relative position measurement fluctuations can lead to high relative velocity fluctuations. The relative inter-vehicle attitude estimates are directly taken from the relative inter-vehicle attitude measured by the relative positioning sensor:

$$\begin{bmatrix} \hat{\phi}_{ij} & \hat{\theta}_{ij} & \hat{\psi}_{ij} \end{bmatrix} = \begin{bmatrix} \phi_{ij_o} & \theta_{ij_o} & \psi_{ij_o} \end{bmatrix}. \tag{5.3}$$

To minimize the effect of the measurement noise, a study of its characteristics is performed, detailed in the next section.

It is important to consider the case when the marker disappears from the camera FOV for a significant time interval, causing the marker pose extraction algorithm to drop the marker. Pose and ID detection has to be reinitialized upon the marker FOV reentry. To avoid an interruption on the marker tracking in this interval, the tracked marker is set into a lost mode, which allows marker measurements with unassociated IDs that are sufficiently close to the current estimated location, to be accepted as valid measurements. The situation returns to normal as soon the ID initialization is finalized. The problem of this positioning system is that it requires a minimum number of three localization beacons from the marker to be observable in the image, so as to allow the algorithm to detect and localize the marker. This can result in problems when considering inter-beacon occlusions in the camera image. In the next sections, the visibility constraints for the marker design are defined in order to guarantee a minimum number of observable beacons in the desirable observation area. Less than three beacons can be possible if the UAV is already being tracked and additional least square [29], or particle filter [19] methods are

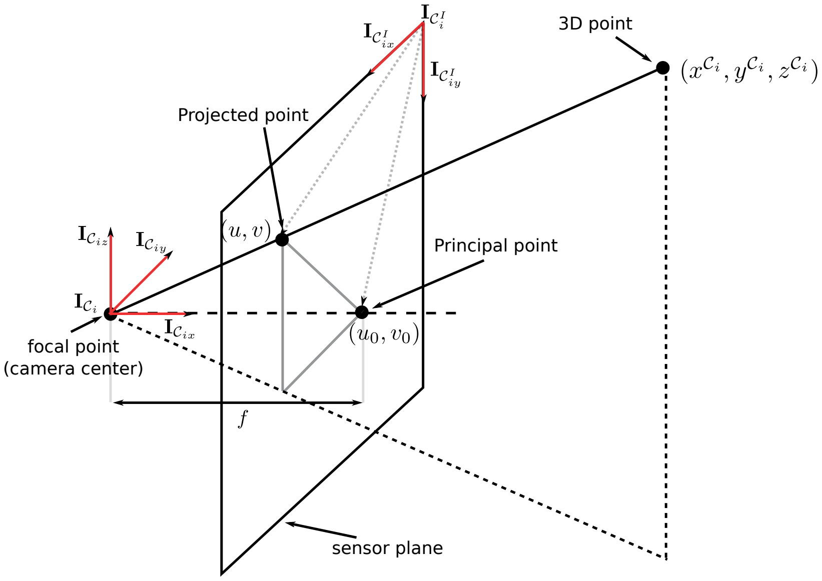

$\mathbf{I}_{\mathcal{C}_i^I}$
$\mathbf{I}_{\mathcal{C}_{ix}^I}$
$\mathbf{I}_{\mathcal{C}_{iy}^I}$
3D point
$(x^{\mathcal{C}_i}, y^{\mathcal{C}_i}, z^{\mathcal{C}_i})$
Projected point
$(u, v)$
Principal point
$\mathbf{I}_{\mathcal{C}_{iz}}$
$\mathbf{I}_{\mathcal{C}_{iy}}$
$\mathbf{I}_{\mathcal{C}_i}$
$\mathbf{I}_{\mathcal{C}_{ix}}$
focal point
(camera center)
$(u_0, v_0)$
$f$
sensor plane

Figure 5.3: Description of the pin-hole camera model. A sensor plane with $N_{p_h}$ horizontal and $N_{p_v}$ vertical pixels form an image consisting of projections of points in 3D space to the sensor plane. The projection of an observed point in the sensor plane corresponds to the intersection of the line connecting the camera focal point and the observed point in the 3D space.

.

used. However, this factor is not considered to be the main focus of this work, and the implementation of such methods has not been considered.

#### 5.1.1.3 System performance characterization

In this work, the camera sensor is assumed to follow the pin-hole model. This model is accurate for cameras with lenses that provide low image distortion, usually characterized by lower FOVs but higher camera resolutions. In this model, the sensor is characterized as a plane with $N_{p_h}$ horizontal pixels and $N_{p_v}$ vertical pixels of a certain size and a focal point centered in the camera position $\mathbf{x}_{\chi_i}^{\mathscr{B}_i}$ and at a certain distance $f$ from the sensor plane, denominated as the camera focal length. The axis perpendicular to the sensor plane and passing through the focal point is called the principal axis. The intersection of this axis with the sensor plane is called the principal point. Ideally this point coincides with the center of the sensor but generally can be any point in the sensor plane, represented in the camera image frame as $(u_0, v_0)$. The projection of a point $\mathbf{x}$ in 3D space in the camera image is modeled by a line connecting $\mathbf{x}$ to the camera focal point. The intersection of this line with the sensor plane corresponds to the projected pixel, which can be represented in the camera image frame $(u, v)$ as shown in Fig. 5.3.

From Fig. 5.3 it is possible to compute the pixel coordinates $\mathbf{p}^{\mathscr{C}_i^I} = (u, v)$ in the camera image frame $\mathbf{I}_{\mathscr{C}_i^I}$ corresponding to a point in 3D space $\mathbf{x}^{\mathscr{C}_i} = (x^{\mathscr{C}_i}, y^{\mathscr{C}_i}, z^{\mathscr{C}_i})$ described in UAV $i$'s camera frame $\mathbf{I}_{\mathscr{C}_i}$. For

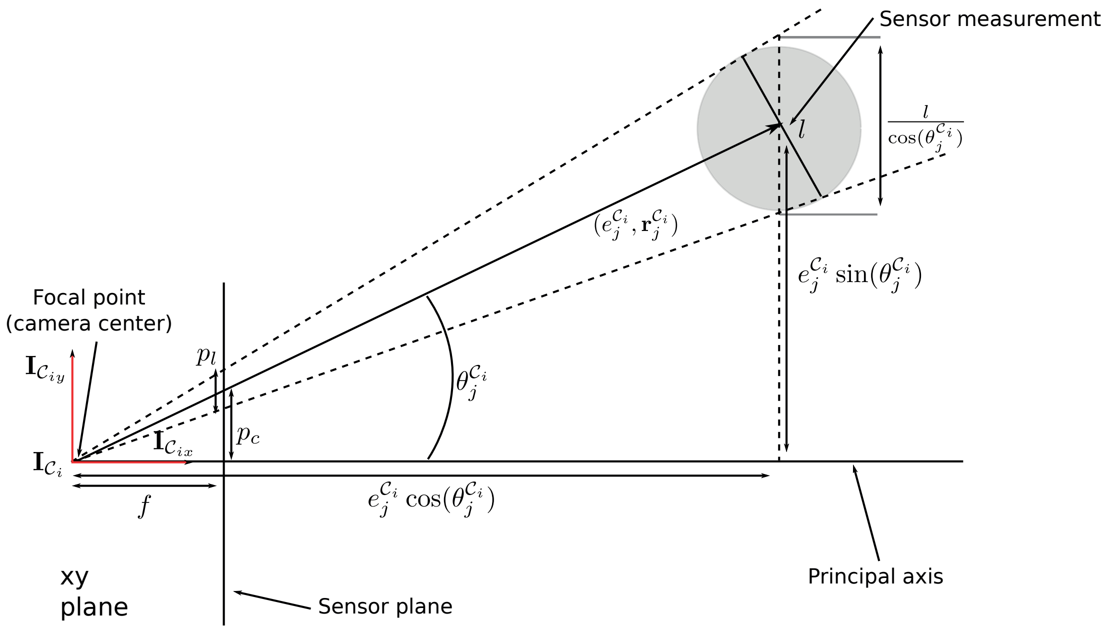

Sensor measurement
$l$
$\frac{l}{\cos(\theta_j^{\mathcal{C}_i})}$
$(e_j^{\mathcal{C}_i}, \mathbf{r}_j^{\mathcal{C}_i})$
$e_j^{\mathcal{C}_i} \sin(\theta_j^{\mathcal{C}_i})$
Focal point (camera center)
$\mathbf{I}_{\mathcal{C}_{iy}}$
$p_l$
$\theta_j^{\mathcal{C}_i}$
$p_c$
$\mathbf{I}_{\mathcal{C}_{ix}}$
$\mathbf{I}_{\mathcal{C}_i}$
$f$
$e_j^{\mathcal{C}_i} \cos(\theta_j^{\mathcal{C}_i})$
xy plane
Sensor plane
Principal axis

Figure 5.4: Modeling noise for the camera-based relative positioning system measurements. The observed marker size is characterized by its circumsphere of diameter $l$ and by its range and bearing measurements with respect to the camera frame. Here the marker is assumed to have a spherical shape, so rotations around itself do not affect the image projection. The marker 3D sphere is represented as a 2D circle for simplification purposes.

the pixel coordinate $u$ ($x$ axis on the sensor plane), the computation is as follows:

$$(u-u_0) = -f\frac{y^{\mathscr{C}_i}}{x^{\mathscr{C}_i}}, \tag{5.4}$$

where $u$, $u_0$ and $f$ are all quantities represented in pixels. For the pixel coordinate $v$ ($y$ axis on the sensor plane) the computation is analogous.

The measurement of marker $j$ acquired from the image processing algorithm described in Section 5.1.1.1 is here modeled by a sphere with a range and a bearing with respect to the camera frame $(e_j^{\mathscr{C}_i}, \mathbf{r}_j^{\mathscr{C}_i})$ and with diameter $l$. To simplify the model characterization, and without loss of generality, this 3D sphere is represented by a 2D circle in the camera frame $xy$ plane $(\mathbf{I}_{\mathscr{C}_{ix}}, \mathbf{I}_{\mathscr{C}_{iy}})$, as depicted in Fig. 5.4. From the figure it is possible to conclude that the circle diameter $l$ has a size in the image $p_l$ of:

$$p_l = \frac{lf}{e_j^{\mathscr{C}_i}\cos^2(\theta_j^{\mathscr{C}_i})}, \tag{5.5}$$

where $\theta_j^{\mathscr{C}_i}$ is the bearing angle formed between $\mathbf{r}_j^{\mathscr{C}_i}$ and $\mathbf{I}_{\mathscr{C}_{ix}}$, and $f$ is the camera focal length. Differentiating Eq. (5.5) in respect to $e_j^{\mathscr{C}_i}$, and rearranging the terms, it is possible to derive the measurement range error, $\delta e_j^{\mathscr{C}_i}$:

$$\delta e_j^{\mathscr{C}_i} = -\frac{(e_j^{\mathscr{C}_i})^2}{lf}\cos^2(\theta_j^{\mathscr{C}_i})\delta p_l \tag{5.6}$$

where $\delta p_l$ is an error on the observed circle diameter caused by pixel noise in the image. Additionally, from the figure it is also possible to conclude that the center of the circle has a projection in the image axis of:

$$p_c = f\tan(\theta_j^{\mathscr{C}_i}). \tag{5.7}$$

Differentiating Eq. (5.7) with respect to $\theta_j^{\mathscr{C}_i}$, and rearranging the terms, it is possible to derive the measurement bearing error, $\delta\theta$:

$$\delta\theta_j^{\mathscr{C}_i} = \frac{\cos^2(\theta_j^{\mathscr{C}_i})}{f}\delta p_c, \tag{5.8}$$

where $\delta p_c$ is an error on the projected circle center caused by pixel noise in the image. This work considers the maximum errors, so Eqs. (5.6) and (5.8) are considered without the cosines, since their maximum value is 1 (which corresponds to a bearing of zero).

From Eqs. (5.6) and (5.8) it is possible to provide a model for the noise term $\eta_{ij}^{\mathscr{L}_i}$, defined in Eq. (5.2). Firstly, note that $\delta p_c$ is much smaller than $\delta p_l$ especially when considering multi-beacon markers where beacon motion blur affects less the computation of the marker center in the image. Secondly, one can observe that the range error is a function of the square of the marker range from the camera and it depends on the object size, which is usually small. The bearing error is small and remains constant along the range axis. For this reason, the measurement noise is assumed to be mostly in the range component, and a transversal-longitudinal model is used for its covariance. A three dimensional Gaussian distribution is considered with a covariance matrix representing an ellipse with its major (longitudinal) axis pointing to the origin of the camera frame, and the other axis belong to the transversal plane, perpendicular to that axis, as shown in Fig. 5.5. The longitudinal axis value increases with the square of the range (see Eq. (5.6)), and the transversal axes values are constant (see Eq. (5.8)). This covariance is firstly defined in the camera frame, and then transformed into the UAV flying frame.

Here, $f$ is taken from the camera intrinsic parameters, and $l$ from the diameter of the smallest sphere that can encapsulate the whole 3D beacon layout of the marker, also called the marker circumsphere. The value of $\delta p_l$ and $\delta p_c$ in this work are considered to be the maximum pixel error that can be observed in the image, and they can be computed from experiments that will be explained in the results section. Finally, note that it is possible to define $f$ as a function of the camera FOV as:

$$f = \frac{N_{p_h}}{2tan(\theta_h/2)}, \tag{5.9}$$

where $N_{p_h}$ is the camera horizontal pixel resolution (number of pixels), and $\theta_h$ is the camera horizontal FOV. From these previous definitions, with Eq. (5.6) it is possible to define a metric for the sensor measurement maximum range error parameterized by the different camera design parameters such as its resolution and FOV. Discarding the bearing error since it is assumed to be small and constant, this

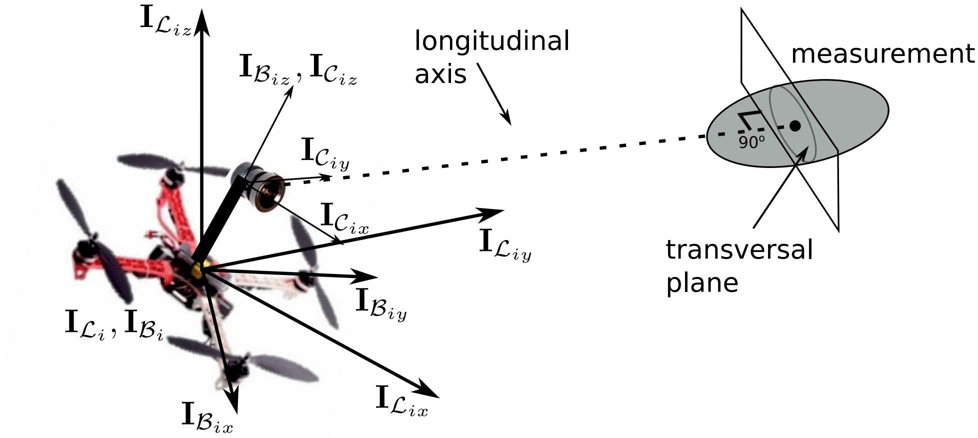

$\mathbf{I}_{\mathcal{L}_{iz}}$
$\mathbf{I}_{\mathcal{B}_{iz}}, \mathbf{I}_{\mathcal{C}_{iz}}$
longitudinal axis
measurement
90°
$\mathbf{I}_{\mathcal{C}_{iy}}$
$\mathbf{I}_{\mathcal{C}_{ix}}$
$\mathbf{I}_{\mathcal{L}_{iy}}$
transversal plane
$\mathbf{I}_{\mathcal{B}_{iy}}$
$\mathbf{I}_{\mathcal{L}_i}, \mathbf{I}_{\mathcal{B}_i}$
$\mathbf{I}_{\mathcal{L}_{ix}}$
$\mathbf{I}_{\mathcal{B}_{ix}}$

Figure 5.5: Frames considered in the estimation process. The flying frame is centered at the quadrotor position, with the same attitude, but has its $z$ axis aligned with gravity (vertical). Observation noise is shown by the gray area, modeled with a longitudinal axis and a transversal plane, computed with respect to the camera frame, and afterwards transformed to the flying frame for estimate updates.

provides a way to map desired performances across different design choices.

This result is considered to be an approximation for two reasons. Firstly, the 3D beacon layout of the marker does not necessarily place each beacon on the boundaries of a sphere with the considered diameter. This can produce errors that are different at different object rotations. Secondly, common distortion effects, such as radial distortion, affect this relationship, since the value of $\theta_h$ changes (usually increases) as the object moves away from the camera principal axis, changing $f$. This factor can be compensated through polynomial calibration of the distortion, such as done in [57] and explained in Section 7.1.1.1.

#### 5.1.1.4 System design

For the system hardware design, the type of beacons and the camera had to be selected, as well as their positions in the UAV's body frame. Regarding the position of the beacons and the camera, they were selected taking into account occlusions caused by the UAV's body in the camera and other beacons. As shown in Fig. 5.6, the camera was placed higher than the UAV's horizontal arms in order to avoid occlusions caused by the UAV's body. For this, a vertical physical support was introduced in the center of the UAV. The position of the vertical support was selected for its symmetry with respect to the UAV's body, minimizing its impact on the UAV's moment of inertia and dynamical stability. The camera orientation was chosen to be between two horizontal arms of the UAV, in order to minimize their occlusion effects, and parallel to the plane spanned by the UAV's arms, as shown in Fig. 5.6b. This defines the sensor front direction $\mathbf{r}_s^{\mathscr{L}_i}$. The FOV areas where occlusions can happen is defined by $\beta_{occ}$. As the camera height on the vertical support increases, the value of $\beta_{occ}$ decreases (see Fig. 5.6c). Ideally, the camera would be placed high enough to make $\beta_{occ}$ go to zero. Considering a length for the UAV arm of $l_{arm}$ and a vertical FOV of $\theta_v$ with its center direction parallel to the UAV's arm plane and centered between the two arms,

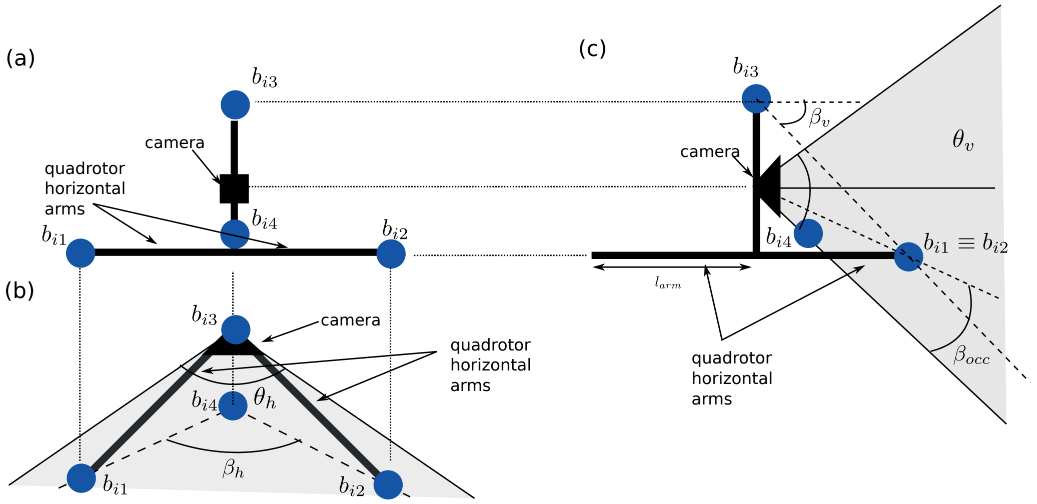


Figure 5.6: Geometric considerations for the camera-based relative positioning system hardware design. The beacon and camera positions in the UAV $i$'s body frame are considered. The ID beacon is not illustrated for simplification purposes. The front view (a), top view (b), and side view (c) of the system are illustrated.

the minimum camera height can be computed as:

$$z_{camera} = l_{arm} \cos(45^o) \tan(\theta_v/2). \tag{5.10}$$

However, note that as the camera height on the vertical support increases, the system approaches to an inverted pendulum system, which is theoretically unstable. This factor was empirically addressed by placing the relative positioning system (designed with the chosen $z_{camera}$ and $l_{arm}$) on the UAV and conducting several flying experiments. Those experiments showed that the UAV remained stable during regular flying operation.

The 3D beacon layout of the marker has been chosen in order to avoid inter-beacon occlusions by images taken from other UAVs. To tackle this problem, this work defines the concept of visibility constraints for an onboard relative positioning system. These constraints are generated from the bidirectional sensing and control assumptions referred as a control assumption in Section 4.2.2. To achieve bidirectional sensing, the visibility constraints state that the relative positioning system on a UAV must allow all neighbors currently observed by the onboard sensor to also be able to detect the observing UAV. This means that the position of the beacons must be such that the UAV can rotate itself without compromising its marker's observability from another UAV.

As shown in Fig. 5.6, four beacons ($b_{i1}$ to $b_{i4}$) are used to define the 3D beacon layout, the minimum number of points needed by the P3P algorithm for reconstruction, as discussed in Section 5.1.1.1. The beacons $b_{i1}$ and $b_{i2}$ are placed on the two arms closest to the camera's front, as shown in Fig. 5.6b. This simplifies the design and avoids body or inter-beacon occlusions that would have been generated in the other two arms when the UAV is pointing its camera directly to the neighbor. Beacon $b_{i3}$ is placed on

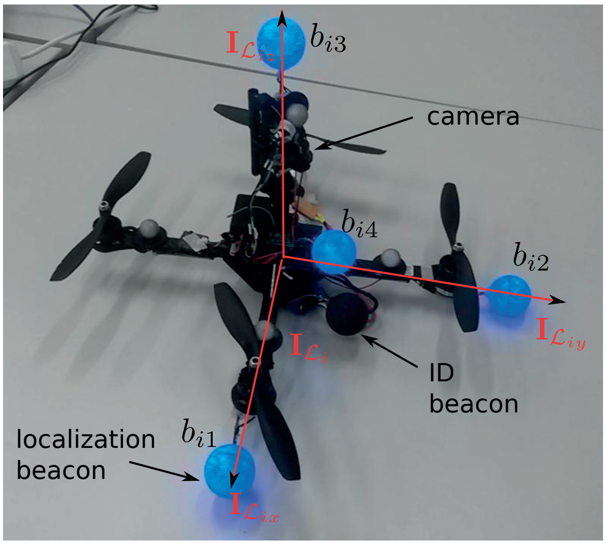

$\mathbf{I}_{\mathcal{L}_{iz}}$
$b_{i3}$
camera
$b_{i4}$
$b_{i2}$
$\mathbf{I}_{\mathcal{L}_{iy}}$
$\mathbf{I}_{\mathcal{L}_i}$
ID beacon
localization beacon
$b_{i1}$
$\mathbf{I}_{\mathcal{L}_{ix}}$

Figure 5.7: Illustration of the camera-based relative positioning system hardware. Localization beacons emit blue light, and the ID beacon emits red light through pulsed signals for ID detection. Note the chosen position for the ID beacon.

the same vertical support to which also the camera is anchored, but at a different height. The value of this height was chosen differently from the length of the UAV arms in order to avoid marker symmetries. Beacon $b_{i4}$ is placed near the UAV's center but at a slightly higher position in order to avoid occlusions caused by the UAV arms. The height of $b_{i3}$ and the position of $b_{i4}$ are computed in order to make the quantities $\beta_v$ and $\beta_h$ depicted in Fig. 5.6, smaller than $\theta_v$ and $\theta_h$, respectively, in order to avoid inter-beacon occlusions in areas that would compromise the previously defined system visibility constraints. These considerations are not reported here since they leverage basic trigonometric formulas dependent on the length of the UAV arm, $l_{arm}$.

In practice, the marker will be visible from most of the 3D space camera poses as inter-beacon occlusions only occur in specific poses. Additionally, only three beacons are required when the marker is already tracked, allowing one inter-beacon occlusion to happen. However, the described system design guarantees the visibility of this marker in the areas that are required to meet the previously defined visibility constraints. The system performance characterization, described in Section 5.1.1.3, requires a marker size ($l$). In this case the size is defined as the circumsphere that encapsulates all the localization beacons. For our marker, shown in Fig. 5.7, a circumsphere of 28 *cm* was measured. The ID beacon is placed bellow $b_{i4}$, as shown in Fig. 5.7. Its position is not relevant, as this beacon does not need to be detected all the time (it is required at the initialization stage to get the ID of the observed neighbor).

The actual beacons are implemented as a set of LEDs inside 2 *cm* plastic diffusers, visible in Fig. 5.7. Localization and ID beacons have different lighting properties, to allow a clean decoupling between the ID extraction and marker localization. The localization beacons emit blue light, but different lighting properties can be chosen, such as IR [22, 29], or the Active Led Markers (ALM) [19] in a DVS system.

To increase the beacon detection reliability in brighter scenarios, blue paint was added to take advantage of the reflections of incoming light on the markers. The ID beacon emits red light, as it was found to be well detected by the cameras on the UAVs. A smaller layer of black paint was added on top to prevent external light in bright environments from overshadowing the beacon emission. Each beacon is about 2 $g$.

The camera chosen for the developed system was the Caspa Camera FS from Gumstix with a resolution of 320x240 pixels and 90° FOV. Its weight is 22.9 $g$. By adding the weight of the support structure needed to place the camera on the UAV, of about 30 $g$, and adding the weight of the four localization beacons and one ID beacon , which is about 10 $g$, the total system weight is about 63 $g$. The system composed of both the camera and beacons requires 2 $W$ of power during operation.

### 5.1.2 Infrared-based system

The previous camera-based relative positioning system is subjected to a compromise between sensor FOV and required computational power: only a relatively small FOV can be considered in order to achieve a sufficient accuracy and measuring frequency in localization. To mitigate the impact of such tradeoff, acoustic or electromagnetic wave technologies can be adopted instead. These technologies allow the wave signals acquired by the onboard receivers to be parameterized by the relative position (range, bearing, and elevation) of the respective nearby emission sources. This in turn enables the computation of the relative localization of the emission sources with less receivers, therefore achieving higher FOV and higher measuring frequencies with lower computational requirements. However, as discussed previously in the literature review section, being active sensing techniques, sound-based and RF-based technologies are subject to strong multi-path and inter-emitter interferences, which may disturb the measurements between the emission source and the receiver, and leading to an exponentially increasing complexity when implemented in large scale. Additionally, sound-based technologies are limited by the low speed of sound in the medium, a speed considerably slower than that of electromagnetic signals. On the other hand, IR-based technologies are characterized by low inter-emitter interference, given their short emission ranges, emitter directionality, and reduced multi-path effects. This makes this technology a good candidate for reliable indoor inter-vehicle localization, and thus our main motivation for selecting it for this thesis.

This work is based on the technology developed in [86,91,92]. In these systems, a set of IR emitters and receivers are placed on each vehicle, as depicted in Fig. 5.8. The emitter placement is done to ensure a homogeneous omni-directional emission intensity for each vehicle. When a vehicle emits, the others measure the RSS at each receiver. They then compute the relative position of the emitting vehicle by fusing the information of multiple RSS measurements. More formally, thanks to this localization system each vehicle $i$ is able to compute the relative position of each neighboring vehicle $j$, $\mathbf{x}_{ij}^{\mathscr{L}_i}$, along with its relative velocity $\mathbf{v}_{ij}^{\mathscr{L}_i}$. The vehicle discrimination is performed by communicating the vehicle's ID within the IR emission. This communication can be performed either using the IR channel itself, as in [86], or a coupled RF communication system as in [91,92].

However, as previously discussed in Section 2.1.4, this technology suffers from several problems. The first problem is related to the fact that an homogeneous omni-directional IR emission intensity for each vehicle is needed in order to allow consistent range measurements at any pose relative to the vehicle. In previous contributions, such homogeneity has been obtained by carefully selecting discrete IR emitters showing similar characteristics despite the wide manufacturing tolerances as well as precisely tuned poses of the components on the vehicle. However, emission intensity irregularities still produced significant errors in range computation, as shown in [86]. Additionally, the emission heterogeneities increase in systems targeting full 3D relative localization, especially because of the high number of discrete components used (e.g., in [92] 100 IR LEDs were deployed). The system proposed in this work replaces the previously described emitter design by leveraging groups of four LEDs encapsulated in dedicated diffusers, obtaining in that way a quasi-homogeneous, small, omni-directional IR beacon, as shown in Fig. 5.8.

The second problem is that, in previous contributions, receivers were required to be placed at specific poses on the vehicle, resulting in physical constraints that might be incompatible or cumbersome to achieve given the complexity of the vehicles' body (especially true for UAVs). In these cases, the use of additional supporting structures on the vehicles is needed, ultimately resulting in an increased system complexity and weight, which makes it difficult to deploy in small-scale UAVs. In this work, a modular receiver design (see Fig. 5.8a) together with the development of a novel 3D estimation algorithm using the acquired RSS measurements allow each receiver to be placed at any pose in the vehicles. As illustrated in Fig. 5.8b, several receivers are clustered together in small receiver stations, at different poses. Multiple receiver stations can be placed at any pose on the UAV. Additional details about the receiver station will be provided in Section 5.1.2.5. Additionally, a calibration algorithm is developed to allow the computation of the actual receiver orientations after their deployment, in order to increase the system accuracy in presence of an imperfect system deployment. This allows for the exploration of different 3D vehicle design geometries for the vehicle's body without requiring additional onboard mechanical support structures.

Additionally, although previous contributions could control different groups of emitters on the same vehicle independently (as in [86]), singular emitters could not act as individual omni-directional emitters in the 3D space given their directionality properties. Therefore, the developed localization algorithms modeled the detected vehicles as single emission sources placed at the center of the vehicle. This in turn constrained the relative positioning system at each vehicle $i$ to only provide natively the relative 3D position of the neighboring vehicle $j$. The relative attitude could be acquired by sharing the relative positions between the vehicles. However, that would result in additional communication overhead and delays that affect the reactiveness of the multi-robot system. Differently from previous designs, the IR beacons developed in this work are stand-alone omni-directional, which makes it possible to consider them as omni-directional emitters in the 3D space. Since these beacons are also small, multiple beacons can also be deployed on each vehicle, as illustrated in Fig. 5.8. Therefore, by controlling each beacon on the same vehicle independently, the developed localization algorithm is able to consider multiple

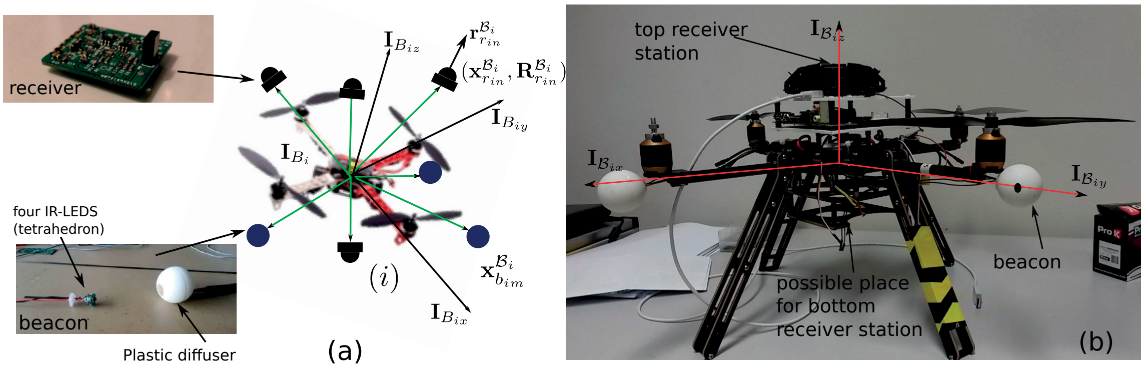

receiver
four IR-LEDS (tetrahedron)
beacon
Plastic diffuser
$\mathbf{I}_{B_{iz}}$
$\mathbf{r}^{\mathcal{B}_i}_{r_{in}}$
$(\mathbf{x}^{\mathcal{B}_i}_{r_{in}}, \mathbf{R}^{\mathcal{B}_i}_{r_{in}})$
$\mathbf{I}_{B_{iy}}$
$\mathbf{I}_{B_i}$
$\mathbf{x}^{\mathcal{B}_i}_{b_{im}}$
$(i)$
$\mathbf{I}_{B_{ix}}$
(a)
top receiver station
$\mathbf{I}_{\mathcal{B}_{iz}}$
$\mathbf{I}_{\mathcal{B}_{ix}}$
$\mathbf{I}_{\mathcal{B}_{iy}}$
beacon
possible place for bottom receiver station
(b)

Figure 5.8: Sketch of the proposed IR-based positioning system. (a) A set of IR beacons and receivers are placed on UAV $i$'s body. Each beacon $m$ ($b_{im}$) has a certain position in the UAV's body frame $\mathbf{x}^{\mathscr{B}_i}_{b_{im}}$. Each receiver $n$ ($r_{in}$) has a certain pose in the UAV's body frame $(\mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{R}^{\mathscr{B}_i}_{r_{in}})$. (b) Possible configuration of the system on a UAV. The receivers are clustered in small IR receiver stations. Each receiver station contains several receivers at different poses. Multiple receiver stations can be placed at any pose on the UAV (e.g., a top and bottom receiver station). The beacons are placed at the end of the UAV arms. Note the UAV's body or marker frame definition $\mathbf{I}_{\mathscr{B}_i}$ on both figures.

emission sources associated to each detected vehicle. This in turn allows the system to provide additional relative attitude information about the detected vehicles, as explained in the next sections. Additional details about the beacon developed in this work will be provided in Section 5.1.2.5.

Finally, it worth noticing that the IR emissions can still generate inter-beacon interference in presence of multiple vehicles. In [86], a communication algorithm based on a CSMA with collision avoidance protocol was implemented to manage possible interferences, resulting in a system that was scalable with the number of vehicles. In [91, 92] the communication protocol was implemented in a parallel RF channel instead, and it was based on a Time Division Multiple Access (TDMA) protocol. This allowed the localization algorithms to achieve higher measuring frequencies using the IR channel (since IR signal processing for the emission and reception was simplified) while benefiting from the larger communication throughput of the RF channel. However, given that RF waves have a large emission range only partially limited in NLOS conditions, and that TDMA communication protocols require the number of vehicles to be known a priori, this system does not scale well with the number of vehicles. In this work, the IR channel is reused for both localization and communication purposes, as in [86]. This can potentially allow this system to be scalable with the number of vehicles since the system can only communicate with neighboring systems within range of the IR signal. In order to further increase the measuring frequency in respect to [86], the communication protocol was implemented on top of a synchronization protocol based on a TDMA algorithm. Given the low number of vehicles concurrently operating considered in this thesis, scalability issues do not arise. However, if the number of vehicles grow, the systems will eventually become separated by multiple communication hops. These conditions can be tackled in the future by implementing a synchronization protocol based on an adaptive TDMA algorithm (or simply by considering again the CSMA protocol in [86]).

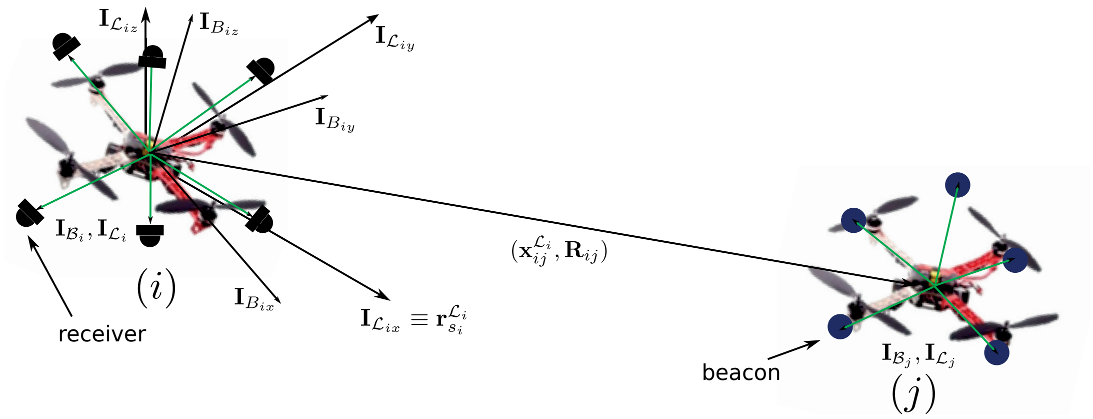

$\mathbf{I}_{\mathcal{L}_{iz}}$
$\mathbf{I}_{B_{iz}}$
$\mathbf{I}_{\mathcal{L}_{iy}}$
$\mathbf{I}_{B_{iy}}$
$\mathbf{I}_{\mathcal{B}_i},\mathbf{I}_{\mathcal{L}_i}$
$(i)$
receiver
$\mathbf{I}_{B_{ix}}$
$\mathbf{I}_{\mathcal{L}_{ix}} \equiv \mathbf{r}_{s_i}^{\mathcal{L}_i}$
$(\mathbf{x}_{ij}^{\mathcal{L}_i},\mathbf{R}_{ij})$
beacon
$\mathbf{I}_{\mathcal{B}_j},\mathbf{I}_{\mathcal{L}_j}$
$(j)$

Figure 5.9: Conceptual diagram of the proposed IR-based positioning system. Marker *j*'s relative pose in marker *i*'s flying frame $(\mathbf{x}_{ij}^{\mathscr{L}_i},\mathbf{R}_{ij})$ can be computed by measuring the RSS of several beacons of marker *j* using a set of IR receivers of marker *i*. The set of receivers on marker *i* is denoted as the virtual sensor of UAV *i*. The UAV *i*'s sensor front $\mathbf{r}_{\mathbf{s_i}}^{\mathscr{L}_\mathbf{i}}$ is defined by the *x* axis of the UAV *i*'s flying frame. Note the UAV *i*'s marker or body ($\mathbf{I}_{\mathscr{B}_i}$) and flying ($\mathbf{I}_{\mathscr{L}_i}$) frames.

The next sections provide the details about the developed localization and communication algorithms for this relative positioning system. Firstly, the used IR transmission system responsible to extract RSS measurements and beacon IDs is detailed. Secondly, the algorithm used to extract relative positions and attitudes of neighboring UAVs from the previous acquired information is described. Thirdly, the tracking algorithm implemented to obtain relative localization and velocity estimates of the neighboring UAVs is introduced, similarly to the camera-based system. Fourthly, the RSS measurement model is further analyzed, motivating the development of a calibration algorithm that is able to correct the receiver orientations after they are deployed on a UAV with slightly different orientations from those prescribed by design. Finally, the overall hardware implementation is summarized.

#### 5.1.2.1 Infrared transmission system

Similarly to the previous described camera-based system, the IR-based relative positioning system defines the pose of each UAV *i* by means of a multi-beacon marker with the same frame as the UAV body frame, as depicted in Fig. 5.8. In this way, the marker and the vehicle localization is the same. The marker is composed by a set of $N_b$ IR beacons. Each beacon $b_{im}$ is placed at a certain position in the marker frame, $\mathbf{x}_{b_{im}}^{\mathscr{B}_i}$. Additionally, a set of $N_r$ IR receivers are placed on the UAV. Each receiver $r_{in}$ is placed with a certain pose in the UAV body frame $(\mathbf{x}_{r_{in}}^{\mathscr{B}_i},\mathbf{R}_{r_{in}}^{\mathscr{B}_i})$. As depicted in Fig. 5.8, the receiver heading $\mathbf{r}_{r_{in}}^{\mathscr{B}_i}$ is characterized by its own pose defined in the marker frame, and it describes the receiver's photosensitive side. Similarly as before, and as illustrated in Fig. 5.9, the set of receivers placed on each UAV *i* defines a virtual sensor centered at the UAV position, with its front defined to be the same as the UAV front. The front of the virtual sensor $(\mathbf{r}_{\mathbf{s_i}}^{\mathscr{L}_\mathbf{i}})$ is not relevant since no FOV constraints are defined for it due to the quasi omni-directionality of this system. The 3D beacon layout of the marker and the receiver poses in

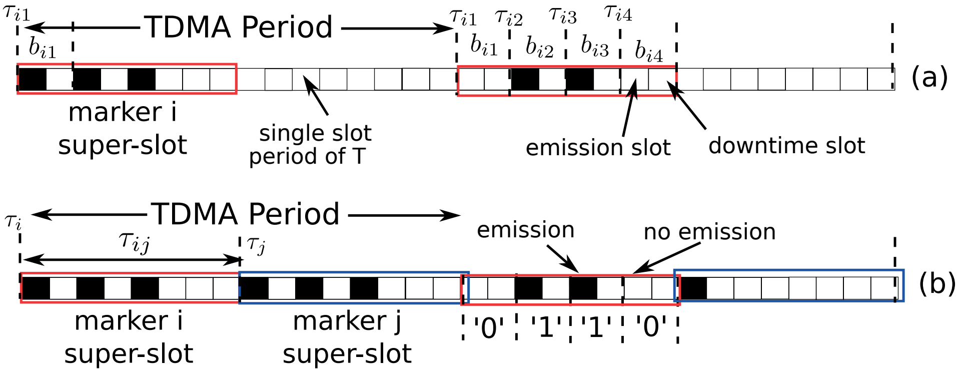


Figure 5.10: Description of the TDMA protocol used in the IR-based positioning system. (a) Emission scheduling for one single marker composed of four beacons ($b_{i1}$ to $b_{i4}$), with a beacon emission slot period of $T$. Each beacon $b_{im}$ emits on the slot $\tau_{im}$. An additional downtime slot is added for each emission slot in order to avoid inter-beacon emission interference. A TDMA period is defined between two consecutive emission slots of the same beacon. (b) Emission scheduling for two markers, each one composed of four beacons. The emission slots of the first beacon of each marker are spaced by $\tau_{ij}$. Note that when a beacon emits in its slot (black square in its emission slot), it is transmitting a digital '1' for the current bit of its digital pulse sequence.

the marker frame are the same for all UAVs, in order to avoid additional computational complexity in the localization algorithm and to simplify the replication process across UAVs.

As discussed in the previous section, each beacon represents a quasi-homogeneous omni-directional emission source. The IR light emitted from these beacons is detected by the receivers on neighboring UAVs, so that the respective UAVs can compute the relative pose of the detected marker. The RSS measurement of each beacon $b_{jm}$ acquired by receiver $r_{in}$, $RSS^{r_{in}}_{b_{jm}}$, is related to the relative pose between the respective beacon and receiver. When multiple beacons are emitting (from the same marker or from different markers), inter-beacon interference can occur, distorting the value of $RSS^{r_{in}}_{b_{jm}}$. Additionally, a mechanism is needed to identify the different beacons, so that separated RSS measurements can be associated to each independent beacon. For these two reasons, the beacons from all markers emit short IR light pulses in a scheduled emission slot according to a TDMA algorithm, depicted in Fig. 5.10.

In the TDMA algorithm, each beacon $b_{jm}$ emits in a specific time slot $\tau_{jm}$ of period $T$, denoted emission slot, which is followed by a second downtime slot in order to avoid emission overlap of sequential beacons. Beacons from the same marker are assigned with sequential emission/downtime slot pairs, as illustrated in Fig. 5.10a. A TDMA period is defined between two consecutive emission slots of the same beacon. This period defines the number of beacons and markers that can emit in the IR channel and is selected a priori. For example, the TDMA period illustrated in Fig. 5.10a is sixteen slots long, allowing the interaction of eight beacons (each one composed by a pair of time slots) in the IR channel. If each marker is composed by four beacons, the TDMA period in this example allows for the interaction of two markers in the IR channel. As depicted in Fig. 5.10b, the period of time that it takes for all the beacons of the same marker to emit is denoted as the super-slots of the marker. Marker $j$'s super-slot starts with the emission slot of its first beacon $\tau_j = \tau_{j1}$ and its period can be computed from the number of beacons

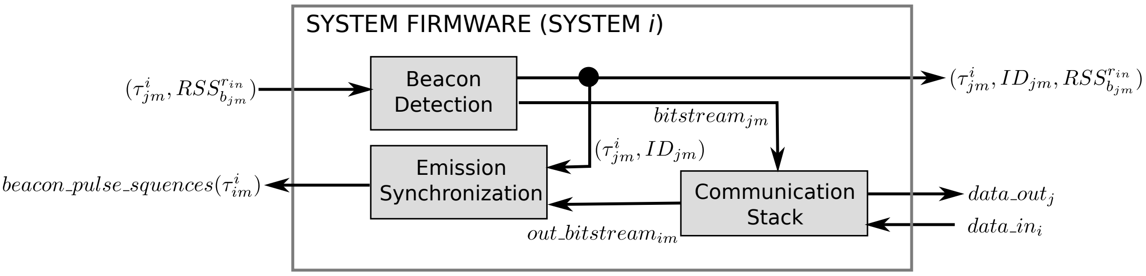

SYSTEM FIRMWARE (SYSTEM *i*)
$(\tau^i_{jm}, RSS^{r_{in}}_{b_{jm}})$
Beacon Detection
$(\tau^i_{jm}, ID_{jm}, RSS^{r_{in}}_{b_{jm}})$
$bitstream_{jm}$
$(\tau^i_{jm}, ID_{jm})$
Emission Synchronization
$beacon\_pulse\_sequences(\tau^i_{im})$
Communication Stack
$data\_out_j$
$data\_in_i$
$out\_bitstream_{im}$

Figure 5.11: IR-based positioning system transmission architecture and information flow, for a specific system $i$. The beacon detection algorithm extracts the beacon IDs and a communication bitstream from the set of RSSs of beacon $b_{jm}$ measured at receiver $r_{in}$, $RSS^{r_{in}}_{b_{jm}}$. Each beacon is detected at its beacon emission slot perceived in system $i$'s time scale, $\tau^i_{jm}$. The communication algorithm converts detected communication bitstreams into communicated data, or vice-versa. The emission synchronization algorithm controls marker $i$'s beacon digital pulse sequences, where it can encode bitstreams of communication data.

of marker $j$. The TDMA algorithm guarantees that markers $i$ and $j$ super-slots are always spaced by $\tau_{ij}$. This value is also selected in a priori for all markers. The details of this algorithm are presented in Appendix C.

In its emission slot, the beacon can either emit or not. This allows the creation of digital pulse sequences for each beacon, on top of which a digital communication algorithm can be implemented. This algorithm enables the transmission of the ID of each individual beacon and an additional bitstream to allow the communication between systems, as depicted in Fig. 5.11. The digital pulse sequence is composed by a set of '1's and '0's. A beacon transmits a '1' when it emits in its slot, and it transmits a '0' when it does not emit in its slot, as illustrated in Fig. 5.10b. Therefore, each beacon emission slot represents a bit in the beacon bitstream. The beacon ID is transmitted by making each beacon continuously transmit an unique digital pulse sequence identifying it. The beacon ID is unique and initially assigned independently of its association with a given marker. On the one hand, this allows the marker ID to be extracted by detecting just one of its beacons. On the other hand, the position of each beacon, uniquely recognized by its ID, is precisely specified in the marker frame. As visible in Fig. 5.11, the detected bitstreams of all marker $j$'s beacons are converted to data transmitted by system $j$, using a communication algorithm. Also, data from the system $i$ can also be transmitted by encoding it into the digital pulse sequences of all marker $i$'s beacons using an emission synchronization algorithm. The details of the communication and synchronization algorithms are presented in Appendices B and C. These algorithms operate exclusively over the IR channel. In this way, the positioning system only has to process information transmitted from neighboring systems. This allows for the distributed communication and localization between systems, which can potentially be scalable with the number of systems.

Each system $i$ collects the RSS of emitting beacons belonging to neighboring markers, measured at each receiver $r_{in}$, as depicted in Fig. 5.11. The RSS measurement of beacon $b_{jm}$, $RSS^{r_{in}}_{b_{jm}}$, is performed at the beacon emission slot, perceived in system $i$'s time scale as $\tau^i_{jm}$. A sufficiently high RSS measurement

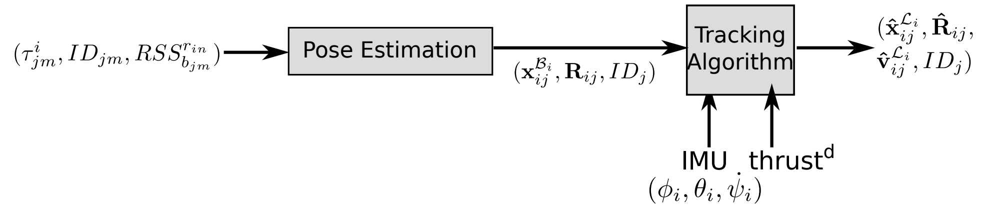

$(\tau^i_{jm}, ID_{jm}, RSS^{r_{in}}_{b_{jm}})$
Pose Estimation
$(\mathbf{x}^{\mathcal{B}_i}_{ij}, \mathbf{R}_{ij}, ID_j)$
Tracking Algorithm
$(\hat{\mathbf{x}}^{\mathcal{L}_i}_{ij}, \hat{\mathbf{R}}_{ij}, \hat{\mathbf{v}}^{\mathcal{L}_i}_{ij}, ID_j)$
IMU
thrust$^d$
$(\phi_i, \theta_i, \dot{\psi}_i)$

Figure 5.12: IR-based positioning system overall relative pose estimation architecture and information flow, for a specific UAV *i*. The detected beacon IDs and beacon emission times slots are used to identify the detected markers. The RSS measurements associated to the markers are used in a pose estimation algorithm that computes the marker relative pose to the UAV according to its specific 3D beacon layout. The UAV egomotion is fused with the previous estimates in order to acquire an accurate estimate while the UAV is moving.

at a specific time slot means the beacon has emitted in that time slot. A RSS measurement close to zero means that the beacon has not emitted in that time slot. The differentiation between an emitting and an non-emitting beacon at a specific time slot is implemented by thresholds, the details of which are presented in Appendix B. From the collected information, the beacons are detected by converting the digital pulse sequence detected at the beacon emission slot, to a beacon ID and an additional communication bitstream. The RSS measurements at that emission slot are then associated to the identified beacon. This information, along with the beacon ID and its emission slot, is given to a pose estimation algorithm, as shown in Fig. 5.12.

#### 5.1.2.2 3D relative pose and ID extraction of a marker

As described in the previous section, and as depicted in Fig. 5.12, each beacon ID, the associated beacon emission slot, and RSS measurements are used in the pose estimation algorithm. The algorithm extracts the relative pose and IDs for each neighboring UAV *j*, described by the ID and the relative pose of its onboard marker, expressed in the UAV *i*'s marker frame, $(\mathbf{x}^{\mathscr{B}_i}_{ij}, \mathbf{R}_{ij}, ID_j)$. In this work, the marker is assumed to be always horizontal in the UAV *i*'s marker frame. This means that the relative attitude of the marker is defined by the Euler angles $(\phi, \theta, \psi) = (0, 0, \psi_{ij})$. This simplification can lead to errors on the extracted marker poses every time the markers tilts. However, this system is envisioned for quadrotor vehicles, which usually move horizontally. Therefore, these errors can be assumed to be small. As previously discussed, the ID of marker *j* is taken from any of its detected beacon IDs (just one beacon is needed). The marker relative pose is then computed by fusing the RSS measurements associated to the multiple detected beacons belonging to marker *j*, with known position association within the marker frame, taken at several receivers of UAV *i*.

As depicted in Fig. 5.13, the RSS of beacon $b_{jm}$ measured at receiver $r_{in}$ can be modeled from the physical model describing the decay of the light energy density as a function of the square of the range between the beacon and the receiver $d^{r_{in}}_{b_{jm}}$, and the model for the received light absorption, proportional to the incidence angle between the light ray and the receiver's heading, $\theta^{r_{in}}_{b_{jm}}$. This model can be written

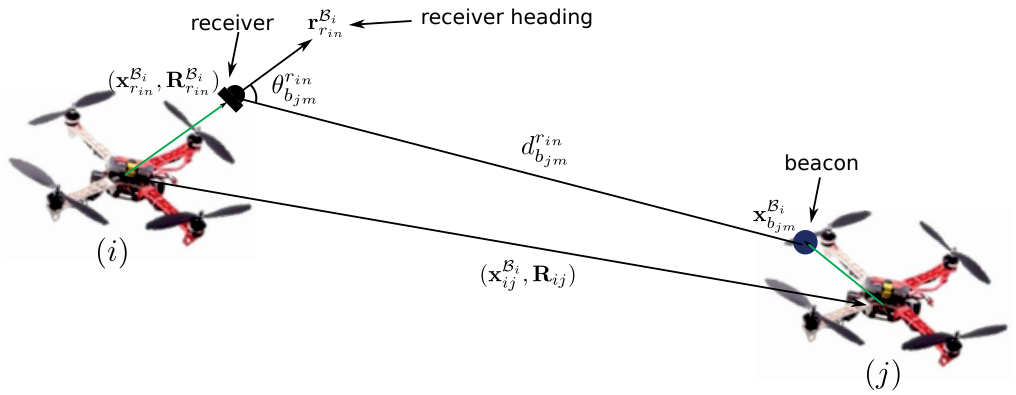

receiver
$\mathbf{r}^{\mathcal{B}_i}_{r_{in}}$
receiver heading
$(\mathbf{x}^{\mathcal{B}_i}_{r_{in}}, \mathbf{R}^{\mathcal{B}_i}_{r_{in}})$
$\theta^{r_{in}}_{b_{jm}}$
$d^{r_{in}}_{b_{jm}}$
beacon
$\mathbf{x}^{\mathcal{B}_i}_{b_{jm}}$
$(i)$
$(\mathbf{x}^{\mathcal{B}_i}_{ij}, \mathbf{R}_{ij})$
$(j)$

Figure 5.13: Model for the RSS of a beacon $b_{jm}$ measured by a receiver $r_{in}$. The beacon is placed at a position in marker $j$'s frame $\mathbf{x}^{\mathscr{B}_j}_{b_{jm}}$ and the receiver is placed at a pose in marker $i$'s frame $(\mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{R}^{\mathscr{B}_i}_{r_{in}})$. Marker $j$ has a pose in marker $i$'s frame $(\mathbf{x}^{\mathscr{B}_i}_{ij}, \mathbf{R}_{ij})$. The measured RSS can be modeled from the distance between the receiver and the beacon $d^{r_{in}}_{b_{jm}}$ and the incidence angle of the respective light ray to the receiver heading $\theta^{r_{in}}_{b_{jm}}$.

as follows:

$$E^{r_{in}}_{b_{jm}} = C_{b_{jm}} \frac{f_{r_{in}}\left(\cos(\theta^{r_{in}}_{b_{jm}})\right)}{(d^{r_{in}}_{b_{jm}})^2} = C_{b_{jm}} \frac{f_{r_{in}}\left(< \mathbf{x}^{\mathscr{B}_i}_{b_{jm}} - \mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{r}^{\mathscr{B}_i}_{r_{in}} >\right)}{||\mathbf{x}^{\mathscr{B}_i}_{b_{jm}} - \mathbf{x}^{\mathscr{B}_i}_{r_{in}}||^2}, \tag{5.11}$$

where $E^{r_{in}}_{b_{jm}}$ is the light energy of the beacon $b_{jm}$ absorbed by the receiver $r_{in}$, $C_{b_{jm}}$ is an intensity gain associated to each beacon, and $f_{r_{in}}$ is the receiver's absorption coefficient function. In this work, $f_{r_{in}}$ is defined with two terms $f_{r_{in}}(\cos(\theta)) = \alpha_{r_{in}} \cos(\theta) + \beta_{r_{in}} \sqrt{\cos(\theta)}$, as both were described in [86,92]. The parameter $C_{b_{jm}}$ is used to address the different beacon emission intensities caused by different electrical current passing through each beacon. The computation of its value is further discussed in Section 5.1.2.4.

The absorbed light energy passes through an amplification stage to generate the respective RSS measurement $RSS^{r_{in}}_{b_{jm}} = g_{r_{in}}(E^{r_{in}}_{b_{jm}})$. In this work, the absorbed light coming from a beacon pulse is modulated with a known frequency (the same for all the beacons), as described in more details in Section 5.1.2.5. This allows the implementation of a cascade amplification algorithm, as in [91, 92], in order to improve the dynamic range of the system. This method consists of applying a series of amplification modules to the absorbed light energy signal. After each amplification module, a filtering module is added in order to reduce the noise corresponding to all other frequencies other than that corresponding to the chosen modulation frequency. The output of the filtering module is fed to the next amplification module and so on. The RSS measurement is formed by performing a cumulative aggregation of the output of all the filtering stages. The cascade amplification algorithm allows absorbed light energy originated from far away emission pulses to stand out from environment noise or noise caused by the emissions of other devices that are not part of this relative positioning system.

The algorithm described above is depicted in Fig. 5.14a-c. As observed in Fig. 5.14b and c, each amplification module output saturates if the input signal is too high. This causes the final output signal

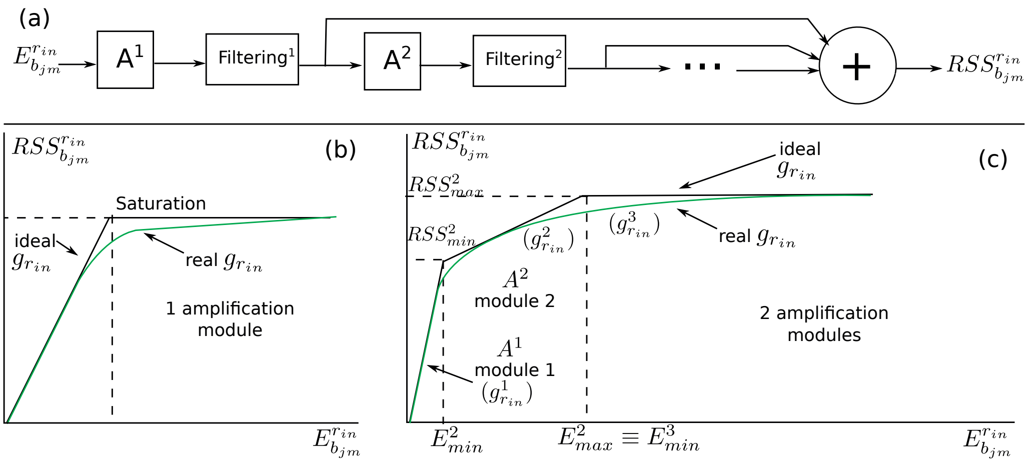


Figure 5.14: Cascade amplification algorithm for each receiver. (a) The RSS measurement ($RSS^{r_{in}}_{b_{jm}}$) is obtained from the cumulative aggregation of multiple amplification and filtering modules applied to the absorbed light energy signal ($E^{r_{in}}_{b_{jm}}$). (b) Each amplification module multiplies the input signal by $A$. If the input signal becomes too large, the module output saturates. (c) The cumulative aggregation of multiple amplification module outputs. Note the non-linear behavior of the real function $g_{r_{in}}$, in contrast to the ideal piece-wise linear function. Therefore, $g_{r_{in}}$ is modeled by a piece-wise non-linear function, with each section $g^s_{r_{in}}$ defined within a set of $E^{r_{in}}_{b_{jm}}$ or $RSS^{r_{in}}_{b_{jm}}$ values.

to be related to the input signal in piece-wise sections. For this reason, a piece-wise function is used to model $g_{r_{in}}$. Each segment $s$ of this piece-wise function, $g^s_{r_{in}}$, is defined for values of $E^{r_{in}}_{b_{jm}} \in [E^s_{min} E^s_{max}]$ (see Fig. 5.14c). Therefore, the RSS of beacon $b_{jm}$ measured at receiver $r_{in}$ is defined as follows:

$$RSS^{r_{in}}_{b_{jm}} = g^s_{r_{in}}(E^{r_{in}}_{b_{jm}}(\mathbf{r}^{\mathscr{B}_i}_{r_{in}}, \mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{x}^{\mathscr{B}_i}_{b_{jm}})) \quad \forall E^{r_{in}}_{b_{jm}} \in [E^s_{min} E^s_{max}]. \tag{5.12}$$

Given the non-linear behavior of the real function $g_{r_{in}}$, each segment of this function is modeled by $g^s_{r_{in}}(E) = \alpha^s_{r_{in}} + \beta^s_{r_{in}} E + \gamma^s_{r_{in}} E^2$. Note that $g_{r_{in}}$ is continuous, and continuously differentiable. Therefore, the parameters of all piece-wise functions $g^s_{r_{in}}(E)$ are also computed in order to obtain a $g_{r_{in}}$ which is continuous, and continuously differentiable. Since $g_{r_{in}}$ is also injective, it is also possible to define $g^s_{r_{in}}$ for values of $RSS^{r_{in}}_{b_{jm}} \in [RSS^s_{min} RSS^s_{max}]$. This is useful when choosing which segment of $g_{r_{in}}$ to use in the localization algorithms, since $E^{r_{in}}_{b_{jm}}$ is not directly observable, but $RSS^{r_{in}}_{b_{jm}}$ is. Functions $f_{r_{in}}$ and $g_{r_{in}}$ parameters are found through a calibration process explained in Section 5.1.2.4.

Each RSS measurement is related to marker $j$'s relative pose by noting that the beacon $b_{jm}$ position in UAV $i$'s marker frame can be described as (see Fig. 5.13):

$$\mathbf{x}^{\mathscr{B}_i}_{b_{jm}} = \mathbf{x}^{\mathscr{B}_i}_{ij} + \mathbf{R}_{ij} \mathbf{x}^{\mathscr{B}_j}_{b_{jm}}. \tag{5.13}$$

where $\mathbf{R}_{ij} = \mathbf{R}_{\psi_{ij}}$ since $\phi_{ij}$ and $\theta_{ij}$ are considered zero, as previously mentioned. Eq. 5.12, together with Eqs. 5.11 and 5.13, gives a direct mathematical correspondence between each RSS measurement and $(\mathbf{x}^{\mathscr{B}_i}_{ij}, \mathbf{R}_{ij})$. It is intuitive that at least four RSS measurements are required to form an equation system able to compute $(\mathbf{x}_{ij}^{\mathscr{B}_i}, \mathbf{R}_{ij})$, since it is composed of four variables: $\mathbf{x}_{ij}^{\mathscr{B}_i} = (x_{ij}^{\mathscr{B}_i}, y_{ij}^{\mathscr{B}_i}, z_{ij}^{\mathscr{B}_i})$ and $\psi_{ij}$. More concretely, the minimally required four RSS measurements must come from a set of receivers with at least three linearly independent headings $\mathbf{r}_{r_{in}}^{\mathscr{B}_i}$ in order to provide information for the three spatial dimensions of a specific beacon. Additionally, these four RSS measurements must also come from at least two different beacons belonging to the same marker in order to provide the marker attitude information.

The desired marker $j$'s relative pose in the UAV $i$'s marker frame $(\mathbf{x}_{ij}^{\mathscr{B}_i}, \mathbf{R}_{ij})$ is then estimated using an Extended Kalman Filter algorithm that fuses multiple individual RSS measurements. The estimation state vector consists of the marker $j$'s 3D relative position and the 2D relative horizontal attitude $(\mathbf{x}_{ij}^{\mathscr{B}_i}, \psi_{ij})$. The estimate uncertainty is defined by a four dimensional covariance matrix. For each set of RSS measurements acquired by the system at the current time step, a prediction step is first applied to the estimate computed at the previous time step. The prediction step consists in a simple random walk motion model, given that there is no prior knowledge about marker $j$'s movement. The motion model is as follows:

$$\begin{bmatrix} \mathbf{x}_{ij}^{\mathscr{B}_i}(k+1) \\ \psi_{ij}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{x}_{ij}^{\mathscr{B}_i}(k) \\ \psi_{ij}(k) \end{bmatrix} + \gamma_{s_{ij}}^{\mathscr{B}_i}. \tag{5.14}$$

The model noise $\gamma_{s_{ij}}^{\mathscr{B}_i}$ is modeled by a four dimensional random variable with a zero mean normal distribution with a standard deviation reflecting how much the relative pose can vary between the two markers, which is related to the maximum linear and angular speeds considered for the UAVs. Afterwards, the acquired RSS measurements are fused together in an update step to improve the estimate at the current time step. The update step uses for each individual RSS measurement the measurement model defined using Eq. (5.12) as follows:

$$RSS_{b_{jm}}^{r_{in}}(k) = g_{r_{in}}^{s}(E_{b_{jm}}^{r_{in}}(\mathbf{r}_{r_{in}}^{\mathscr{B}_i}, \mathbf{x}_{r_{in}}^{\mathscr{B}_i}, \mathbf{x}(k)_{b_{jm}}^{\mathscr{B}_i})) + \eta_{s_{ij}}, \tag{5.15}$$

where $E_{b_{jm}}^{r_{in}}$ is defined in Eq. (5.11) and $\mathbf{x}_{b_{jm}}^{\mathscr{B}_i}(k)$ is related to the estimate $(\mathbf{x}_{ij}^{\mathscr{B}_i}(k), \psi_{ij}(k))$ through Eq. (5.13). The model noise for each RSS measurement $\eta_{s_{ij}}$ is modeled by a scalar random variable with a zero mean normal distribution with a standard deviation reflecting how the RSS measurement taken from a static receiver varies while detecting a static beacon. Refer to Appendix A for details on the implementation of this filter.

In this work, not all RSS measurements are selected to be fused in the update step. RSS measurements with a value smaller than a certain threshold are discarded. Additionally, the model representing the absorption of the light energy by the receiver, defined in Eq. (5.11), indicates that the light energy is better absorbed when the light ray is aligned with the receiver heading, corresponding to a zero incidence angle. According to the definition of function $f_{r_{in}}$ used in that model, when the incidence angle of the light increases the light energy absorbed by the receiver decreases. If the incidence angle becomes too large, it is most likely that the light energy absorbed by the receiver will be related to rays that come from the environment or IR ray reflections, which would generate RSS measurements substantially different from the ones predicted by the model mentioned above. Therefore, receivers for which the incidence angle of the light is larger than a threshold are not used. In this work, an incidence angle threshold of 72° was used, since the measured $f_{r_{in}}$ functions indicated that only 30% of the light energy or less was absorbed at the receiver for incidence angles above this value. Finally, since the receivers with smallest incidence angles to the emission source measure the largest RSSs, only the six largest RSS measurements taken for each beacon are used at each update step for a given marker. In this way, the RSS measurements with minimal distortions will be selected, minimizing the effect of environment reflections on the relative pose estimation algorithm.

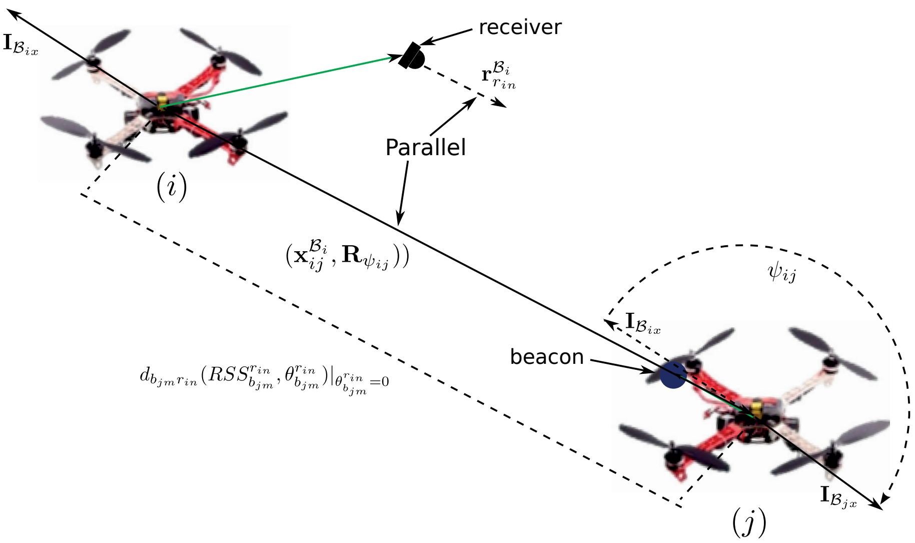
Despite the RSS measurement pruning mentioned above, the estimation algorithm always verifies that the minimal number of RSS measurements have been gathered. If after the RSS measurement pruning such verification fails, the algorithm will select additional RSS measurements to achieve the minimal set required. The RSS measurements are selected so that three non-collinear receiver sets are used in the estimation. In case this is not achieved, the estimation will diverge in the 3D space. Therefore, the algorithm selects the largest RSS measurements that have not been selected yet, associated to receivers with headings that are non-collinear to the ones from the already selected receivers. No additional selection is made if only one beacon from the marker is detected. This situation might happen when the neighboring UAV is too far, or due to beacon occlusions. This will cause the attitude estimation to diverge, but the position estimation can still be used.

When marker $j$'s beacons are detected for the first time by the UAV $i$'s onboard system, the estimate is initialized using the receiver that measured the largest RSS belonging to one of the beacons of marker $j$. The relative pose of marker $j$ is initialized as depicted in Fig. 5.15. Basically, the marker relative pose is placed with a relative bearing and elevation that define the heading direction of the chosen receiver. The marker relative range to UAV $i$ is computed using the RSS light decay model through space described in Eq. (5.11) with a zero incidence angle, for the chosen RSS measurement. The marker attitude is set to be such that the beacon to which the used RSS measurement belongs to is facing UAV $i$. The previous initialization process provides only an approximated estimate of marker $j$'s relative pose. However, the estimate is sufficiently close so that it can converge to its final value with just a couple of additional measurements. The covariance matrix defining the estimate uncertainty is initialized as an identity matrix.

#### 5.1.2.3 Tracking algorithm

Similarly to the camera-based system, each acquired marker $j$ relative pose measurement is forwarded to a tracker. This tracker keeps a list of observed markers, each described by its ID, its estimated position and velocity, and its attitude in the UAV $i$'s flying frame $(\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}, \hat{\mathbf{R}}_{ij}, \hat{\mathbf{v}}_{ij}^{\mathscr{L}_i}, ID_j)$. Each marker is tracked independently at discrete time intervals of $\Delta t$ seconds, with the algorithm described in Section 4.2.1. Marker $j$'s motion model is defined by Eq. (4.4), which uses the UAV $i$'s egomotion expressed in the UAV flying frame that in turn can be computed as described in Section 4.2.1.

For each measurement of marker $j$, generated by the previous described estimation algorithm, the tracker updates the 3D relative inter-vehicle localization and velocity estimations of the respective marker



$\mathbf{I}_{\mathcal{B}_{ix}}$
receiver
$\mathbf{r}^{\mathcal{B}_i}_{r_{in}}$
Parallel
$(i)$
$(\mathbf{x}^{\mathcal{B}_i}_{ij}, \mathbf{R}_{\psi_{ij}}))$
$\psi_{ij}$
$\mathbf{I}_{\mathcal{B}_{ix}}$
beacon
$d_{b_{jm}r_{in}}(RSS^{r_{in}}_{b_{jm}}, \theta^{r_{in}}_{b_{jm}})|_{\theta^{r_{in}}_{b_{jm}}=0}$
$\mathbf{I}_{\mathcal{B}_{jx}}$
$(j)$

Figure 5.15: IR-based positioning system relative pose estimation initialization. The recently detected marker $j$ is placed with a relative range computed from the RSS of the detected beacon $b_{jm}$ measured by receiver $r_{in}$, $d^{r_{in}}_{b_{jm}}$, and with a relative bearing and elevation defined by the receiver heading. Marker $j$'s relative attitude $\mathbf{R}_{ij}$, defined by $\psi_{ij}$ ($\mathbf{R}_{\psi_{ij}}$), is computed so that the detected beacon faces marker $i$'s frame. The angle $\psi_{ij}$ is computed from the angle between the $x$ axes of the two marker frames ($\mathbf{I}_{\mathscr{B}_{ix}}$ and $\mathbf{I}_{\mathscr{B}_{jx}}$).

on the tracking list. First, the measurement is converted from the marker to the flying frame, using the current $\phi_i$ and $\theta_i$ values acquired by the UAV $i$'s IMU sensor. Marker $j$'s pose in the flying frame is then updated with the measurement model defined in Eq. (4.5), where $\eta^{\mathscr{L}_i}_{ij}$ is the term defining the uncertainty of the relative positioning sensor measurements for each observed position of marker $j$ relative to the UAV $i$'s flying frame. In this case, $\eta^{\mathscr{L}_i}_{ij}$ is defined by the uncertainty matrix computed by the previous estimation algorithm, described by Eqs. (5.14) and (5.15). Note that this uncertainty matrix has to be first transformed into the UAV $i$'s flying frame using the current rotation of the UAV, before being used in the measurement model.

When the marker leaves the sensor detection area, the tracking algorithm still keeps its estimation for a while before dropping it. Once the marker gets back to the detection area the estimation process is re-initialized and resumed. Since the developed IR-based positioning sensor is provided with a larger FOV than the developed camera-based sensor, the marker only leaves the sensor detection area mostly when it is beyond the detection range. However, the vertical FOV constraints can still be breached when the detected marker moves sufficiently under the UAV to enter an occlusion zone or incur into interference effect with the rotating propellers, as detailed in Section 5.1.2.5.

#### 5.1.2.4 Calibration algorithm

As described in the previous sections, the model for the RSS measurement used in the pose estimation algorithm requires the knowledge of two functions, $f_{r_{in}}$ and $g_{r_{in}}$, for each receiver $r_{in}$. Function $f_{r_{in}}$

is the absorption coefficient function, describing how much receiver $r_{in}$ light absorption capabilities decrease in function of the incidence angle of the respective light rays. The function $g_{r_{in}}$ describes how the amplification stage implemented on receiver $r_{in}$ acquires the RSS measurement from the light energy absorbed by the receiver. These functions are defined by a set of parameters: two parameters for function $f_{r_{in}}$ ($\alpha_{r_{in}}$ and $\beta_{r_{in}}$), and three parameters for each segment $s$ of function $g_{r_{in}}$ ($\alpha^s_{r_{in}}$, $\beta^s_{r_{in}}$, and $\gamma^s_{r_{in}}$). Additionally, as described in Eq. (5.11), the parameter $C_{b_{jm}}$ is also used to address the different beacon emission intensities caused by different electrical current passing through each beacon. Finally, the model also requires the knowledge of the position and the heading, on UAV $i$, of the receiver $r_{in}$ that acquired the RSS measurement $(\mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{r}^{\mathscr{B}_i}_{r_{in}})$. The correct values for all the previous parameters have to be known by the estimation algorithm in order for it to provide the correct estimates. Their values are computed using a calibration algorithm, similar to [44] but using a different method.

The calibration algorithm is summarized in Algorithm 1. Firstly, a single beacon $b_{(j^*)m}$ is chosen, and its parameter $C_{b_{(j^*)m}}$ is set to one. Secondly, each sensor $i$ is calibrated independently. The calibration starts by having each receiver $r_{in}$ of the system measuring the RSS of the beacon $RSS^{r_{in}}_{b_{(j^*)m}}$ at different relative positions between the system and the beacon, $\mathbf{x}^{\mathscr{B}_i}_{b_{(j^*)m}}$. The value of $\mathbf{x}^{\mathscr{B}_i}_{b_{(j^*)m}}$ can be measured by hand or with a MCS for a better calibration accuracy. The set of relative positions must include zero and non-zero incidence angles of the light ray on the receiver at different ranges between the beacon and each receiver, as shown in Fig. 5.16a. After the $(\mathbf{x}^{\mathscr{B}_i}_{b_{(j^*)m}}, RSS^{r_{in}}_{b_{(j^*)m}})$ data pairs are collected, the parameters of the functions $f_{r_{in}}$ and $g_{r_{in}}$ of each receiver, together with the position and heading of the receiver on the UAV $i$ $(\mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{r}^{\mathscr{B}_i}_{r_{in}})$. are calibrated using a two-step calibration algorithm. This calibration algorithm is conducted independently for each receiver $r_{in}$.

Firstly, initial values are given to all the parameters to be calibrated. Then, the first step of the algorithm calibrates the parameters of the functions $f_{r_{in}}$ and $g_{r_{in}}$. This is achieved by means of an optimization algorithm that computes the functions $g_{r_{in}}$ and $f_{r_{in}}$ parameters in order to minimize the errors between the $RSS^{r_{in}}_{b_{(j^*)m}}$ of each data pair and $g_{r_{in}}$, as follows:

$$\min_{\mathbf{a}_{in}} \sum_{l=1}^{L} RSS^{r_{in}}_{b_{(j^*)m}}(l) - g_{r_{in}}(E^{r_{in}}_{b_{(j^*)m}}(\mathbf{r}^{\mathscr{B}_i}_{r_{in}}, \mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{x}^{\mathscr{B}_i}_{b_{(j^*)m}}(l), \mathbf{a}_{in})), \tag{5.16}$$

where $l$ is the data pair number, $(\mathbf{x}^{\mathscr{B}_i}_{b_{(j^*)m}}(l), RSS^{r_{in}}_{b_{(j^*)m}}(l))$ is the collected data pair, and $\mathbf{a}_{in} = (\alpha^s_{r_{in}}, \beta^s_{r_{in}}, \gamma^s_{r_{in}}, \alpha_{r_{in}}, \beta_{r_{in}}) \forall s$ corresponds to functions $g_{r_{in}}$ and $f_{r_{in}}$ parameters, with $s$ representing the section number of the piece-wise function $g_{r_{in}}$. Note that $E^{r_{in}}_{b_{(j^*)m}}$ is predicted using Eq. (5.11) with the measured $\mathbf{x}^{\mathscr{B}_i}_{b_{(j^*)m}}(l)$. This optimization algorithm is divided into two sub-steps. The first sub-step ($optimize_g$ in Algorithm 1) optimizes function $g_{r_{in}}$'s parameters with a polynomial leasts squares algorithm using the gathered data pairs. The second sub-step ($optimize_f$ in Algorithm 1) optimizes function $f_{r_{in}}$'s parameters with a non-linear least squares algorithm in order to minimize the remaining errors between the $RSS^{r_{in}}_{b_{(j^*)m}}$ of each data pair and $g_{r_{in}}$ (now using the parameters of $g_{r_{in}}$ computed in the first sub-step). The two previous sub-steps are iterated several times in order to achieve better results. Both sub-steps use the initial values given to $(\mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{r}^{\mathscr{B}_i}_{r_{in}})$.

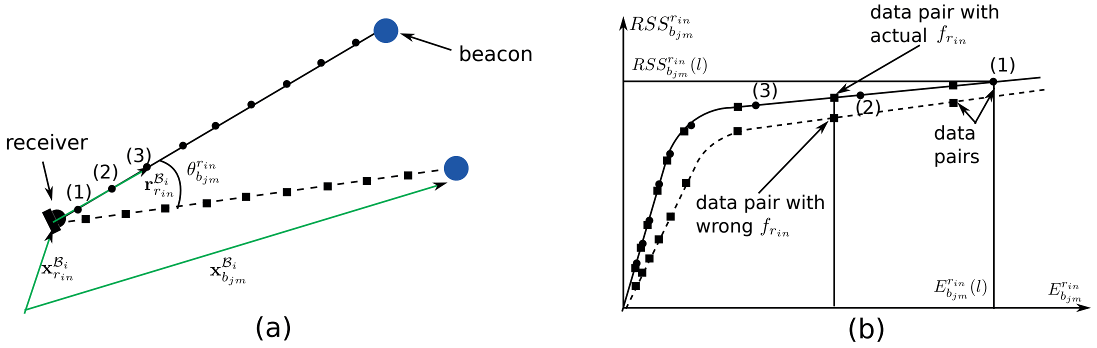


Figure 5.16: Illustration of the IR-based system calibration procedure. (a) RSS of beacon $b_{jm}$ measured by receiver $r_{in}$ at different relative positions $\mathbf{x}_{b_{jm}}^{\mathscr{B}_i}$. The circles and squares correspond to $(\mathbf{x}_{b_{jm}}^{\mathscr{B}_i}, RSS_{b_{jm}}^{r_{in}})$ pairs gathered at relative positions with zero and non zero incidence angle $\theta_{b_{jm}}^{r_{in}}$, respectively. (b) Function $g_{r_{in}}$ (solid line) computed by the calibration algorithm. All $(E_{b_{jm}}^{r_{in}}, RSS_{b_{jm}}^{r_{in}})$ pairs should lay on this line. However, deviations can occur for measurements taken at positions with non-zero $\theta_{b_{jm}}^{r_{in}}$ (squares on the dashed line) if the parameters of $f_{r_{in}}$ were wrongly chosen.

If the previous optimization algorithm is successful, all the data pairs formed by the measured $RSS_{b_{(j^*)m}}^{r_{in}}$ and the $E_{b_{(j^*)m}}^{r_{in}}$ predicted using Eq. (5.11) will fall on the curve defined by function $g_{r_{in}}$. This is shown in Fig. 5.16b for a zero and a non-zero incidence angle of the light ray on the receiver. If a problem occurred, the positions of the data pairs will deviate from this curve. For example, if the parameters of function $f_{r_{in}}$ are wrongly computed, data pairs corresponding to non-zero incidence angles will not match the curve defined by function $g_{r_{in}}$, as shown in Fig. 5.16b with the dashed line. Note that these deviations will also occur if errors exist in the $(\mathbf{x}_{r_{in}}^{\mathscr{B}_i}, \mathbf{r}_{r_{in}}^{\mathscr{B}_i})$ parameters.

Usually, the values for $(\mathbf{x}_{r_{in}}^{\mathscr{B}_i}, \mathbf{r}_{r_{in}}^{\mathscr{B}_i})$ should come directly from their theoretical values chosen during sensor design, and indeed those are the values to which these parameters were previously set with. However, deployment process inaccuracies can lead to different deployed headings. Fig. 5.17 illustrates an example of this problem. In this example, two relative positions between the beacon and the receiver with symmetric incidence angle $\theta_{b_{(j^*)m}}^{r_{in}}$ would result in the same light energy absorbed by the receiver. This can be predicted using the $f_{r_{in}}$ and the predicted receiver heading. However, if the actual receiver heading is different than the predicted one (e.g., rotated by $a$ degrees as illustrated in Fig. 5.17), the absorbed light will now be different at the supposedly symmetric relative positions. This will again generate deviations of the $(E_{b_{(j^*)m}}^{r_{in}}, RSS_{b_{(j^*)m}}^{r_{in}})$ pairs from the curve defined by function $g_{r_{in}}$. The second step of the calibration algorithm ($optimize_{\mathbf{r}}$ in Algorithm 1) optimizes the $\mathbf{r}_{r_{in}}^{\mathscr{B}_i}$ parameter in order to further minimize those deviations, as follows:

$$\min_{\mathbf{r}_{r_{in}}^{\mathscr{B}_i}} \sum_{l=1}^{L} RSS_{b_{(j^*)m}}^{r_{in}}(l) - g_{r_{in}}(E_{b_{(j^*)m}}^{r_{in}}(\mathbf{r}_{r_{in}}^{\mathscr{B}_i}, \mathbf{x}_{r_{in}}^{\mathscr{B}_i}, \mathbf{x}_{b_{(j^*)m}}^{\mathscr{B}_i}(l), \mathbf{a}_{in})). \tag{5.17}$$

Note that the functions $g_{r_{in}}$ and $f_{r_{in}}$ computed in the first step of the calibration algorithm are used for this second step. The two steps described in Eqs. (5.16) and (5.17) are repeated several times in

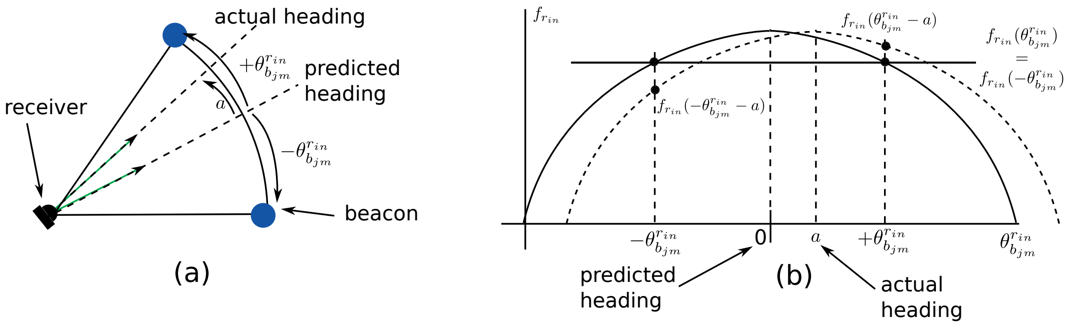

actual heading
$+\theta^{r_{in}}_{b_{jm}}$
predicted heading
receiver
$a$
$-\theta^{r_{in}}_{b_{jm}}$
beacon
(a)
$f_{r_{in}}$
$f_{r_{in}}(\theta^{r_{in}}_{b_{jm}} - a)$
$f_{r_{in}}(\theta^{r_{in}}_{b_{jm}})$
$=$
$f_{r_{in}}(-\theta^{r_{in}}_{b_{jm}})$
$f_{r_{in}}(-\theta^{r_{in}}_{b_{jm}} - a)$
$-\theta^{r_{in}}_{b_{jm}}$
0
$a$
$+\theta^{r_{in}}_{b_{jm}}$
$\theta^{r_{in}}_{b_{jm}}$
predicted heading
(b)
actual heading

Figure 5.17: Absorption coefficient prediction errors due to deployment process inaccuracies. (a) Two relative positions between a beacon and a receiver which have symmetric incidence angles $\theta^{r_{in}}_{b_{jm}}$ with respect to the predicted receiver heading, but non-symmetric with respect to the actual receiver heading (which is rotated $a$ degrees away from the predicted heading). (b) Predicted (continuous curve) and actual (dashed curve) absorption coefficient functions, $f_{r_{in}}(\theta)$ and $f_{r_{in}}(\theta - a)$ respectively, and the respective absorption coefficients at the two relative positions that were considered.

order to achieve better calibration results. The receiver and beacon positions in the respective UAV frames $(\mathbf{x}^{\mathscr{B}_i}_{r_{in}}, \mathbf{x}^{\mathscr{B}_j}_{b_{(j^*)m}})$ are taken directly from the theoretical values chosen during sensor design and are not optimized, since it was observed that small position errors caused by deployment process inaccuracies only produce small errors.

The previous calibration algorithm used one single beacon. All parameters of functions $f_{r_{in}}$ and $g_{r_{in}}$ were computed using $C_{b_{(j^*)m}} = 1$. Different beacons would have a different $C_{b_{jm}}$. Those intensity gains are computed with an experiment where the other beacons are placed in front of a specific receiver from a specific system $r_{(i^*)(n^*)}$ at a fixed known relative position. The value for $C_{b_{jm}}$ is computed by comparing the acquired RSS measurements with the measurements obtained using the first beacon at the same relative position.

In this calibration algorithm, the parameters of the functions $f_{r_{in}}$ and $g_{r_{in}}$, and receiver heading in the marker $i$'s frame $\mathbf{r}^{\mathscr{B}_i}_{r_{in}}$ are all computed independently for each receiver $r_{in}$. The beacon intensity gains $C_{b_{jm}}$ are computed for each beacon $b_{jm}$, independently from the individual systems. This algorithm allows the use of receivers with different light absorption and amplification characteristics, which might not be accurately known. Furthermore, this algorithm allows the deployment process to be less strict, as deployment inaccuracies can be corrected.

#### 5.1.2.5 System design

For the system hardware design, the IR beacons and receivers positions had to be first selected. Similarly to the camera-based positioning system, the position of the beacons and receivers were selected taking into account occlusions caused by the UAV body on the receivers and beacons. To set up the localization marker, any number $N_b$ of beacons can be accommodated by the algorithms presented above. However, four beacons are considered in this work, each one placed on the edge of each quadrotor's arms, as shown in Fig. 5.18. This configuration ensures that at least two beacons are visible from any considered relative

**Algorithm 1** IR-based positioning system calibration algorithm

```
procedure IRBasedSystemCalibration
    $select\ b_{(j^*)m}$;
    $C_{b_{(j^*)m}} \leftarrow 1$;
    ForEach system $i$
        ForEach $r_{in}$
            $collect(\mathbf{x}^{\mathscr{B}_i}_{b_{(j^*)m}}, RSS^{r_{in}}_{b_{(j^*)m}})$
            $\mathbf{a}_{in} \leftarrow (\alpha^{s,0}_{r_{in}}, \beta^{s,0}_{r_{in}}, \gamma^{s,0}_{r_{in}}, \alpha^{0}_{r_{in}}, \beta^{0}_{r_{in}})\forall s$
            $\mathbf{r}^{\mathscr{B}_i}_{r_{in}} \leftarrow \mathbf{r}^{\mathscr{B}_i,0}_{r_{in}}$
            Iterate
                Iterate
                    $(\alpha^{s}_{r_{in}}, \beta^{s}_{r_{in}}, \gamma^{s}_{r_{in}})\forall s \leftarrow optimize_g(E^{r_{in}}_{b_{(j^*)m}}, RSS^{r_{in}}_{b_{(j^*)m}}, \mathbf{a}_{in}, \mathbf{r}^{\mathscr{B}_i}_{r_{in}})$
                    $(\alpha_{r_{in}}, \beta_{r_{in}}) \leftarrow optimize_f(E^{r_{in}}_{b_{(j^*)m}}, RSS^{r_{in}}_{b_{(j^*)m}}, \mathbf{a}_{in}, \mathbf{r}^{\mathscr{B}_i}_{r_{in}})$
                $\mathbf{r}^{\mathscr{B}_i}_{r_{in}} \leftarrow optimize_{\mathbf{r}}(E^{r_{in}}_{b_{(j^*)m}}, RSS^{r_{in}}_{b_{(j^*)m}}, \mathbf{a}_{in}, \mathbf{r}^{\mathscr{B}_i}_{r_{in}})$
    ForEach $b_{jn} \neq b_{(j^*)n}$
        $collect(\mathbf{x}^{\mathscr{B}(i^*)}_{b_{jm}}, RSS^{r(i^*)(n^*)}_{b_{jm}})$
        $C_{b_{jm}} \leftarrow RSS^{r(i^*)(n^*)}_{b_{jm}} / RSS^{r(i^*)(n^*)}_{b_{(j^*)m}}$
```

poses, thus enabling the previous described pose estimation algorithm to work properly.

IR receiver placement also considers the weight of the positioning system, in order to allow its deployment on lightweight UAVs. The IR receivers are placed on a receiver station, as illustrated in Fig. 5.18b and Fig. 5.20b, in order to minimize the amount of wiring that has to be used. This station is considered to be an half sphere with 10 *cm* of diameter. Each receiver is placed on this sphere independently. The actual position of the receivers is on the border of the receiver station, as illustrated in Fig. 5.18b. The station has eight receiver sections, each one with one receiver pointing horizontally and another pointing 45° up, as depicted in Fig. 5.20b. One receiver station is placed on top of the UAV body pointing upwards.

For quadrotor vehicles, the main sources of occlusion to be considered for beacons and receivers are the propellers. Signal interference can be generated if the light rays between the beacon and the receiver collide with a propeller. This happens when the beacon that is being detected is lower than the UAV body to which the receiver taking the measurement is associated to, as depicted in Fig. 5.18b. When the propellers are turned off, this interference can result in a complete occlusion. However, this case is not considered because when the propellers are off the UAV is typically on the floor, and there are no markers to be detected lower than the UAV body. However, when the UAV is in the air, interference will occur when the moving propeller briefly passes through the light ray between the beacon and the receiver. Experiments showed that this interference is mainly caused by the propellers of the UAV that also carry the receivers. This interference does not preclude the detection of the beacons of another UAV but generates a RSS decay which can be as different as up to 30% of the signal, distorting the relative pose estimations. Another source of occlusion are the UAV arms, which therefore generate sensory blind spots.

The previous occlusions and interferences can be mitigated using a spacer, as illustrated in Fig. 5.18b. This spacer allows a single receiver station placed on top of the UAV body to ensure a lower bound in terms of vertical FOV ($\theta_{v_{low}}$) without occlusions. The spacer height $l_{spacer}$ can be computed as:

$$l_{spacer} = l_{arm}\tan(\theta v_{low}) - l_{body}, \tag{5.18}$$

where $l_{arm}$ is the length of the UAV's arms and $l_{body}$ is the UAV's body height as depicted in Fig. 5.18b. In order to consider the occlusion generated by the propellers, Eq. (5.18) has to be modified to incorporate the propeller length and its height with respect to the UAV arms. However, note that as the $\theta_{v_{low}}$ increases so does $l_{spacer}$. Similarly to the camera-based positioning system, when $l_{spacer}$ increases the system approaches to an inverted pendulum system, which is theoretically unstable. Therefore, this value should be kept small. In this work, there is no study of how large $l_{spacer}$ can be. But if $\theta_{v_{low}}$ is sufficiently large (for example to detect ground vehicles), a second receiver station placed on the bottom of the UAV body pointing downwards should be considered, as presented in Fig. 5.18b. The top receiver station would have a free line of sight to markers that are above the UAV, and the bottom receiver station would have a free line of sight to markers that are bellow the UAV. Due to the height difference between receiver stations (see Fig. 5.18b) and the non-zero size of the UAV body, there would still exist a sensor blind spot near the UAV, around the plane formed by the its arms. However, the marker would have to be extremely close to the UAV in order to enter this blind spot, which is not realistic. Therefore, this blind spot does not need to be considered. However, the deployment of two receiver stations involves additional weight and energy consumption, and it requires additional legs on the UAV, as depicted in Fig. 5.20c, to increase the safety of the sensor in case of UAV crashes.

For the system hardware design, the beacons and receivers had to be developed. Each IR beacon contain four high-power IR LEDs inside a plastic diffuser, connected in series and displayed in a tetrahedron configuration to homogenize the emission intensity in each direction, as illustrated in Fig. 5.19b. The chosen LEDs were the VSMY98545 High Power IR Emitting Diode from Vishay Semiconductors[2]. The plastic diffuser consists of a 4 *cm* diameter sphere. Each beacon emits with a different intensity since the electrical current passing through its LEDs is different. This fact is compensated using the previous parameter $C_{b_{jm}}$ for marker beacon $b_{jm}$ when calibrating sensor $i$. Additionally, in spite of using a plastic diffuser, the light intensity of the beacons still present an anisotropic behavior. Experiments showed that the light rays coming from the beacon are stronger in the direction that aligns the UAV body with the beacon. This direction is named as beacon heading, and it is illustrated in Fig. 5.18a. The experiments revealed a maximum signal intensity decay of 20% for an angle of $120^o$ with respect to the beacon heading (see Fig. 7.8). Angles larger than this value will likely result in beacon occlusion by the UAV body. Therefore, each beacon can be considered to have a quasi-homogeneous signal intensity on all the directions where the beacon can be properly detected.

The IR receiver is implemented individually using a modular design, as illustrated in Fig. 5.19a.

[2] www.vishay.com

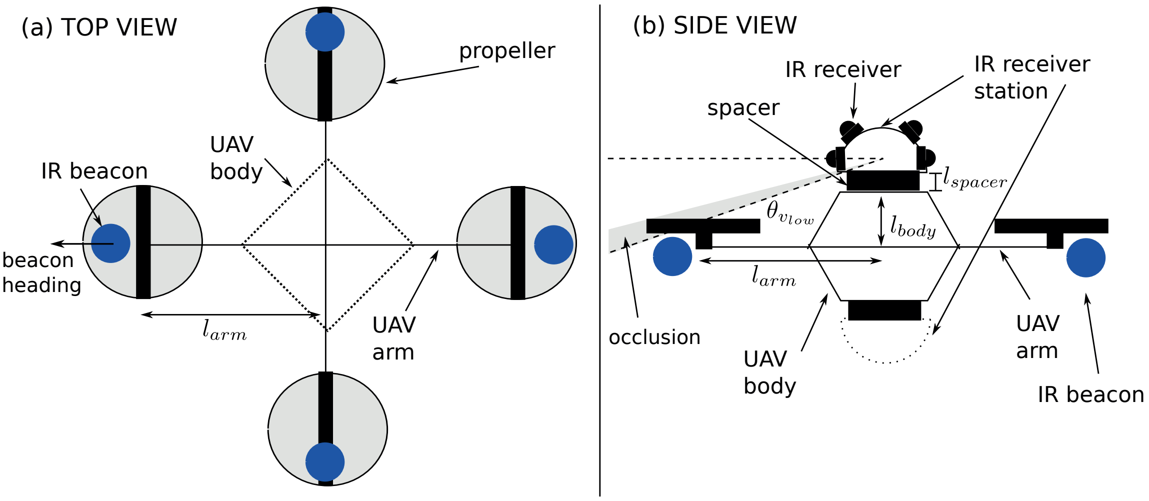


Figure 5.18: Geometric considerations for the hardware design of the IR-based relative positioning system. The beacon and receiver poses in the UAV body frame are considered. The receivers are placed into receiver stations, to be inserted on top and bottom of the UAV body frame. The top (a) and side (b) view of the system are illustrated. Note the definition of the beacon heading.

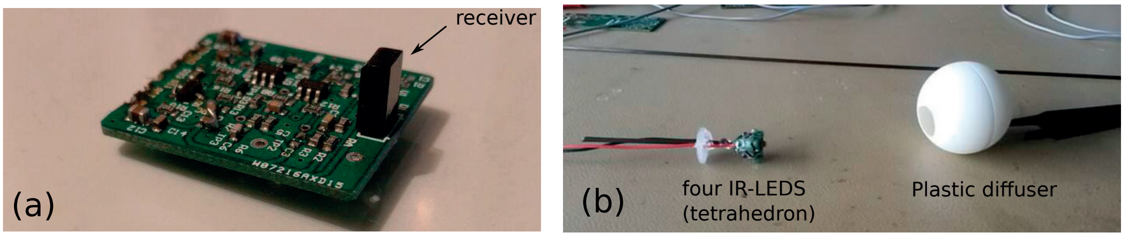


Figure 5.19: IR beacon and IR receiver hardware specifications for the IR-based relative positioning system. (a) Single receiver. (b) Single beacon.

This allows the easy deployment of any number of receivers on any desired pose in the UAV, making this system adaptable to different hardware geometries. The receiver is composed by the actual receiving device (SFH 225 FA Silicon PIN Photodiode with Daylight Blocking Filter from OSRAM Opto Semiconductors Inc.[3]), and the amplification stage implementing the cascaded approach described earlier. As previously discussed, the beacons pulses are modulated by squares signals of a predefined frequency. In this work the predefined frequency is 1MHz. This allows the filtering stages of the cascade amplification to filter out environment and interfering device noise. This enables an increase of the receiver sensitivity, increasing the sensor dynamic range. In this work, the amplification stages are directly connected, generating a single output signal for the entire amplification series, in contrast with the work in [91,92]. This provides a simpler and lighter design as less output signals need to be analyzed. This work uses fewer amplification stages (only two) for simplicity, allowing detection ranges smaller than the ones in observed in [91,92]. However, additional stages can be added to improve the detection range. It is worth to noticing that the IR wavelength chosen for the receiver detection is 950 *nm*.

[3] www.osram.com

Both the beacons and receivers are managed by a control board placed as a base for the receiver station, as depicted in 5.20a. This approach again minimizes the wiring used for the sensor. The control is performed with three main microcontrollers. Two microcontrollers are responsible for sampling the receivers for each different emission slot, described in Section 5.1.2.1. Each sampling microcontroller is capable of supporting up to eight receivers. Therefore, two dedicated microcontrollers are required to sample the sixteen receivers that are included in each receiver station. The third microcontroller is the main processing unit on the board, responsible to perform the tasks depicted in Fig. 5.11. The beacon ID is detected from the receiver measurements and their respective time slots acquired by the sampling microcontrollers. This information is then relayed to the higher layers so that the relative pose estimation algorithm can be applied. Ingoing and outgoing communication is also processed, as described in Section 5.1.2.1. Finally, the emission slot period and the beacon pulse sequence of each beacon is controlled according to the currently available time slots in the IR channel and the outgoing bitstreams computed from the communication algorithm.

This controller board defines the sensor measuring frequency and the communication speed of the described IR-based relative positioning system. The measuring frequency dictates how long the main processing unit on the board has to wait before relaying new sensor information to the higher levels, necessary for the relative pose estimation algorithm. This period is defined by a certain number of TDMA periods (this period is defined in Section 5.1.2.1). If the information relay period is defined by $N_c$ TDMA periods, each TDMA period has $N_s$ emission slots, and each slot has a period of $T$ seconds, the measuring frequency of the system can be computed as:

$$f_{sensor} = \frac{1}{N_s N_c T}. \tag{5.19}$$

In this work, $N_c = 20$, $N_s = 60$, and $T = 10\ us$, which gives a theoretical measuring frequency of 83 $Hz$. $N_c$ can be tuned in order to give more processing time to the microcontrollers between information relays (for example, if some digital filtering needs to be done to the RSS measurements before it is given to the higher layers). $N_s$ can be tuned in order to add or remove possible beacon positions in the IR channel (in theory with $N_s = 60$ slots, thirty beacons can be placed in the IR channel). Finally, $T$ can be tuned if the amplification stages are implemented with stronger filters that require more time to stabilize (in this work, we used first order filters).

Note that when two independent controller boards are placed on the same UAV (in case of both top and bottom stations are used), the higher layers have to be able to fuse the information flow from both controllers. In this work, an higher level sensor driver is developed in order to allow the connection of any number of sensor controller boards. In this driver, one of the boards is defined as a master, which will be in charge of the beacon emission and slot synchronization. The other boards (the slaves) synchronize their times with this master. The sensor driver collects the RSS measurements taken by all receivers at all receiver stations, and associates each measurement to a receiver pose in the marker frame. This information can then be used in the relative pose estimation algorithm defined in the previous sections.

Finally it is worth noticing that the IR-based positioning system deployed as in Fig. 5.20 (four bea-

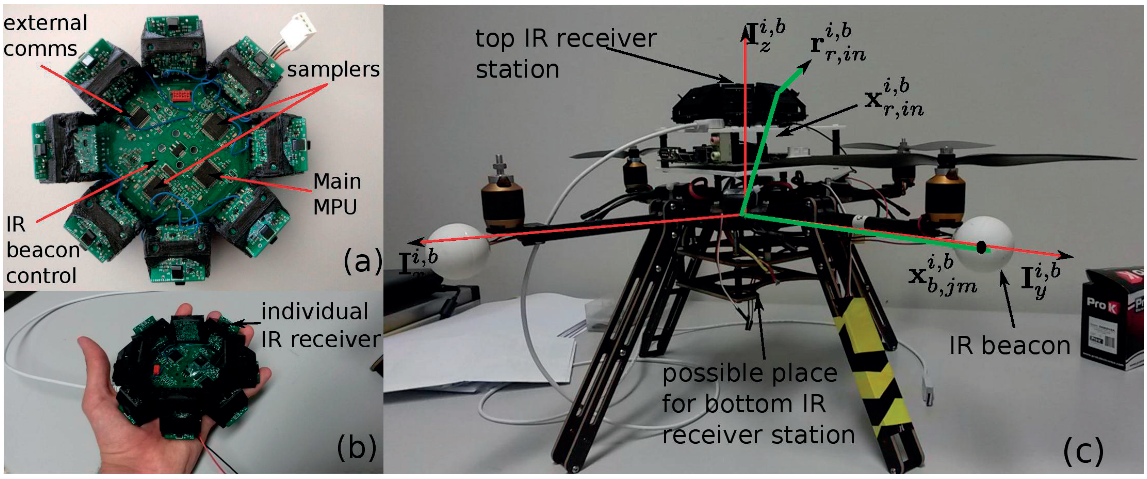


Figure 5.20: Illustration of the IR-based relative positioning system hardware. (a) IR receiver station controller board. (b) Receiver station with sixteen receivers, wight pointed horizontally and eight pointed 45° up, all with 45° spacing. (c) Possible configuration of the system on a quadrotor UAV. Note the UAV body or marker frame definition.

cons plus one receiver station pointing upwards) has a weight of 110 $g$. An additional 90 $g$ is required to add a bottom receiver station, making a maximum system weight of 200 $g$. The system composed of the four beacons and the two receiver stations require 7 $W$ of power during the system operation. The entire system is at least two times lighter than the ones reported in the literature [92].

## 5.2 Onboard formation controller

The main task addressed in this work is the formation control of UAVs using exclusively onboard sensors and control algorithms. The formation control problem is addressed by making use of a graph-based formation control algorithm, relying on relative inter-vehicle localization measurements from sensors on board each UAV, as described in Section 4.2. The relative inter-vehicle localization measurements are acquired in this work using the relative positioning systems developed in Section 5.1. These systems are either camera-based or IR-based. Additionally, the formation control algorithm requires onboard OF and height sensors on each UAV in order to be able to move the formation in the environment.

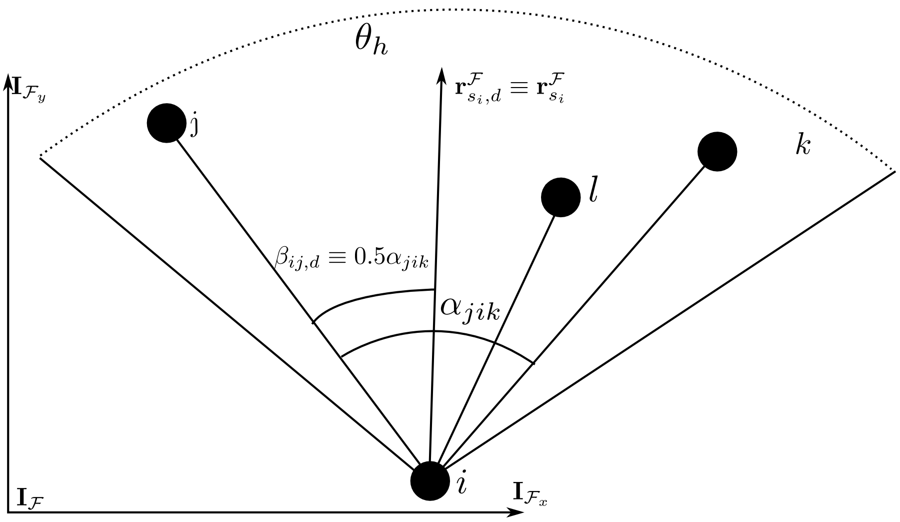
Although the proposed relative positioning systems already acquire the necessary measurements required to control the desired inter-vehicle geometric constraints, the baseline formation control presented in Section 4.2.2 is affected by some problems. The first problem is related to the sensing constraints of the relative positioning systems. An example of such constraints are the FOV limitations of the camera-based system developed Section 5.1.1. These constraints can severely limit the number of neighbors that can be observed by the UAV, and therefore, the number of possible geometric configurations achievable by the formation. Additionally, since the observed neighbor can fall out from the FOV of the sensor, these constraints also severely limit the reactiveness of each UAV, and therefore, of the entire formation. This work tackles this problem by designing the formation geometric configurations taking these sensing constraints into account. These sensing constraints include those that involve several neighbors inside the sensor FOV, in contrast with the previous literature presented in Section 2.2.3. In order to accomplish this, the baseline formation control algorithm presented in Section 4.2.2 is modified in order to directly consider the FOV constraints in the formation control algorithm, so that they will be respected during formation operation. These modifications are detailed in Section 5.2.1.

The second problem is related to the steering of the formation to the desired goal in the environment. Two approaches are considered, the first one being a leader-follower approach and the second a virtual structure approach (see Section 4.2 for details). The leader-follower approach, already implemented in the control algorithm presented in Section 4.2.2, simplifies the task of directing the formation to the desired direction using the leader. However, as explained in Section 2.2.2, motion lags between the leader and the followers caused by delays present on the UAVs' perception-to-action-loops can severely compromise the formation reactiveness. The virtual structure approach reduces these delays by giving the desired motion of the virtual structure to all UAVs simultaneously. The UAVs will then simultaneously translate the desired motion of the virtual structure to their individual desired motions. These individual motions will be consistent across all the UAVs if there is a consistency of the virtual structure within the multi-UAV system. When only relative inter-vehicle localization measurements are present, this consistency is achieved in the literature by means of additional consensus algorithms using additional inter-vehicle communications. When available, features externally to the formation can also be used. However, inter-vehicle communication can become unfeasible for highly dynamical systems such as the UAVs, and features externally to the formation might not be available. This work proposes a novel formation steering algorithm using a virtual structure approach solely using relative inter-vehicle localization measurements. The consistency of the virtual structure is achieved without resorting to additional inter-vehicle communication or any features externally to the formation. Instead, the information about the virtual structure is directly encoded into the inter-vehicle geometric constraints of the neighborhood of each UAV. The details of the steering algorithm are described in Section 5.2.2.

As a side effect of developing the novel formation steering algorithm, this work also shows that the formation can be steered as a whole using a teleoperator measuring the center of the formation without requiring individual UAV pose measurements. This makes this approach scalable with the number of UAVs.

### 5.2.1 Field of view constraints

In order to account for the sensor FOV constraints of the relative positioning systems, such as the camera-based system described in Section 5.1.1, this work adapts the formation control algorithm presented in Section 4.2.2 in order to account for these constraints. The sensor FOV constraints can be mathematically described through the concept of inter-edge aperture. Given two nodes $j$ and $k$ connected to node $i$, the inter-edge aperture of the respective connection edges, $\alpha_{jik}$, is defined by the angle between the relative position vectors represented by those edges, as shown in Fig. 5.21. This concept is divided into horizontal and vertical components, by projecting the edges in the respective planes defined in Fig. 4.1. Therefore,



$\theta_h$
$\mathbf{I}_{\mathcal{F}_y}$
$\mathbf{r}^{\mathcal{F}}_{s_i,d} \equiv \mathbf{r}^{\mathcal{F}}_{s_i}$
j
k
$l$
$\beta_{ij,d} \equiv 0.5\alpha_{jik}$
$\alpha_{jik}$
$i$
$\mathbf{I}_{\mathcal{F}}$
$\mathbf{I}_{\mathcal{F}_x}$

Figure 5.21: Inter-edge aperture between the edges connecting nodes $j$ and $k$, to node $i$, both within the FOV of node $i$. Inter-edge apertures formed by different node pairs, for example using node $l$, are contained inside it. Note the desired sensor direction, $\mathbf{r}^{\mathscr{F}}_{s_i,d}$, equally dividing the occupied area on both sides of the FOV. All quantities are expressed in the formation frame. For simplification, the third dimension is omitted.

the inter-edge aperture of every pair of edges belonging to $\mathscr{G}_S$ for a given UAV $i$ must be smaller than $\theta_h$ in the horizontal case, or $\theta_v$ in the vertical case. Only the biggest aperture for each UAV needs to be considered, because if this value is smaller than the FOV limit, all the other apertures will be also smaller, as illustrated in Fig. 5.21.

As previously described in Section 4.2.2, the desired formation geometric configuration is expressed in the formation frame. Note that this configuration has to verify the previously defined FOV constraints. This means that all horizontal or vertical inter-edge apertures formed by all pairs of UAVs $j$ and $k$ with sensing edges to UAV $i$ (i.e. $\mathscr{E}_{S_{ij}}$ exists) have to be smalller than $\theta_h$ or $\theta_v$ respectively. If the FOV constraints are verified, there is always a $\psi_{fi}$, describing UAV $i$'s front (and therefore identifying the direction of the respective onboard sensor $\mathbf{r}^{\mathscr{F}}_{s_i}$), that allows all neighbors to be observed. In fact, it is possible to define a desired sensor direction $\mathbf{r}^{\mathscr{F}}_{s_i,d}$, illustrated in Fig. 5.21, defined so that it equally distributes the sensing area around the center of the largest inter-edge aperture, optimizing the measurement safety margins. This desired direction is used to define the UAV $i$'s attitude expressed in the formation frame $\mathbf{R}_{\psi_{fi,d}}$.

In this work, we consider primarily the horizontal FOV constraints, since quadrotors in close range formation typically have similar height in order to avoid interference from propeller airflow. Therefore, this work assumes geometric configurations where all UAVs have a height that lies in the vertical FOV of the neighboring vehicles. Note that the sensor direction $\mathbf{r}^{\mathscr{F}}_{s_i}$ is always directed and controlled horizontally. Therefore, the vertical FOV direction can be assumed to be centered with the horizontal plane, as defined in Section 4.2.2. The baseline formation control algorithm can now be modified to guarantee that the defined FOV constraints are respected during formation operation. This is done by directly controlling the inter-edge apertures between the UAVs, as described in the following section.

#### 5.2.1.1 Control algorithm

In order to guarantee that the previously described FOV constraints are respected during formation operation, the controller defined in Section 4.2.2 has to be modified. Firstly, the attitude controller, defined in Eq. (4.11), is changed in order to direct the relative positioning sensor on board UAV $i$ towards the previously described desired direction $\mathbf{r}_{s_i,d}^{\mathscr{F}}$. UAV $i$ first selects the two neighbors $j$ and $k$ that form the largest measured inter-edge aperture. Only these two neighbors are used in the controller defined in Eq. (4.11), by setting $c_{\psi_{ij}}$ and $c_{\psi_{ik}}$ to 1, and the remaining constants to 0. In this way, the control law becomes:

$$\omega_{i_z,d} = k_\psi \mathbf{L}_{ij} \left(\beta_{ij,d} - \beta_{ij}\right) + k_\psi \mathbf{L}_{ik} \left(\beta_{ik,d} - \beta_{ik}\right).$$

Note that the direction of the UAV $i$'s sensor $\mathbf{r}_{s_i}^{\mathscr{F}}$ corresponds to a constant bearing in that UAV's frame, $\beta_{s_i}$. With the help of Fig. 5.21, it is possible to observe that $\beta_{s_i} = 0.5(\beta_{ij,d} + \beta_{ik,d})$. Therefore, by selecting $k_\psi$ in both $j$ and $k$ terms in order to equalize the $\mathbf{L}_{ij}$ and $\mathbf{L}_{ik}$ gains, the control law can be rewritten as:

$$\omega_{i_z,d} = -k_\psi \left(0.5(\beta_{ij} + \beta_{ik}) - \beta_{s_i}\right), \tag{5.20}$$

which will control the attitude of UAV $i$ such that the sensor direction $\mathbf{r}_{s_i}^{\mathscr{F}}$ remains centered with the largest measured inter-edge aperture.

Although the previous attitude controller is able to optimize the measurement safety margins, it does not guarantee that the desired inter-edge aperture is kept during formation operation. In order to achieve this last requirement, a new term is added to the controller defined in Eq. (4.13) in order to control the largest inter-edge aperture measured by UAV $i$, $\alpha_{kij}$, as follows:

$$\acute{\acute{\mathbf{a}}}_{i_h,d}^{\mathscr{L}_i} = \acute{\mathbf{a}}_{i_h,d}^{\mathscr{L}_i} + k_\alpha(\alpha_{kij,d} - \alpha_{kij})(K\mathbf{r}_{c_i}^{\mathscr{L}_i} + K_\perp \mathbf{r}_{c\perp_i}^{\mathscr{L}_i}), \tag{5.21}$$

where $\alpha_{kij,d}$ is the desired inter-edge aperture between UAV $i$ and neighbors $j$ and $k$ currently defining the largest inter-edge aperture, $k_\alpha$ is a control gain, and $K$ and $K_\perp$ will be chosen according to the stability analysis presented in Proposition 1. Vector $\mathbf{r}_{c_i}^{\mathscr{L}_i}$, illustrated in Figs. 5.22 and 5.23, is a unitary vector defining the direction between UAV $i$ and the averaged formation center, $\mathbf{C}_i^{\mathscr{L}_i}$. This center is defined for each UAV, and its displacement from the robot is computed using the first term of Eq. 4.10:

$$\mathbf{C}_i^{\mathscr{L}_i} = \mathbf{r}_{c_i}^{\mathscr{L}_i} e_{c_i} = \sum_{j=1}^{N} \mathbf{r}_{ij_h}^{\mathscr{L}_i} \mathbf{L}_{ij} e_{ij_h}, \tag{5.22}$$

where $e_{c_i}$ is the range between UAV $i$ and $\mathbf{C}_i^{\mathscr{L}_i}$. The range between a neighbor $j$ and $\mathbf{C}_i^{\mathscr{L}_i}$ is defined as $e_{c_{ij}}$. Note that $\mathbf{r}_{c_i}^{\mathscr{L}_i}$ is always in between the neighbors closer to the FOV edges. Therefore, if $\alpha_{kij}$ is too large, robot $i$ generates a force pointing backwards, in a direction that will always decrease $\alpha_{kij}$. The contrary occurs when $\alpha_{kij}$ is too small. This corresponds to a direct control of the sensor FOV constraint

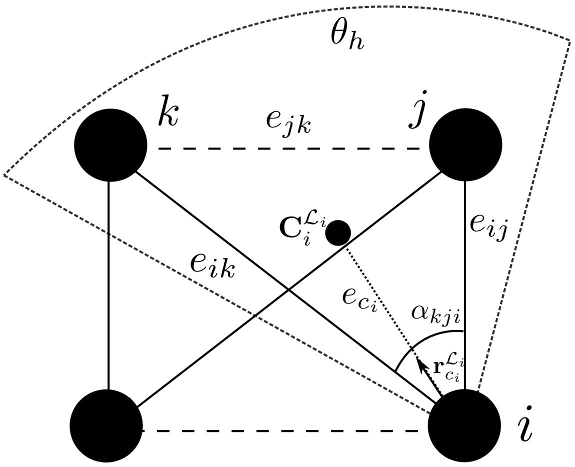


Figure 5.22: Possible $\mathscr{G}_F$ for a square formation and an horizontal FOV of less than 90°. Filled edges correspond to direct range control. Dashed edges are included if inter-edge aperture control is activated. Note UAV $i$'s weighted formation center.

during formation operation.

With the additional control terms, each UAV also indirectly controls the ranges between the neighbors themselves, without them knowing the respective relative position information. Consider the example in Fig. 5.22, when the sensor FOV is less than 90° (experiments using this configuration are examined in detail in Section 7.2). The most complete $\mathscr{G}_S$ possible with these constraints is the one shown in the figure (excluding the dashed lines). So, with just direct range control, no rigid graph can be defined. However, if UAV $i$ additionally controls $\alpha_{jik}$, one can see that $e_{jk}$ is fully expressed in terms of the direct controlled quantities, $e_{ij}$, $e_{ik}$, and $\alpha_{jik}$. So, the edge between neighbors $j$ and $k$ is automatically included in $\mathscr{E}_F$, which does not need to belong to $\mathscr{G}_S$. Each UAV can control an additional edge in this way, allowing the establishment of rigid, or even fully connected, formation graphs that could not possibly be formed before.

The next proposition shows that the proposed formation control algorithm is stable. The algorithm can be easily extended to control more than one inter-edge aperture for each UAV, but the presented stability properties are related to the largest. Additionally, it is not clear that these properties hold if the neighbors' positions forming the aperture change in time. Since this latter situation is rare, in this thesis we focus on characterizing the algorithm behavior when the inter-edge aperture remains the same during operation. Finally, the stability properties are presented for the control algorithm using all the components except those related to the control of the UAV horizontal velocity and height in the environment, presented in Eqs. (4.13) and (4.12). These components are treated as disturbances applied to the algorithm. The impact of these disturbances is analyzed later, either in simulation or reality.

**Proposition 1.** *As long the necessary FOV constraints are not violated, the multi-UAV system, with each UAV i described by the dynamics in Eq.* (3.17) *and applying the controller presented in Eqs.* (4.9), (4.10), (5.20)*, and* (5.21)*, is stable, for any chosen set of weights described in* $\mathbf{L}$*, and any* $k_p$*,* $k_v$*,* $K$ *and* $K_\perp$ *greater than zero.*

*Proof.* Note that [87] already proved convergence properties for the vertical controller components in Eq. (4.9). Also, the attitude controller defined in Eq. (5.20) is independent from the linear formation control. Therefore, if the formation converges, the desired sensor direction $\mathbf{r}_{s_i,d}^{\mathscr{F}}$ converges as well. In this way, the attitude controller is able to stabilize $\mathbf{r}_{s_i}^{\mathscr{F}}$ to this desired direction since it is a proportional control applied to a single integrator system.

The proof for the horizontal controller components of $\ddot{\mathbf{a}}_{i_h,d}^{\mathscr{L}_i}$ follows the reasoning of [31], which performs the analysis separately for each UAV while its neighbors are assumed to have fixed positions, and then combines the results at the end. To simplify the proof, the bias terms ($\alpha_{kij,d}$ and $e_{ij,d}$) are set to zero, but the stability still holds for non-zero terms (such terms would only change the equilibrium point). For this proof, all the quantities are expressed in the absolute frame $\mathbf{I}_{\mathscr{W}}$. Therefore, $\ddot{\mathbf{a}}_{i_h,d}^{\mathscr{L}_i}$ is converted to the absolute frame $\ddot{\mathbf{a}}_{i_h,d}^{\mathscr{W}} = \mathbf{R}_{\psi_i}\ddot{\mathbf{a}}_{i_h,d}^{\mathscr{L}_i}$.

Let us assume the case for UAV $i$, described in Fig. 5.23, representative of the horizontal control components. UAVs $j$ and $k$ correspond to UAV $i$ neighbors forming the largest inter-edge aperture in its FOV. The position of UAV $i$'s formation center in the absolute frame, $\mathbf{C}_i^{\mathscr{W}}$, can be expressed in the absolute frame as $\mathbf{x}_i^{\mathscr{W}} + \mathbf{r}_{c_i}^{\mathscr{W}} e_{c_i}$, where the second term comes from Eq. 5.22. This expression does not depend on $\mathbf{x}_i^{\mathscr{W}}$ if $\sum_{j\neq i}\mathbf{L}_{ij} = 1$, which means that $\mathbf{C}_i^{\mathscr{W}}$ doesn't change with UAV $i$'s movements. If the neighbors are assumed to have fixed positions, their distances to $\mathbf{C}_i^{\mathscr{W}}$, $e_{c_{ij}}$ and $e_{c_{ik}}$, are constant. The axes of UAV $i$'s local frame are changed to $\mathbf{r}_{c_i}^{\mathscr{W}}$ and $\mathbf{r}_{c\perp_i}^{\mathscr{W}}$, representing respectively the radial and orthogonal axis with respect to $\mathbf{C}_i^{\mathscr{W}}$. The velocity of $\mathbf{C}_i^{\mathscr{W}}$ in this new frame is decomposed on the radial, $\dot{e}_{c_i}$ and orthogonal, $\dot{e}_{c\perp_i}$, axes. Note that this velocity corresponds to the second term of Eq. 4.10, and that $\dot{e}_{c_i}$ is the velocity of $e_{c_i}$, representing the range between UAV $i$ and $\mathbf{C}_i^{\mathscr{W}}$. From the previous definitions, a simple Lyapunov function is chosen to analyze the stability of the system for UAV $i$:

$$V_i(\alpha_{kij}, e_{c_i}, \dot{e}_{c_i}, \dot{e}_{c\perp_i}) = \frac{1}{2}(k_\alpha \alpha_{kij}^2 + k_p e_{c_i}^2 + (\dot{e}_{c_i})^2 + (\dot{e}_{c\perp_i})^2),$$

which is greater than zero except in $V_i(0,0,0,0)$. The four components were considered because they represent the states that are being controlled (inter-edge aperture, range, and radial and orthogonal velocity). The derivative of $V_i$ with respect to time can be expressed as:

$$\dot{V}_i = k_\alpha \alpha_{kij}\dot{\alpha}_{kij} + k_p e_{c_i}\dot{e}_{c_i} + \dot{e}_{c_i}\ddot{e}_{c_i} + \dot{e}_{c\perp_i}\ddot{e}_{c\perp_i},$$

where $\ddot{e}_{c_i}$ and $\ddot{e}_{c\perp_i}$ can be expressed from the $\ddot{\mathbf{a}}_{h_i,d}^{\mathscr{L}_i}$ terms in Eqs. 4.10 and 5.21, projected into the radial and orthogonal components respectively:

$$\ddot{e}_{c_i} = K k_\alpha \alpha_{kij} - k_p e_{c_i} - k_v \dot{e}_{c_i},$$
$$\ddot{e}_{c\perp_i} = K_\perp k_\alpha \alpha_{kij} - k_v \dot{e}_{c\perp_i}.$$

Therefore, $\dot{V}_i$ can be simplified, by removing the equal terms in its expression, to:

$$\dot{V}_i = k_\alpha \alpha_{kij}\dot{\alpha}_{kij} + k_\alpha \alpha_{kij}(\dot{e}_{c_i}K + \dot{e}_{c\perp_i}K_\perp) - k_v(\dot{e}_{c_i})^2 - k_v(\dot{e}_{c\perp_i})^2.$$

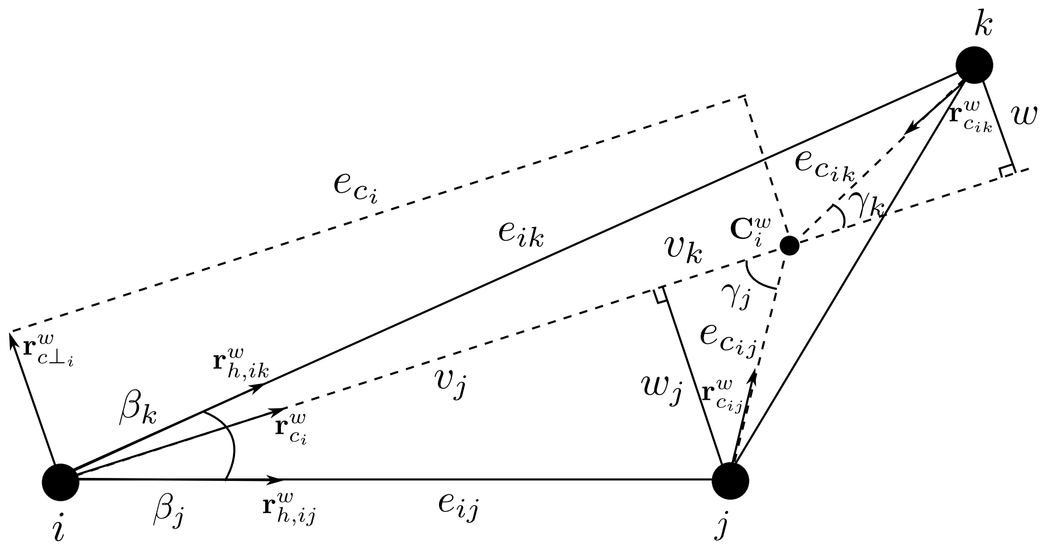


Figure 5.23: Definition of important quantities necessary for the formation control algorithm stability analysis. Only the horizontal component of the control algorithm is considered, but these quantities are defined for both horizontal and vertical components.

The last two terms are always negative, which leaves the study of the first component. From Fig. 5.23, the aperture $\alpha_{kij}$ can be divided into $\beta_j + \beta_k$, where $\beta_{(*)} = \arctan(w_{(*)}/v_{(*)})$, and $(*)$ is either $j$ or $k$. By differentiating $\beta_{(*)}$, $\dot{\alpha}_{kij}$ can be defined as:

$$\dot{\alpha}_{kij} = \frac{v_j\dot{w}_j - \dot{v}_j w_j}{e_{ij}^2} + \frac{v_k\dot{w}_k - \dot{v}_k w_k}{e_{ik}^2}.$$

From the figure, one can define $v_{(*)} = e_{c_i} - e_{c_{i(*)}}\cos(\gamma_{(*)})$ and $w_{(*)} = e_{c_{i(*)}}\sin(\gamma_{(*)})$, where $\gamma_{(*)}$ is the angle going from $\mathbf{r}^{\mathscr{W}}_{c_{i(*)}}$ to $\mathbf{r}^{\mathscr{W}}_{c_i}$. Recalling that neighbors have fixed positions with respect to $\mathbf{C}^{\mathscr{W}}_i$, $\gamma_{(*)}$ only depends on $\dot{e}_{c\perp_i}$, according to the linear-to-angular velocity equation, $\dot{e}_{c\perp_i} = \dot{\gamma}_{(*)}e_{c_i}$. Using the previous result for $\dot{\gamma}_{(*)}$, and recalling that $e_{c_{i(*)}}$ is constant, the derivatives of the previous expressions for $v_{(*)}$ and $w_{(*)}$ are as follows:

$$\dot{v}_{(*)} = \dot{e}_i + \dot{e}_{c\perp_i}\sin(\gamma_{(*)})\frac{e_{c_{i(*)}}}{e_{c_i}}, \quad \dot{w}_{(*)} = \dot{e}_{c\perp_i}cos(\gamma_{(*)})\frac{e_{c_{i(*)}}}{e_{c_i}}.$$

From the previous result, and noting that the alternative definitions $w_{(*)} = e_{i(*)}\sin(\beta_{(*)})$ and $v_{(*)} = e_{c_{(*)}}\cos(\beta_{(*)})$, the previous expression for $\dot{\alpha}_{kij}$ can be re-arranged, to isolate the terms in $\dot{e}_{c\perp_i}$ and $\dot{e}_{c_i}$, as follows:

$$\begin{aligned}\dot{\alpha}_{kij} = \dot{e}_{c\perp_i}&\left(\frac{e_{c_{ij}}}{e_{c_i}}\frac{\cos(\gamma_j+\beta_j)}{e_{ij}} + \frac{e_{c_{ik}}}{e_{c_i}}\frac{\cos(\gamma_k+\beta_k)}{e_{ik}}\right)\\ -\dot{e}_{c_i}&\left(\frac{\sin(\beta_j)}{e_{ij}} + \frac{\sin(\beta_k)}{e_{ik}}\right).\end{aligned}$$

One can now choose $K$ and $K_\perp$ of the aperture controller to eliminate the previous components, ending with the following result:

$$\begin{aligned}
K &= \left(\frac{\sin(\beta_j)}{e_{ij}} + \frac{\sin(\beta_k)}{e_{ik}}\right) \\
K_\perp &= -\left(\frac{e_{c_{ij}}}{e_{c_i}}\frac{\cos(\gamma_j+\beta_j)}{e_{ij}} + \frac{e_{c_{ik}}}{e_{c_i}}\frac{\cos(\gamma_k+\beta_k)}{e_{ik}}\right) \\
\dot{V}_i &= -k_v(\dot{e}_{c_i})^2 - k_v(\dot{e}_{c\perp_i})^2.
\end{aligned}$$

Note that $\dot{V}_i \leq 0$, and therefore using the Lyapunov theorem, the system with the proposed controller is stable and converges to a subset of the state-space defined by $\dot{V}_i = 0$. This can be extended to all UAVs, by setting $V = V_1 + ... + V_N$, where $V_i$ is the previous Lyapunov function but for each UAV. $V > 0$ except in $V(0) = 0$, and $\dot{V} <= 0$, and therefore, the system as a whole is also stable. Finally, note that $k_p \sum_{j\neq i} \mathbf{L_{ij}}$ can always be transformed into $k_p^* \sum_{j\neq i} \mathbf{L_{ij}^*}$ for each UAV, where $\sum_{j\neq i} \mathbf{L_{ij}^*} = 1$, necessary to guarantee that $\mathbf{C}_i^{\mathscr{W}}$ does not depend on robot $i$'s movement. This allows the use of any Laplacian matrix in this system, regardless of $k_p$. □

The system is stable, but it converges to the set described as $\dot{V} \leq 0$, which, from the previous proof, only guarantees that the UAV velocities are zero. Deadlocks can occur, especially if the configuration is ill defined, i.e. the set of desired ranges and apertures correspond to an impossible geometric configuration. In this case, the system will converge to a situation where the aperture controller will counter-act the range controller, creating the deadlock. Investigating how such deadlocks can be avoided is considered to be future work. However, note that those already existed in [31], referenced as local minima. As previously discussed, the horizontal velocity and height control terms in Eqs. (4.12) and (4.12) are considered as disturbances applied to the formation control algorithm. In Section. 7.2.2, an experiment shows the nature of these disturbances and how the formation control algorithm is able to cope with them, while the formation moves in the environment using a leader-follower approach.

Finally, the gains $K$ and $K_\perp$ found for the inter-edge aperture controller are analyzed. The value of $K$ is related to the controller radial component. Its value is intuitive, saying that it is always bigger than zero, as $\beta_{(i)} \leq \pi$, and it is bigger as the angle increases to $\pi/2$, corresponding to the point of maximum influence of the controller in the angle. Also, as $e_{ij}$ decreases, the gain increases since the influence on the aperture also increases. The value of $K_\perp$ is related to the controller orthogonal component, and it is less intuitive. However, note that it uses cosine instead of sine functions, indicating that it is controlling an axis orthogonal to the one K controls. For example, if all UAVs are found in a line, $K = 0$, since moving on the line does not control the aperture, but $K_\perp \neq 0$, since moving orthogonally to the line increases the aperture. For simplicity, and due to time constraints, this work considers $K_\perp = 0$, but the experiments show that the system can converge without this component. Future work will include experiments done with a non-zero value of $K_\perp$ in the controller.

### 5.2.2 Formation steering using virtual structure

To steer the formation through the environment, this work considers two approaches. The first approach consists in a leader-follower approach, implemented by the baseline controller presented in Section 4.2.2. The second approach consists in a virtual structure approach, which is more robust to motion lags between UAVs allowing for more reactive formations (see introduction of Section 5.2 for details). Both approaches operate by defining at each instant the desired horizontal velocity ($\mathbf{v}_{i_h,d}^{\mathscr{L}_i}$) and desired height ($z_{i,d}^{\mathscr{L}_i}$) of each UAV. These quantities are controlled on each UAV through Eqs. (4.12), (4.13). The leader-follower approach simply defines the value of the previous defined quantities for a set of leader UAVs. The virtual structure approach steers the entire formation as a virtual structure, similarly to [37,98]. The virtual structure is defined by the position and the attitude of the formation frame (see Section 4.2.2 for details about the definition of this frame). In this work, the position of the formation frame is also named as formation center. The desired motion of the virtual structure is defined in terms of the translation of its 3D position (i.e. the formation center) in the environment, or its angular velocity (i.e. rotation around the formation center).

These motion directives are implemented by means of formation motion commands issued by a system external to the formation. The commands consist of a desired formation horizontal velocity $\mathbf{u}_{vf_h} = (u_{vf_x}, u_{vf_y})$, a desired formation height $u_{zf}$, and a desired formation angular velocity $\mathbf{w}_{wf} = (u_{wf_x}, u_{wf_y}, u_{wf_z})$, all expressed in the formation frame as depicted in Fig. 5.24b. The angular velocity commands used in this work are set to zero around the $x$ and $y$ axes ($u_{wf_x}$ and $u_{wf_y}$) in order to avoid unwanted horizontal movement. The same motion command is sent to all UAVs of the formation at the same time via a communication channel. Once these motion commands are received by the UAVs, they are transformed into the desired motion commands of each UAV. This is done as follows: when UAV $i$ receives the $\mathbf{u}_{vf_h}$ and $u_{zf}$ commands, it has to adjust the desired velocity $\mathbf{v}_{i_h,d}^{\mathscr{W}}$ and height $z_{i,d}^{\mathscr{W}}$ in the environment as follows:

$$\begin{cases} \mathbf{v}_{i_h,d}^{\mathscr{W}} = \mathbf{I}_{\mathscr{F}_x}^{\mathscr{W}} u_{vf_x} + \mathbf{I}_{\mathscr{F}_y}^{\mathscr{W}} u_{vf_y} \\ z_{i,d}^{\mathscr{W}} = u_{zf} - (1/(N-1)) \sum_{j=1}^{N} z_{ij,d}^{\mathscr{W}} \end{cases}, \tag{5.23}$$

where ($\mathbf{I}_{\mathscr{F}_x}^{\mathscr{W}}, \mathbf{I}_{\mathscr{F}_y}^{\mathscr{W}}$) correspond, respectively, to the $x$ and $y$ axes of $\mathbf{I}_{\mathscr{F}}$ expressed in the absolute frame. When UAV $i$ receives the $\mathbf{u}_{wf_z}$ commands, it has to further adjust $\mathbf{v}_{i_h,d}^{\mathscr{W}}$ to allow a rotation of the UAV with respect to the formation center, as follows:

$$\mathbf{v}_{i_h,d}^{\mathscr{W}} = e_{c_i} \mathbf{r}_{c\perp_i}^{\mathscr{W}} u_{wf_z}, \tag{5.24}$$

where ($e_{c_i} \mathbf{r}_{c_i}^{\mathscr{W}}$) corresponds to the position of the formation center relative to UAV $i$ ($-\mathbf{x}_{fi}^{\mathscr{W}}$) expressed in polar coordinates, and $\mathbf{r}_{c\perp_i}^{\mathscr{W}}$ is the vector orthogonal to $\mathbf{r}_{c_i}^{\mathscr{W}}$, as depicted in Fig. 5.24a. Note that $\mathbf{r}_{c\perp_i}^{\mathscr{W}} = \mathbf{I}_{\mathscr{W}_z} \times \mathbf{r}_{c_i}^{\mathscr{W}}$. The motion command transformation described in Eqs. (5.23) and (5.24) can be transformed into UAV $i$'s flying frame by multiplying each side of the equations by the inverse of the current UAV attitude in the absolute frame $\mathbf{R}_{\psi_i}$, similarly to what was done in Section 3.3. The law for transforming the formation motion commands into individual UAV motion commands becomes:

$$\begin{cases} \mathbf{v}_{i_h,d}^{\mathscr{L}_i} = \mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i} u_{vf_x} + \mathbf{I}_{\mathscr{F}_y}^{\mathscr{L}_i} u_{vf_y} + e_{c_i} \mathbf{r}_{c\perp_i}^{\mathscr{L}_i} u_{wf_z} \\ z_{i,d}^{\mathscr{L}_i} = u_{zf} - (1/(N-1)) \sum_{j=1}^{N} z_{ij,d}^{\mathscr{L}_i} \end{cases}, \tag{5.25}$$

where $z_{i,d}^{\mathscr{L}_i}$ and $\mathbf{v}_{i_h,d}^{\mathscr{L}_i}$ are, respectively, the desired height and horizontal velocity components used in the formation control algorithm defined by Eqs. (4.12) and (4.13). Most of the quantities in Eq. (5.25) are defined either by the formation motion command, or by the desired formation configuration. However, the formation frame expressed in UAV $i$'s flying frame $(\mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i}, \mathbf{I}_{\mathscr{F}_y}^{\mathscr{L}_i}, \mathbf{r}_{c_i}^{\mathscr{L}_i}, e_{c_i})$ has still to be computed. In this work, the information about those quantities is directly encoded into the inter-vehicle geometric constraints of the neighborhood of each UAV $i$, as explained in the following section. This allows each UAV $i$ to transform formation motion commands into individual UAV motion in its flying frame without additional inter-vehicle communication or information about its individual absolute pose, which contrasts with the work performed in past and current literature.

#### 5.2.2.1 Locally interpreting formation motion commands

For each UAV $i$ to implement the control laws in Eq. (5.25) it requires the knowledge of the position of the formation center and the attitude of the formation frame in its flying frame $(\mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i}, \mathbf{I}_{\mathscr{F}_y}^{\mathscr{L}_i}, \mathbf{r}_{c_i}^{\mathscr{L}_i}, e_{c_i})$. This can be achieved using the desired inter-vehicle geometric constraints, previously defined in the formation frame. Firstly, note that $\mathbf{I}_{\mathscr{F}_y}^{\mathscr{L}_i} = \mathbf{I}_{\mathscr{W}_z} \times \mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i}$ and $\mathbf{I}_{\mathscr{F}_z}^{\mathscr{L}_i} = \mathbf{I}_{\mathscr{W}_z}$ since both the flying and formation frames were defined with their $z$ axes aligned with the absolute frame. Therefore, only $\mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i}$ and $(\mathbf{r}_{c_i}^{\mathscr{L}_i}, e_{c_i})$ need to be computed. Secondly, note that if the desired geometric constraints between the UAVs are perfectly met, UAV $i$ can compute $\mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i}$ by rotating the desired bearing vector associated to an observed neighbor $j$ $(\mathbf{r}_{ij,d}^{\mathscr{L}_i})$ by an offset angle $\gamma_{ij}$, as shown in Fig. 5.24b. Therefore, $\mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i}$ can be expressed in the UAV $i$'s flying frame as follows:

$$\mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i} = (\cos(\beta_{ij,d} + \gamma_{ij}), \sin(\beta_{ij,d} + \gamma_{ij}), 0), \tag{5.26}$$

where $\beta_{ij,d}$ is the desired bearing angle between UAVs $i$ and $j$. Only one UAV $j$, different than $i$, is required to compute $\mathbf{I}_{\mathscr{F}_x}^{\mathscr{L}_i}$.

Thirdly, note that the position of the formation center can be expressed in the UAV $i$'s flying frame $\mathbf{x}_{fi}^{\mathscr{L}_i} = e_{c_i}\mathbf{r}_{c_i}^{\mathscr{L}_i}$ as follows:

$$\mathbf{x}_{fi}^{\mathscr{L}_i} = e_{c_i}\mathbf{r}_{c_i}^{\mathscr{L}_i} = \frac{1}{N-1}\sum_{j=1}^{N} \mathbf{x}_{ij,d}^{\mathscr{L}_i}, \tag{5.27}$$

where $\mathbf{x}_{ij,d}^{\mathscr{L}_i}$ is the desired relative position between UAVs $i$ and $j$ expressed in the UAV $i$'s flying frame.

From Eqs. (5.26) and (5.27), each UAV $i$ can have an estimate for the formation frame $x$ axis and the position of the formation center expressed in its flying frame, respectively $\hat{\mathbf{I}}_{\mathscr{F}_x}^{\mathscr{L}_i}$ and $(\hat{e}_{c_i}, \hat{\mathbf{r}}_{c_i}^{\mathscr{L}_i})$. From these estimates, UAV $i$ is able to estimate the position of the formation center and the attitude of the formation frame in its flying frame $(\hat{\mathbf{I}}_{\mathscr{F}_x}^{\mathscr{L}_i}, \hat{\mathbf{I}}_{\mathscr{F}_y}^{\mathscr{L}_i}, \hat{\mathbf{r}}_{c_i}^{\mathscr{L}_i}, \hat{e}_{c_i})$. From these estimates, each UAV $i$ can transform the received formation motion commands into desired motion in its flying frame by applying the control law in Eq. (5.25). However, while the formation is in operation, its geometric configuration can suffer distortions from the desired shape. This can happen due to noise in the UAV onboard sensors and actuators or the presence of obstacles. Such distortions cause deviations on the geometric constraints between the UAVs with respect to their desired values. This in turn causes the estimates $(\hat{\mathbf{I}}_{\mathscr{F}_x}^{\mathscr{L}_i}, \hat{\mathbf{I}}_{\mathscr{F}_y}^{\mathscr{L}_i}, \hat{\mathbf{r}}_{c_i}^{\mathscr{L}_i}, \hat{e}_{c_i})$, computed using Eqs. (5.26) and (5.27), to be different for each UAV. These differences lead to different

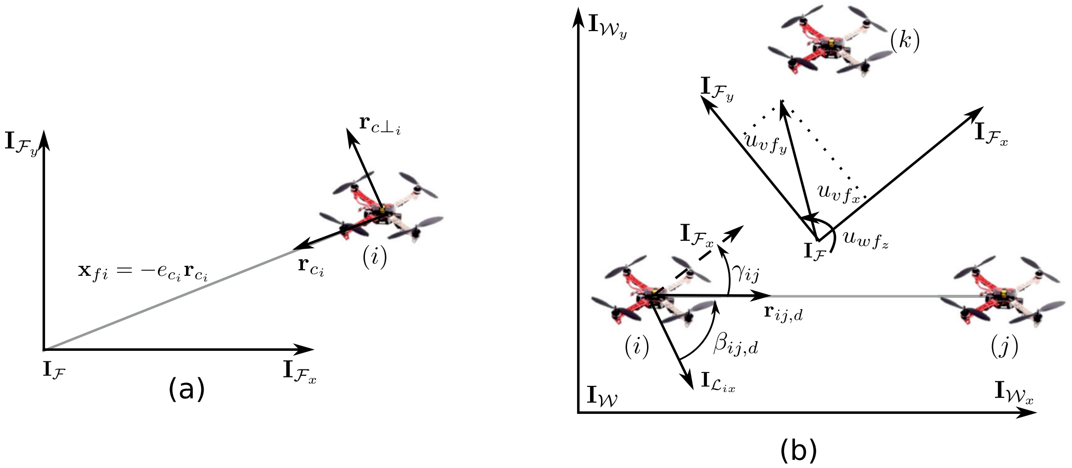

$\mathbf{I}_{\mathcal{F}_y}$
$\mathbf{r}_{c\perp i}$
$\mathbf{x}_{fi} = -e_{c_i}\mathbf{r}_{c_i}$
$\mathbf{r}_{c_i}$
$(i)$
$\mathbf{I}_{\mathcal{F}}$
$\mathbf{I}_{\mathcal{F}_x}$
(a)
$\mathbf{I}_{\mathcal{W}_y}$
$(k)$
$\mathbf{I}_{\mathcal{F}_y}$
$u_{vf_y}$
$\mathbf{I}_{\mathcal{F}_x}$
$u_{vf_x}$
$\mathbf{I}_{\mathcal{F}_x}$
$\mathbf{I}_{\mathcal{F}}$
$u_{wf_z}$
$\gamma_{ij}$
$\mathbf{r}_{ij,d}$
$(i)$
$\beta_{ij,d}$
$(j)$
$\mathbf{I}_{\mathcal{L}_{ix}}$
$\mathbf{I}_{\mathcal{W}}$
$\mathbf{I}_{\mathcal{W}_x}$
(b)

Figure 5.24: Definition of the formation motion commands. (a) Expressing the formation center in the UAV $i$'s flying frame and in polar coordinates $(e_{c_i}, \mathbf{r}_{c_i})$. (b) Defining the formation motion command in the formation frame $(u_{vf_x}, u_{vf_y}, u_{wf_z})$. Note also how the $x$ axis of the formation frame $\mathbf{I}_{\mathscr{F}_x}$ can be computed by rotating the desired bearing vector between two UAVs $\mathbf{r}_{ij,d}$ by an offset angle $\gamma_{ij}$. The $z$ axis is not displayed for illustration simplicity.

interpretations of the formation motion commands by each UAV. Fig. 5.25 illustrates an example of the previous distortion problem. In this example, $\hat{\mathbf{I}}^{\mathscr{L}_i}_{\mathscr{F}_x}$ is different than $\hat{\mathbf{I}}^{\mathscr{L}_j}_{\mathscr{F}_x}$ which will lead to different desired velocities for each UAV ($\mathbf{v}^{\mathscr{W}}_{i_h,d}$ and $\mathbf{v}^{\mathscr{W}}_{j_h,d}$) computed using the same formation motion command (as seen in Fig. 5.25b).

The previous distortions can be mitigated if each UAV $i$ acquires the quantities in Eqs. (5.26) and (5.27) by using the relative positions of neighboring UAVs that they measured using their onboard relative positioning systems. In fact, each UAV $i$ can combine all those measurements in order to compute the quantities presented in Eqs. (5.26) and (5.27) as follows:

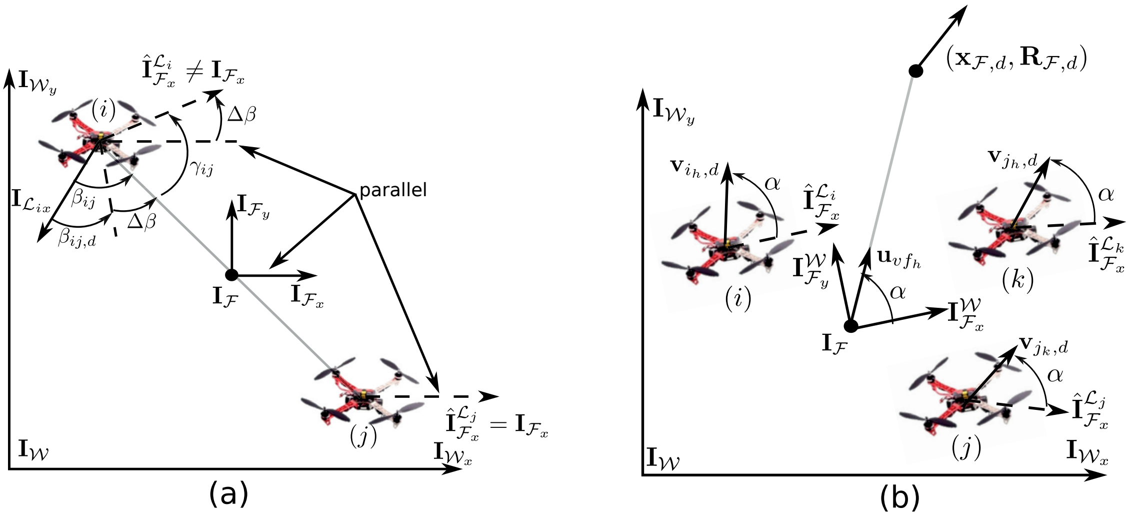
$$\hat{\mathbf{I}}^{\mathscr{L}_i}_{\mathscr{F}_x} = \frac{1}{N_i}\sum_{j\in\mathscr{N}_i}(\cos(\beta_{ij}+\gamma_{ij}), \sin(\beta_{ij}+\gamma_{ij}), 0), \tag{5.28}$$

$$\hat{\mathbf{x}}^{\mathscr{L}_i}_{fi} = \hat{e}_{c_i}\hat{\mathbf{r}}^{\mathscr{L}_i}_{c_i} = \frac{1}{N_i}\sum_{j\in\mathscr{N}_i}\mathbf{x}^{\mathscr{L}_i}_{ij}, \tag{5.29}$$

where $\mathscr{N}_i$ stands for the neighborhood of UAV $i$, $N_i$ the number of neighbors in that neighborhood, and $\beta_{ij}$ and $\mathbf{x}^{\mathscr{L}_i}_{ij}$ are respectively the relative bearing and position of UAV $j$ measured by the UAV $i$'s onboard relative positioning sensor.

#### 5.2.2.2 Formation steering in the environment

Sections 5.2.2 and 5.2.2.1 describe a way to move all UAVs of the formation in a consistent direction when a formation motion command is issued. However, the goal is to move the formation to a desired position in the environment. Since the UAVs are assumed not to rely on their absolute pose, an association



$\mathbf{I}_{\mathcal{W}_y}$
$\hat{\mathbf{I}}^{\mathcal{L}_i}_{\mathcal{F}_x} \neq \mathbf{I}_{\mathcal{F}_x}$
$(i)$
$\Delta\beta$
$\gamma_{ij}$
$\mathbf{I}_{\mathcal{L}_{ix}}$
$\beta_{ij}$
$\Delta\beta$
$\beta_{ij,d}$
parallel
$\mathbf{I}_{\mathcal{F}_y}$
$\mathbf{I}_{\mathcal{F}}$
$\mathbf{I}_{\mathcal{F}_x}$
$\hat{\mathbf{I}}^{\mathcal{L}_j}_{\mathcal{F}_x} = \mathbf{I}_{\mathcal{F}_x}$
$(j)$
$\mathbf{I}_{\mathcal{W}}$
$\mathbf{I}_{\mathcal{W}_x}$
(a)
$(\mathbf{x}_{\mathcal{F},d}, \mathbf{R}_{\mathcal{F},d})$
$\mathbf{I}_{\mathcal{W}_y}$
$\mathbf{v}_{i_h,d}$
$\alpha$
$\hat{\mathbf{I}}^{\mathcal{L}_i}_{\mathcal{F}_x}$
$\mathbf{v}_{j_h,d}$
$\alpha$
$\mathbf{I}^{\mathcal{W}}_{\mathcal{F}_y}$
$\mathbf{u}_{vf_h}$
$\hat{\mathbf{I}}^{\mathcal{L}_k}_{\mathcal{F}_x}$
$(i)$
$\alpha$
$(k)$
$\mathbf{I}^{\mathcal{W}}_{\mathcal{F}_x}$
$\mathbf{I}_{\mathcal{F}}$
$\mathbf{v}_{j_k,d}$
$\alpha$
$\hat{\mathbf{I}}^{\mathcal{L}_j}_{\mathcal{F}_x}$
$\mathbf{I}_{\mathcal{W}}$
$(j)$
$\mathbf{I}_{\mathcal{W}_x}$
(b)

Figure 5.25: Interpretation inconsistencies of the formation motion commands by each UAV. (a) Estimation inconsistencies of the attitude of the formation frame on each UAV are caused by deviations of the formation geometric configuration from the desired shape. (b) Estimation inconsistencies of the attitude of the formation frame result in different motion directions for each UAV ($\mathbf{v}_{i_h,d}, \mathbf{v}_{j_h,d}, \mathbf{v}_{k_h,d}$) computed using the same formation motion command ($\mathbf{u}_{vf_h}$). The $z$ axis is not displayed for illustration simplicity.

of the UAV moving direction with a desired goal in the environment is not possible. Therefore, this work considers an external teleoperation system (either manual or autonomous). This system has to detect the absolute pose of the virtual structure and send formation motion commands to the UAVs so that the virtual structure moves closer to its goal in the environment. The absolute pose of the virtual structure is defined by the 3D position of the formation center $\mathbf{x}^{\mathscr{W}}_{\mathscr{F}}$, and the attitude of the formation frame $\mathbf{R}^{\mathscr{W}}_{\mathscr{F}}$. The attitude of the formation frame describes the axes of that frame expressed in the world frame ($\mathbf{I}^{\mathscr{W}}_{\mathscr{F}_x}, \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_y}, \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_z}$), as depicted in Fig. 5.25b.

As previously discussed in Section 4.2, this work assumes that the external system is not able to estimate the individual UAV poses, and it has only access to a rough measurement of the formation center. This makes the external system scalable with respect to the number of UAVs. This was not the case with other approaches in the literature that individually tracked and controlled each UAV with MCS, such as in [3, 110]. Since the individual UAV poses are not measured, the previous method to measure the position of the formation center and the attitude of the formation frame, using Eqs. (5.28) and (5.29), does not apply for the external system. However, an estimator for the position of the formation center $\mathbf{x}^{\mathscr{W}}_{\mathscr{F}}$ and its velocity $\mathbf{v}^{\mathscr{W}}_{\mathscr{F}}$ in the environment can still be implemented assuming that measurements of the position of the formation center $\mathbf{x}^{\mathscr{W}}_{o\mathscr{F}}$ can be acquired by the external system. In this work, a Kalman Filter is used to implement the estimator. The states of the estimator are the position $\hat{\mathbf{x}}^{\mathscr{W}}_{\mathscr{F}}$ and velocity $\hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}}$ of the formation center in the environment. The states are propagated through time using a constant speed motion model as follows:

$$\begin{bmatrix} \mathbf{x}^{\mathscr{W}}_{\mathscr{F}}(k+1) \\ \mathbf{v}^{\mathscr{W}}_{\mathscr{F}}(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{I_3} & \Delta t \mathbf{I_3} \\ \mathbf{0_3} & \mathbf{I_3} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{\mathscr{W}}_{\mathscr{F}}(k) \\ \mathbf{v}^{\mathscr{W}}_{\mathscr{F}}(k) \end{bmatrix} + \mathbf{w}_m(k)\,. \tag{5.30}$$

After each measurement of the position of the formation center is acquired $\mathbf{x}^{\mathscr{W}}_{o\mathscr{F}}$, the estimator states can be updated using the following measurement model:

$$\mathbf{x}^{\mathscr{W}}_{o\mathscr{F}}(k) = \begin{bmatrix} \mathbf{I_3} \, \mathbf{0_3} \end{bmatrix} \begin{bmatrix} \mathbf{x}^{\mathscr{W}}_{\mathscr{F}}(k) \\ \mathbf{v}^{\mathscr{W}}_{\mathscr{F}}(k) \end{bmatrix} + \mathbf{w_o}(k). \tag{5.31}$$

Note the noise terms of each model ($\mathbf{w}_m$ and $\mathbf{w}_o$), which are modeled as zero mean normal distributions. The covariance of $\mathbf{w}_m$ is related to how inaccurately is the formation motion command executed due to the individual sensory and actuation inaccuracies of each UAV. The covariance of $\mathbf{w}_o$ is related to the uncertainty of the position measurements of the formation center acquired by the external system. The presented motion and measurement models are similar to the ones used in Eqs. (3.14) and (3.15) to estimate the individual UAV absolute position and velocity. Here, they are used to estimate the absolute position and velocity of the formation center.

Note that Eq. (5.23 actually relates the desired horizontal velocity of the formation center $\mathbf{v}^{\mathscr{W}}_{\mathscr{F}_h,d} = (v^{\mathscr{W}}_{\mathscr{F}_{x,d}}, v^{\mathscr{W}}_{\mathscr{F}_{y,d}})$ with the horizontal velocity command issued to the formation $\mathbf{u}_{vf_h} = (u_{vf_x}, u_{vf_y})$ as follows:

$$\mathbf{v}^{\mathscr{W}}_{\mathscr{F}_h,d} = \begin{bmatrix} \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_x} & \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_y} \end{bmatrix} \mathbf{u}_{vf_h}.$$

This relationship allows the external system to compute an estimate of $\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}$ using the horizontal component of $\hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}}$ acquired in the previous estimator, as follows:

$$\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}(k) = \begin{bmatrix} \hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}_h}(k) & \hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}\perp_h}(k) \end{bmatrix}^{-1} \mathbf{u}_{vf_h}(k) \tag{5.32}$$

where $\hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}_h}(k)$ is the current estimate of the horizontal component of $\hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}}$, $\hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}\perp_h}(k) = \mathbf{I}_{\mathscr{W}_z} \times \hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}_h}(k)$, and $\mathbf{u}_{vf_h}(k)$ is the current horizontal velocity command issued to the formation. Recall that $\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_y} = \mathbf{I}_{\mathscr{W}_z} \times \hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}$ and $\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_z} = \mathbf{I}_{\mathscr{W}_z}$. In order to account for noise in $\hat{\mathbf{v}}^{\mathscr{W}}_{\mathscr{F}_h}$ in this simple estimator, the angular changes of $\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}$ are averaged throughout time using a low-pass filter. When the formation is moving with its desired geometry configuration, $\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}$ and the individual interpretations of the attitude of the formation frame of each UAV $\hat{\mathbf{I}}^{\mathscr{L}_i}_{\mathscr{F}_x}$ will be consistent. However, in the presence of the previously discussed geometry distortions, the individual interpretations will not be consistent, as shown in Fig. 5.25b, but estimates for both $\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}$ and $\hat{\mathbf{I}}^{\mathscr{L}_i}_{\mathscr{F}_x}$ are still possible using the previously described estimation algorithms.

The estimation algorithm for the position and velocity of the formation center, using the models defined in Eqs. (5.30) and (5.31), and the estimation algorithm for the attitude of the formation frame, defined in Eq. (5.32), allow the external system to estimate the pose of the virtual structure in the environment. By leveraging those estimates, the external system can steer the virtual structure in the environment using formation motion commands. A teleoperator observing the formation usually estimates the pose of the virtual structure implicitly and provides commands through a joystick. However, an external autonomous system can also be implemented using the previous estimators for pose of the virtual structure, and a control law for generating the formation motion commands. For the latter case, this work explores simple control laws with the objective of minimizing the error between the current pose $(\mathbf{x}^{\mathscr{W}}_{\mathscr{F}}, \mathbf{R}^{\mathscr{W}}_{\mathscr{F}})$ and a goal pose $(\mathbf{x}^{\mathscr{W}}_{\mathscr{F},d}, \mathbf{R}^{\mathscr{W}}_{\mathscr{F},d})$ of the virtual structure in the environment. These control laws are described in Section 7.2.3.

# Chapter 6

# Experimental Setups

Several experimental setups were used for conducting the experiments of this work. These setups include different environments and UAV platforms. A sensing and control stack was developed in order to operate each UAV. The architecture depicted in Fig. 6.1 was adopted for this software stack in order to be easily adapted to the different platforms. This means that the sensory inputs, UAV actuation or the interactions with the UAVs have the same structure throughout the different UAV platforms. This chapter starts by giving an overview of this software architecture in Section 6.1, while the hardware of the UAV platforms and environments are described in Section 6.2.

## 6.1 UAV sensing and control stack

The sensing and control stack developed in this work consists of multiple software modules, as depicted in Fig. 6.1. These modules are responsible for estimating the UAV *i*'s self-localization in the environment and the relative inter-vehicle localization between UAV *i* and its neighbors, as well as controlling the UAV *i* as a single vehicle or in formation. The functionality of these modules are as follows:

- *Neighbor Pose Estimator*: This module receives the measurements from the relative positioning systems and generates an estimate of the relative inter-vehicle localization between the UAV and its neighbors. The localization and estimation algorithms used for this process are described in Section 5.1 (for both the camera-based and infrared-based systems developed in this work).

- *Self-Pose Estimator*: this module is responsible for providing accurate estimations of the vehicle absolute localization in the environment. This work assumes that the UAV can only estimate its horizontal velocity and height ($\hat{\mathbf{v}}_i^{\mathscr{L}_i}, \hat{z}_i^{\mathscr{L}_i}$) using OF and height sensors, as discussed in Section 3.4. If MCS measurements are communicated by the Offboard Processing Unit module, this module can also estimate the absolute pose and velocity of the UAV ($\hat{\mathbf{x}}_i^{\mathscr{W}}, \hat{\mathbf{R}}_i^{\mathscr{W}}, \hat{\mathbf{v}}_i^{\mathscr{W}}$).

- *Formation Controller*: this module is responsible for controlling the desired relative inter-vehicle localization between the UAV and its neighbors, necessary to achieve a desired formation configuration. For this, the controller runs a version of the formation control algorithm described

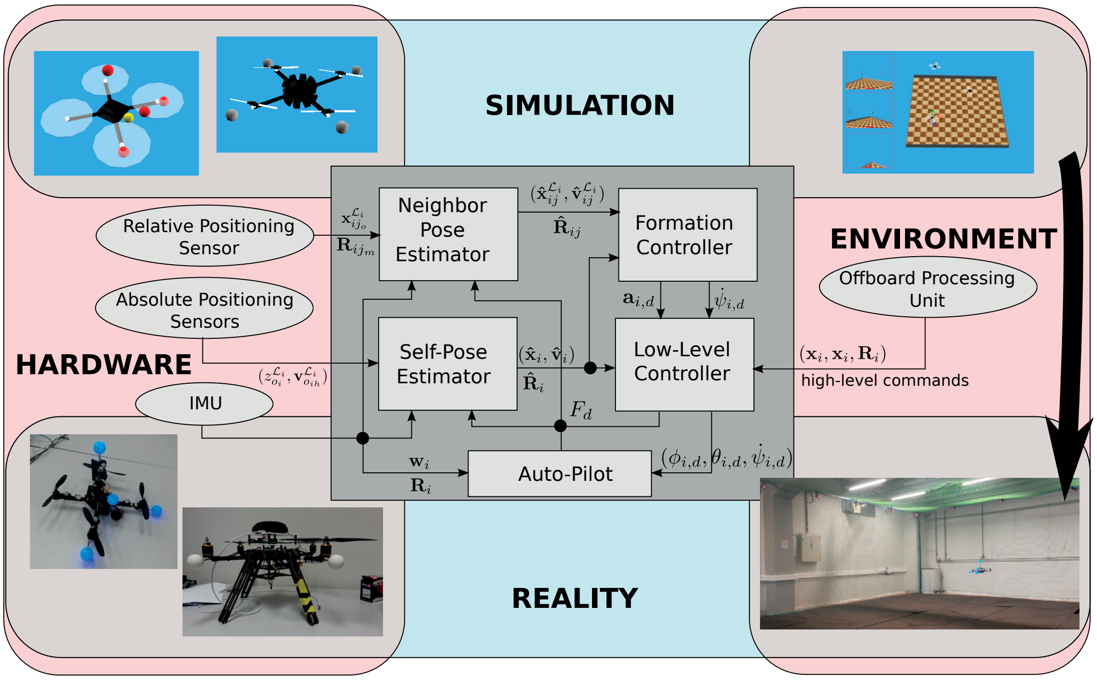

SIMULATION
HARDWARE
ENVIRONMENT
REALITY
Relative Positioning Sensor
Absolute Positioning Sensors
IMU
Offboard Processing Unit
Neighbor Pose Estimator
Formation Controller
Self-Pose Estimator
Low-Level Controller
Auto-Pilot
$\mathbf{x}_{ij_o}^{\mathcal{L}_i}$
$\mathbf{R}_{ij_m}$
$(\hat{\mathbf{x}}_{ij}^{\mathcal{L}_i}, \hat{\mathbf{v}}_{ij}^{\mathcal{L}_i})$
$\hat{\mathbf{R}}_{ij}$
$\mathbf{a}_{i,d}$
$\dot{\psi}_{i,d}$
$(\hat{\mathbf{x}}_i, \hat{\mathbf{v}}_i)$
$\hat{\mathbf{R}}_i$
$F_d$
$(z_{o_i}^{\mathcal{L}_i}, \mathbf{v}_{o_{ih}}^{\mathcal{L}_i})$
$\mathbf{w}_i$
$\mathbf{R}_i$
$(\phi_{i,d}, \theta_{i,d}, \dot{\psi}_{i,d})$
$(\mathbf{x}_i, \mathbf{x}_i, \mathbf{R}_i)$
high-level commands

Figure 6.1: Description for the UAV sensing and control stack.

by Eqs. (4.9) and (4.10) defined in Section 4.2.2. This control algorithm requires the estimates generated by the Neighbor Pose Estimator module, and it generates the desired UAV acceleration and angle rate about the $z$ axis $(\mathbf{a}_{i,d}^{\mathscr{L}_i}, \dot{\psi}_{i,d})$. This module is also responsible for computing the desired UAV horizontal velocity and height $(\mathbf{v}_{i,d}^{\mathscr{L}_i}, z_{i,d}^{\mathscr{L}_i})$ when it receives formation motion commands from the Offboard Processing Unit module in order to steer the formation in the environment. The formation motion command and the formation steering algorithm are described in Section 5.2.2.

- *Low-Level Controller*: this module is responsible for converting the desired UAV acceleration to the respective auto-pilot inputs $(F_d, \phi_{i,d}, \theta_{i,d}, \dot{\psi}_{i,d})$, as discussed in Section 3.3 and depicted in Fig. 3.2, in order to move the UAV. The desired UAV acceleration is computed by applying Eqs. (4.11), (4.12) and (4.13) using $(\mathbf{a}_{i,d}^{\mathscr{L}_i}, \dot{\psi}_{i,d}, \mathbf{v}_{i,d}^{\mathscr{L}_i}, z_{i,d}^{\mathscr{L}_i})$ provided by the Formation Controller module. The previous computations require the $(\hat{\mathbf{v}}_i^{\mathscr{L}_i}, \hat{z}_i^{\mathscr{L}_i})$ estimates provided by the Self-Pose Estimator module. In case MCS measurements are available to the UAV and a desired UAV pose in the environment is communicated, the previous controller can be replaced with the PD controller defined in Eq. (3.10) using the $(\hat{\mathbf{x}}_i^{\mathscr{W}}, \hat{\mathbf{v}}_i^{\mathscr{W}})$ estimates provided by the Self-Pose Estimator module. The Low-Level Controller module has also the ability of terminating the UAV control whenever it determines there are no flying conditions (e.g., due to low battery, total loss of height information, loss of all neighbor relative positioning information, etc.).

- *Offboard Processing Unit*: this module is responsible for providing MCS measurements (when

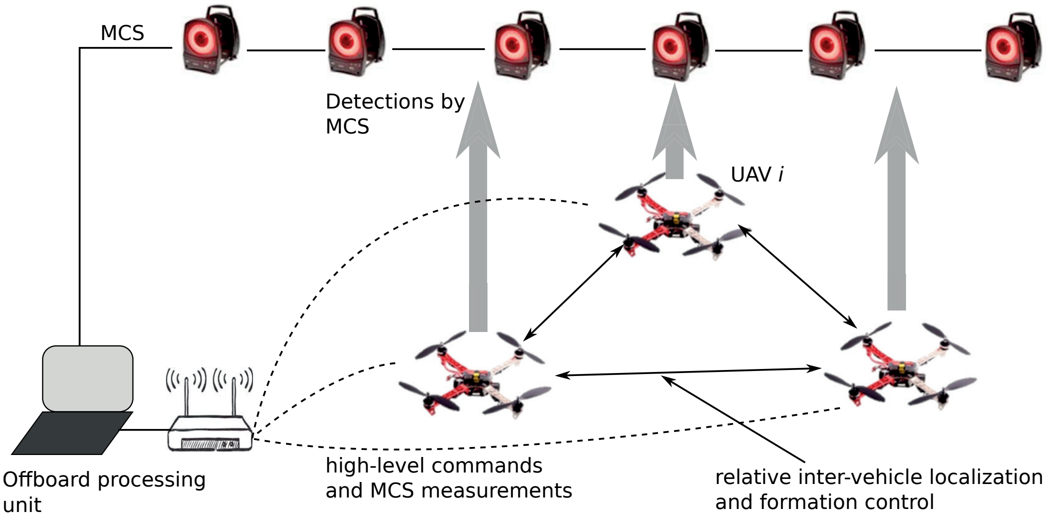

MCS
Detections by MCS
UAV *i*
Offboard processing unit
high-level commands and MCS measurements
relative inter-vehicle localization and formation control

Figure 6.2: Experimental setup sketch.

available) and high-level commands to the UAVs. The MCS measurements are provided to the UAV Low-Level Controller module, with the objective of monitoring its range to the environment limits (and terminate the control if the limits are violated), or to move the UAV to initial locations prior to the execution of the estimation and control algorithms. Additionally, these measurements are also used as localization ground truth to allow the accurate assessment of the developed systems and algorithms. Besides these previous cases, UAV control is performed without the use of MCS measurements.

The interaction between the onboard and offboard processing units is performed by communication of high-level commands using a wireless communication network. However, no inter-UAV communication is considered so as to avoid subjecting the UAV formation to inter-vehicle communication, as discussed in the previous sections. This interaction is depicted in Fig. 6.2 and is further described in the next section. Note that the communication is performed under the User Datagram Protocol (UDP) in order to provide faster transmission speeds, and to be more robust to small communication interruptions that might occur during operation. Under this protocol, occasional packet loss occurs in the network. However, since the low-level interaction between vehicles is performed without the use of communication, this problem will only impact the formation when communicating commands for group motion, with the algorithms described in Section 5.2.2.

In this work, the Robotic Operating System (ROS) framework is used on both the onboard and offboard processing units in order to achieve an easier compatibility of the sensing and control stack to different platforms. This framework provides a high-level and cross-platform inter-process communication within a Linux environment. However, this framework can cause unwanted problems with respect to system information jitter and delays since it does not provide Real Time (RT) guarantees.

### 6.1.1 Interaction between the onboard and offboard processing unit

As previously stated, the interaction between the onboard and offboard processing units is performed by communication of high-level commands using a wireless communication network. A brief description of these high-level commands is as follows.

- *formation-config*: this command is sent to all the UAVs belonging to a formation, and it provides a formation configuration file with all the desired inter-vehicle geometric constraints. These constraints allow the Formation Controller module of each UAV to compute the Laplacian matrix **L** defined in Section 4.2.2 used in the formation control algorithm. Also using these constraints, the Formation Controller module sets the desired horizontal velocity and height in the environment $(z^{\mathscr{L}_i}_{i,d}, \mathbf{v}^{\mathscr{L}_i}_{h_i,d})$ by considering that a default formation motion-command (described in Section 5.2.2) was sent with a desired formation horizontal velocity of zero, and a height of 1 *m*.

- *takeoff*: this command sets the Low-Level Controller module of each UAV into controlling the UAV in hover mode (simple height and zero velocity control).

- *operate*: this command allows the Low-Level Controller module of each UAV to consider the control outputs provided by the Formation Controller module.

- *formation*: this command starts the formation control algorithms run by the Formation Controller module of each UAV.

- *stop*: this command turns off the Formation Controller module of each UAV, and sets the Low-Level Controller module of each UAV into controlling the UAV in hover mode.

- *land*: this command starts the landing procedure executed by the Low-Level Controller module of each UAV. This procedure consists in gradually decreasing the desired height of the UAV down to zero.

- *goto*: this command gives a specific position in the environment for the UAV to follow (the desired velocity is assumed to be zero). This command is only accepted if absolute positioning information is available to the UAV, for example using a MCS. In a normal situation this information does not exist, and this command will not be accepted. This command is then useful to position the UAV in specific positions in a MCS arena prior to initializing the formation control algorithms.

- *vgoto*: this command gives a specific horizontal velocity and height for the UAV to follow, setting the desired quantities for $(\mathbf{v}^{\mathscr{L}_i}_{i_h,d}, z^{\mathscr{L}_i}_{i,d})$ in the controllers defined in Eqs. (4.12) and (4.13). This command is useful to move the leader of the formation when implementing a leader-follower approach to move the formation.

- *fgoto*: this command gives a specific formation motion-command to a group of UAVs, as defined in Section 5.2.2, to be interpreted by the Formation Controller module of each UAV.

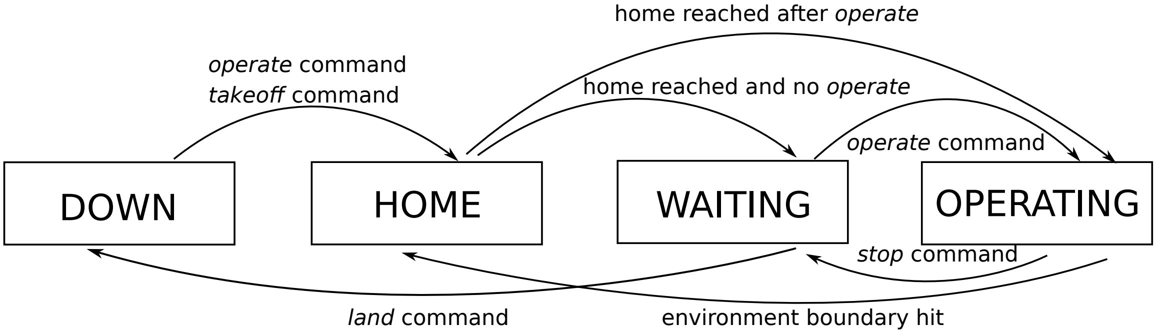


Figure 6.3: Illustration of the state machine of the UAV onboard Low-Level Controller module. Note that the emergency landing (not illustrated in the figure) can occur at any state, and it immediately overrides the state the UAV is in.

These high-level commands are managed by the Low-Level Controller module through a state machine composed of four states: DOWN, HOME, WAITING, OPERATING. The DOWN state indicates that the UAV is on the ground, and it is waiting to take off. HOME indicates that the UAV is flying and it is being controlled to reach a defined home position. Given that, as discussed in the previous sections, the available onboard positioning systems consist of height and velocity sensors, this home position is usually a hover position with a desired height. If a MCS is available to provide the full UAV absolute localization in the environment, the home position can be defined as a position in the environment. The WAITING state indicates that the UAV is stopped at a hovering position and is waiting either to be operated or to land. The OPERATING state indicates that the UAV can perform several tasks such as goal following or formation control with other UAVs. The state machine is depicted in Fig. 6.3.

Besides the high-level commands issued by the offboard processing unit, the UAV behavior is also defined by internal safety triggers that are generated by the previously described software nodes. Those triggers are generated according to the current information that is being collected online by the UAV. In this work, four main main triggers are defined as follows.

- *Absence of relative inter-vehicle localization measurements for a previously tracked neighbor*. Here, the Formation Controller module disconnects the inter-vehicle link and issues a warning. If the neighbor that was lost was defined as a formation leader, the Formation Controller module is turned off and the UAV switches to the *WAITING* state, where the Low-Level Controller module is locked in hover mode until an *operate* or *land* command is issued.

- *Absence of own height and horizontal velocity measurements*, provided by the onboard height and OF sensors, or by an external MCS. In this case, when the UAV is operating for more than a certain period without receiving these measurements, the Low-level Controller module triggers an emergency landing procedure, assuming that the onboard positioning sensors are damaged, or the external MCS is down. This procedure consists of gently turning off the motors so that the UAV lands on the ground as smoothly as possible. At this point the UAV is set on its DOWN state. No *takeoff* commands can be given to the UAV before height and horizontal information is restored.

- *Low battery*. In this case, the Low-Level Controller module immediately triggers the emergency landing procedure.
- *Environment boundaries reached*. In this case, the Low-Level Controller module orders the vehicle to return to its home position. Note that this low-level safety trigger is only available when the UAV self-localization in the environment is available using an external MCS.

## 6.2 Experimental setups

All the sensing, actuation, and high-level commands were simulated in a simulation environment to initially test the developed algorithms belonging to the sensing and control stack. After the algorithms have been validated, the sensing and control stack were deployed on the real environments and the UAV platforms. This process is depicted in Fig. 6.1. The simulated and real UAV platforms as well as environments are described in the next sections.

### 6.2.1 Quadrotor platforms

In this section, the characteristics of the quadrotor platforms used in the experiments are described. This includes a description of the onboard sensors and computational units.

**Simulated platform**

The simulated platform was implemented in the high-fidelity robotic simulator Webots[4], using a standard motion model for quadrotors, similar to the one described in Eq. (3.6). The mechanical/dynamical parameters of this model are not tuned in simulation to match the behavior observed in reality. The platform is depicted in Fig. 6.4. The sensing and control stack run in the same computer as the Webots simulator. The individual sensory devices (IMU, camera, IR emitters and receivers, OF and height sensors) and their measurements were faithfully reproduced in Webots using dedicated sensor nodes. The UAV actuation output from the sensing and control stack ($F_d, \phi_{i,d}, \theta_{i,d}, \psi_{i,d}$) is then processed by the Webots API that controls the simulated propellers to move the simulated platform. Note that noise was added to the thrust of each propeller, modeled as a zero mean normal distribution with a standard deviation of 5% of the generated thrust. Wireless communication between onboard and offboard processing units is simulated with an idealized communication network.

The noise and performance of the OF and height sensors were simulated with normal distributions according to the noise observed in the real sensors used in this work. The measured noise standard deviation of the real height sensor was 7 $cm$. The real OF was much less reliable and its standard deviation was not precisely measured (a standard deviation of 20 $cm/s$ was considered in the simulator).

[4] https://www.cyberbotics.com/

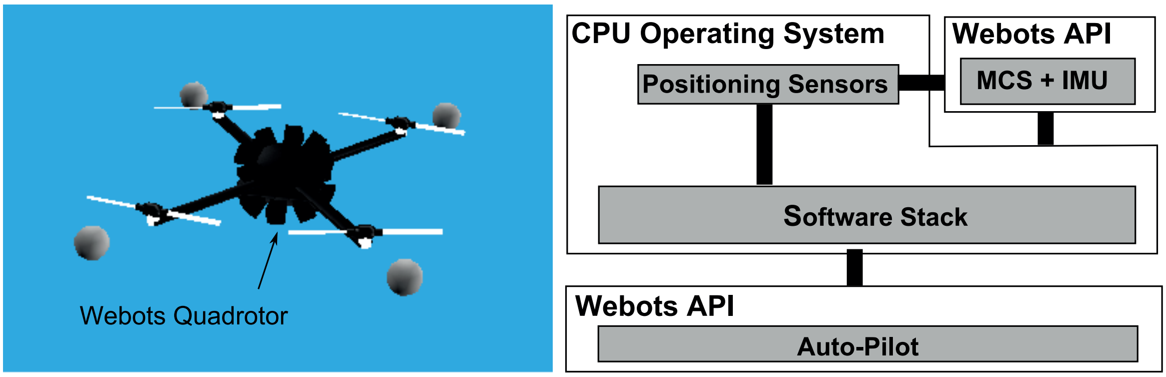


Figure 6.4: Illustration of the platform used in simulation experiments.

The relative positioning systems developed in this work, being the camera-based and the IR-based systems (see Section 5.1 for the description of those systems and Section 7.1 for their performance analysis), were also simulated as follows. Regarding the camera-based system, the active beacons from the marker were simulated as colored blobs, and an ideal camera sensor was simulated with the same resolution (320x240 pixels) and FOV measured in reality. The simulated blobs and the images taken by the simulated camera are depicted in Fig. 6.7. The simulated UAVs acquire their relative inter-vehicle localization in the previous setup using the algorithms described in Section 5.1.1. The sensor noise is generated by adding perturbations to the blob positions in the image perceived by the simulated camera. The intensity of these perturbations was set according to a maximum of 2.5 pixels around the ideal blob position. This value corresponds to the value of $\delta p$ from Eq. (5.6) found in Section 7.1.1.2.

Regarding the IR-based system, each IR beacon and IR receiver were individually modeled in simulation. Each simulated UAV was equipped with two IR receiver stations, one placed on top of the UAV body pointing upwards and the other placed on the bottom of the UAV body pointing downwards, as explained in Section 5.1.2.5. This results in thirty two IR receivers for each simulated UAV. The calibration parameters acquired for one of the IR receivers using the procedure described in Section 7.1.2.1 were used to characterize all the IR receivers of the simulation. A total of four simulated beacons were placed on each simulated UAV as explained in Section 5.1.2.5. Each IR beacon was simply modeled by its 3D position and the parameter $C_{b_{jm}}$ that allows different emission intensities for each IR beacon. The IR beacons emission was simulated with an omnidirectional profile (the quasi-omnidirectional profile discussed in Section 7.1.2.1 was ignored). The simulated UAVs acquire their relative inter-vehicle localization in the previous setup using the algorithms described in Section 5.1.2. Each IR receiver of UAV $i$'s IR-based system simulates RSS measurements of IR beacons belonging to neighboring UAVs according to the model described in Eq. (5.12). To use this model, the chosen calibration parameters and relative pose between each simulated IR receiver and IR beacon are considered. The intensity of the scalar noise for this model $\eta_{s_{ij}}$, used in Eq. (5.15), is set to the value found during the performance evaluation of the real IR-based system, in Section 7.1.2.1.

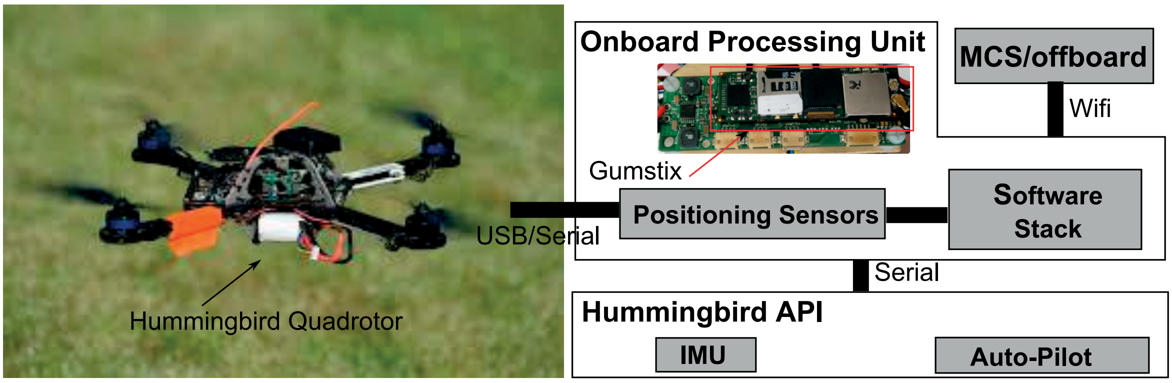


Figure 6.5: Illustration of the Hummingbird platform used in real experiments.

In order to account for calibration errors of inertial measurement units, optical flow sensors, and bearing measurements provided by the relative positioning systems, an additional noise source is inserted in the system. This noise source consists of a rotation in space of the desired accelerations computed by the formation controller in Eq. (4.13) (or Eq. (4.13) in case the FOV constraints are being considered) by a $\psi_b$ bias angle, as shown in Fig. 6.7. This bias makes the robots move in a direction that is different from the desired one.

### Hummingbird platform

The Hummingbird platform consisted of an Hummingbird quadrotor, manufactured by Ascending Technologies[5]. Its weigh is about 200 $g$ without battery and 400 $g$ with battery. The developed IR-based or camera-based relative positioning systems are placed on board the UAV platform as described in the respective sensor design sections (Sections 5.1.2.5 and 5.1.1.4). No OF or height sensors were added to this UAV platform. The noise and performance of the relative positioning systems are analyzed in Section 7.1. The developed sensing and control stack is run on the Gumstix Airstorm embedded computer[6] (5.6 $g$), as depicted Fig. 6.5. The relative positioning systems are also interfaced to this computer using an Universal Asynchronous Receiver-Transmitter (UART) protocol on an Universal Serial Bus (USB) connection in order to deliver their measurements to the respective relative inter-vehicle localization algorithms present in the sensing and control stack. The UAV actuation output from the sensing and control stack ($F_d, \phi_{i,d}, \theta_{i,d}, \psi_{i,d}$) is given to the Hummingbird auto-pilot API, which is connected to the embedded computer using an UART protocol on a serial interface. This auto-pilot then controls the UAV propellers according to the given actuation output. The embedded computer has also a Wi-Fi network module that allows it to communicate to a normal Wi-Fi network. The communication between the internal and the external processing unit (allowing the external control and monitoring of the UAV) is realized using this module.

[5] http://www.asctec.de/
[6] https://www.gumstix.com/

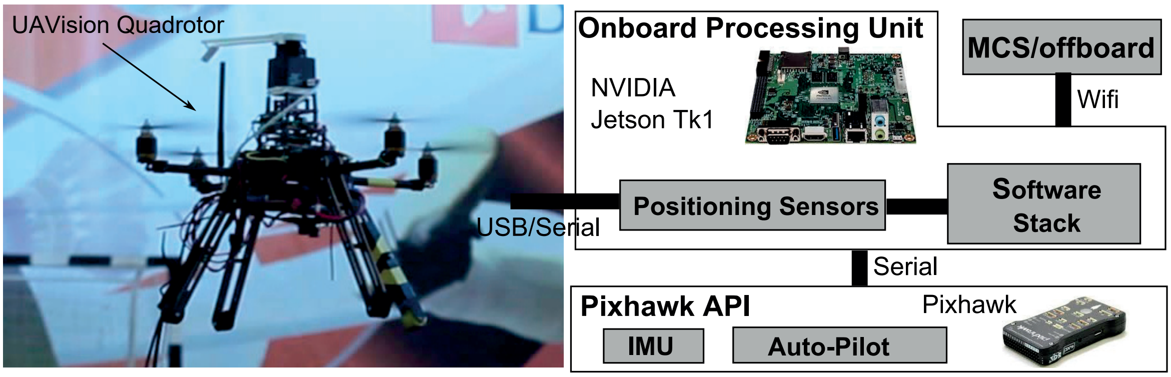


Figure 6.6: Illustration of the UX-401 platform used in real experiments.

### UX-401 platform

The UX-401 QuadCopter platform is a quadrotor manufactured by UAVision[7]. It weighs about 1200 $g$ without battery and 1800 $g$ with battery. The developed IR-based or camera-based relative positioning systems are placed on board the UAV platform as described in the respective sensor design sections (Sections 5.1.2.5 and 5.1.1.4). An OF and height sensors were added to this UAV platform in order to allow the stabilization of the UAV in a specific place in the environment without the help of any MCS. The OF used for this work was the PX4flow from Pixhawk[8]. The noise and performance of the OF and height sensors were discussed when describing the simulated platform. The noise and performance of the relative positioning systems are analyzed in Section 7.1. The developed sensing and control stack is run on the NVIDIA Jetson TK1 embedded computer[9] (143 $g$), as depicted Fig. 6.6. The relative positioning systems are also interfaced to this computer using a UART protocol on a USB connection in order to deliver their measurements to the respective relative inter-vehicle localization algorithms present in the sensing and control stack. The UAV actuation output from the sensing and control stack $(F_d, \phi_{i,d}, \theta_{i,d}, \psi_{i,d})$ is given to the PX4 auto-pilot API running on a Pixhawk auto-pilot hardware (see footnote 8), as depicted Fig. 6.6. The Pixhawk is connected to the embedded computer using an UART protocol on an USB interface. This auto-pilot then controls the UAV propellers according to the given actuation output. The embedded computer has no internal Wi-Fi module, so an external Wi-Fi dongle has been added in order to provide those capabilities to the UAV platform. The communication between the internal and the external processing unit (allowing the external control and monitoring of the UAV) is conducted using this Wi-Fi dongle, which was connected to the embedded computer using an USB interface.

Note that this sensing and control stack was shown to be applicable to different UAV platforms, without requiring a large amount of computation resources. Therefore, it can be easily adapted to many

[7] https://www.uavision.com/

[8] https://pixhawk.org/

[9] http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html

other platforms that might be considered for similar operation.

### 6.2.2 Experimental environments

The sensing and control stack was first tested in simulation. After simulation, the sensing and control stack was deployed on the real UAV platforms, described in the previous sections, and a set of real environments were used. The simulated and real environments are now described.

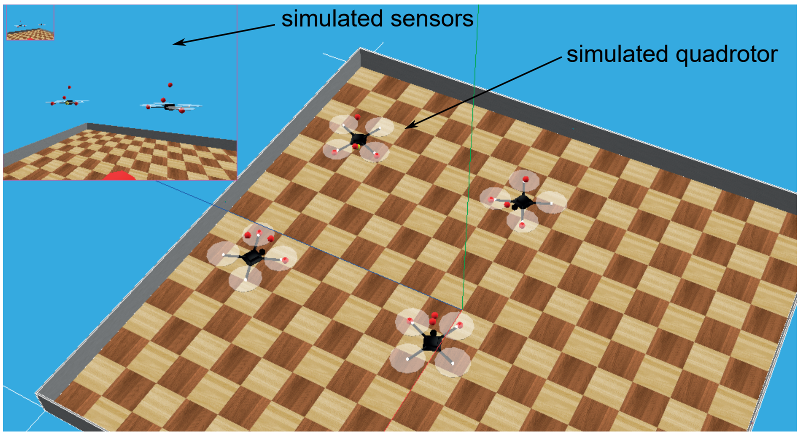


Figure 6.7: Illustration of the used Simulation environment. Note the simulated UAV platforms having a deployed camera-based relative positioning sensor.

### Simulation environment

The Simulation environment uses the high-fidelity robotic simulator Webots (see simulated platform in Section 6.2.1). The simulated platform was the one depicted in Fig. 6.4. This environment provides a convenient way of emulating a large number of UAVs operating in an environment of any size. The simulation contains the physics engine that is able to move the simulated UAV in the environment according to the actuation commands given by the sensing and control stack. The UAV motion is obtained by propagating the simulated UAV structure in the environment according to the simulated forces generated at each individual propeller (note that no fluid dynamics is included in the used propeller models). A supervisor node implemented in ROS is able to emulate the measurements of the sensors on the UAVs, and the MCS measurements.

### Maillefer flying arena

The Maillefer flying arena, depicted in Fig. 6.8b, was part of the former experimental facility run by the Distributed Intelligent Systems and Algorithm Laboratory (DISAL) at EPFL. The Hummingbird platform described in Section 6.2.1 was leveraged in this environment. This arena was equipped with a MCS in order to provide ground truth with millimetric and sub-degree accuracy of the full UAV poses. This MCS was manufactured by Motion Analysis Inc.[10] and composed of 20 Osprey cameras able to track the UAVs using a set of reflective markers in an useful volume of $4x7x2.5\ m$. The arena was large enough to both calibrate the developed relative positioning systems on board each UAV and to fly the UAVs. A wireless communication network was deployed using a Wi-Fi router placed on the arena. The Wi-Fi modules of the UAV onboard embedded computers connected to the network using this router as an access point. An IP address was then given to each UAV which could be used by the external

[10] https://www.motionanalysis.com/

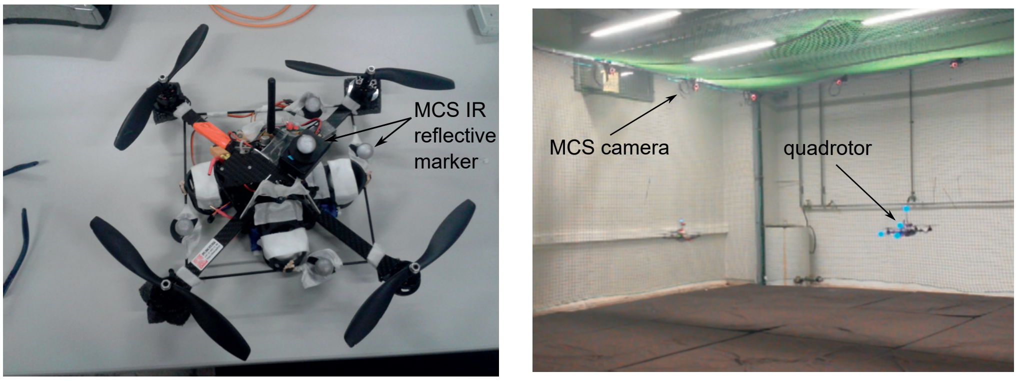


Figure 6.8: Illustration of the Maillefer flying arena. (a) The UAV platform equipped with the IR reflective markers used to localize its full 3D pose in the arena showed in (b).

processing unit to communicate to each UAV. This allowed the external processing unit to teleoperate the UAV, individually or in formation, or to monitor states internally to the UAVs for analysis.

### LBL calibration arena

The LBL calibration arena, depicted in Fig. 6.9b, is an experimental facility run by the Laboratório de Biomecânica de Lisboa (LBL) at IST. The UX-401 platform described in Section 6.2.1 was leveraged in this environment (see Fig. 6.9a for a UX-401 endowed with appropriate reflective markers). This arena was equipped with a MCS in order to provide ground truth with millimetric and sub-degree accuracy of the full UAV poses. This MCS was manufactured by Qualisys AB[11] and composed of 14 cameras able to track the UAVs using a set of reflective markers in an useful volume of $2.5x5x2.5\ m$. The arena was large enough to calibrate the developed relative positioning systems on each UAV but not to fly the UAVs themselves. In this arena, the same wireless communication network as in the Maillefer flying arena was deployed.

### Pavilion flying arena

The Pavilion flying arena, depicted in Fig. 6.9c, is the sports pavilion run by the students association at IST (AEIST). The UX-401 platform described in Section 6.2.1 was leveraged in this environment. This arena had a larger volume than the LBL calibration arena, which made it possible to fly the UAVs. However, it had no embedded MCS. An ad-hoc MCS was developed and used instead. This ad-hoc MCS consists of synchronizing and fusing height measurements provided by the height sensors on each UAV with bearing measurements acquired by an offboard static camera looking downwards to the UAVs. This method allowed the computation of the absolute position of the UAVs in the environment, which in turn enabled the computation of the relative range and elevation between the UAVs. The UAV attitudes could not be computed. A rough estimate of the relative bearing between the UAVs could be computed by

[11] https://https://www.qualisys.com//

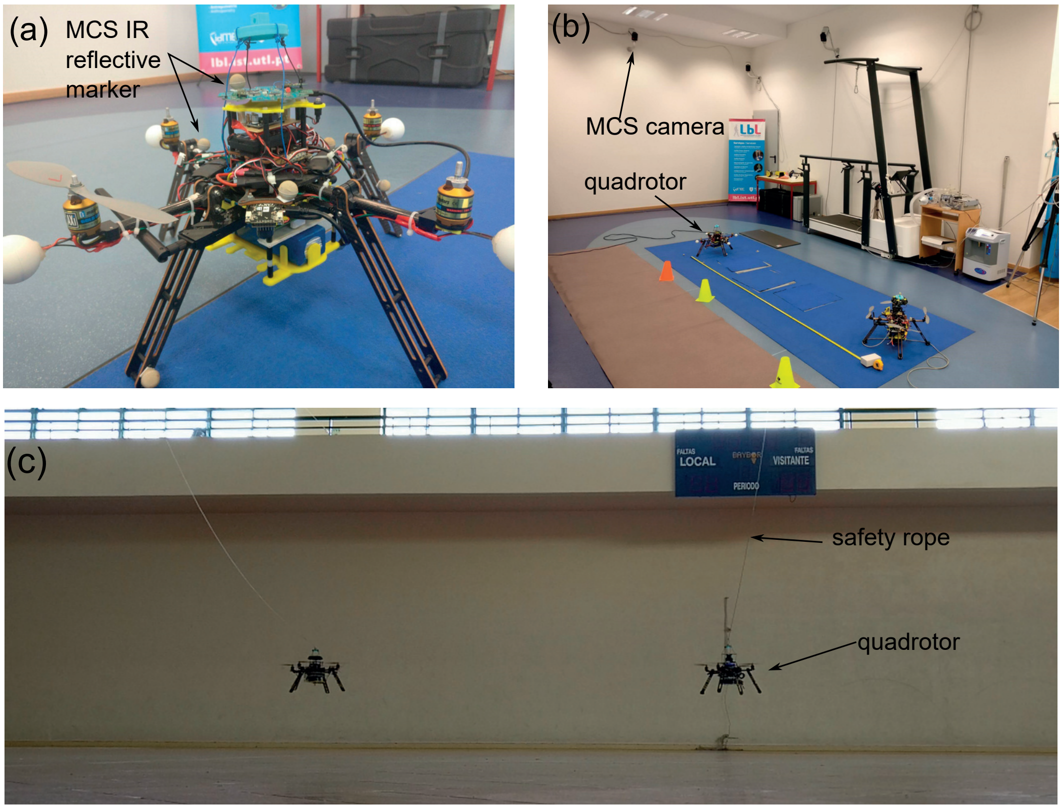


Figure 6.9: Illustration of the LBL calibration arena and the Pavilion flying arenas. (a) The UX-401 quadrotor used in these arenas. Note the IR reflective markers used to localize the quadrotor full 3D pose in the LBL calibration arena. (b) The LBL calibration arena. (c) The Pavilion flying arena.

assuming that the UAVs are always pointing to a specific direction in the environment. More details of this ad-hoc MCS can be found in Appendix D. This MCS is capable of providing measurements of the relative range and of the relative elevation between two UAVs with an accuracy of 15 *cm* and 2.7° respectively. The relative bearing measurements were only considered as rough estimates, and their performance was not assessed. In this arena, the same wireless communication network than in the Maillefer flying arena was deployed. The commands could not include the ones to move single UAVs to specific environment positions, since the ad-hoc MCS could only provide offline absolute localization information.

# Chapter 7

# Experiments and Results

In this chapter, the main results of the thesis are presented, using the experimental setup described in Chapter 6. Firstly, the performance evaluation of the developed relative positioning systems is presented in Section 7.1. This performance evaluation is used to model these sensors in a simulated environment where the developed formation control algorithms are validated (see Section 6.2.1). The characterization of these algorithms is conducted in Section 7.2. After the algorithms have been validated in simulation, they are deployed on board real UAVs along with the respective relative positioning systems, and a set of real experiments are conducted in order to assess the performance of the complete system under an inaccurate sensing and actuation scenario. These experiments are reported in Section 7.3. The chapter concludes by discussing the results acquired for the formation control system and performing an overall comparison between the two types of relative positioning systems used in this work, in Section 7.4.

## 7.1 Onboard relative positioning system performance

As previously discussed in Sections 4.2 and 5.1, two technologies are considered for the onboard relative positioning sensor in this work: IR-based and camera-based. This section presents the main results acquired on the performance of these two relative positioning systems. The performance results for the camera-based system are presented in Section 7.1.1 and for the IR-based system are presented in Section 7.1.2.

### 7.1.1 Camera-based system

To evaluate the camera-based relative positioning system performance, a set of experiments were conducted in the Maillefer flying arena using the Hummingbird platform, described in Section 6.2. The system was placed on each UAV as described in Section 5.1.1.4, and as depicted in Fig. 5.7. The camera used in the system had a resolution of 320x240 pixels and 90° FOV. The system has been designed to reliably operate at a range up to 3.5 *m*, but operation can happen at higher ranges.

The evaluation starts in Section 7.1.1.1 with the system calibration, where the set of algorithm thresh-

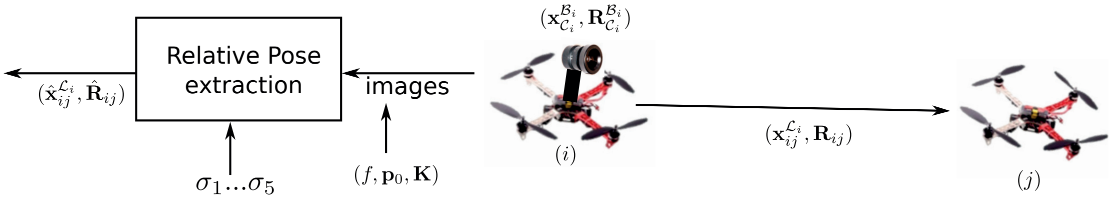

$(\mathbf{x}_{\mathcal{C}_i}^{\mathcal{B}_i}, \mathbf{R}_{\mathcal{C}_i}^{\mathcal{B}_i})$
Relative Pose extraction
$(\hat{\mathbf{x}}_{ij}^{\mathcal{L}_i}, \hat{\mathbf{R}}_{ij})$
images
$(i)$
$(\mathbf{x}_{ij}^{\mathcal{L}_i}, \mathbf{R}_{ij})$
$\sigma_1 ... \sigma_5$
$(f, \mathbf{p}_0, \mathbf{K})$
$(j)$

Figure 7.1: Overview of the algorithmic process of measuring relative pose of neighboring UAVs from camera images acquired by the camera-based system.

olds and the camera intrinsic and extrinsic parameters are calibrated. The evaluation then proceeds with the assessment of the system accuracy in Section 7.1.2.2. This assessment validates the longitudinal-transversal noise described in Section 5.1.1.3. Moreover, it shows that the system is capable of estimating the relative velocity of neighbors with high accuracy, which allow the UAV to stabilize itself without the help of other sensors. Finally, the evaluation concludes in Section 7.1.1.3 by assessing the visibility constraints of the system. The results show that the UAV can use the entire FOV of its camera to detect neighboring UAVs without precluding those neighbors from detecting the UAV (which were the requirements established in Section 5.1.1.4).

#### 7.1.1.1 Calibration

The camera-based system estimates the relative pose of neighboring UAVs expressed in the vehicle flying frame ($\mathbf{x}_{ij}^{\mathscr{L}_i}, \mathbf{R}_{ij}$), as depicted in Fig. 7.1. The algorithms used to acquire these estimates are presented in Section 5.1.1.1. These algorithms can be seen as a function $\phi$ that maps images into relative poses. This function depends on several parameters, which are: the algorithm thresholds ($\sigma_{c1}$ to $\sigma_{c5}$ defined in Section 5.1.1.1), the camera intrinsic parameters ($f$ and $\mathbf{p}_0$ defined in Section 5.1.1.3, and the distortion parameters $\mathbf{K}$) and the camera extrinsic parameters ($\mathbf{x}_{\chi_i}^{\mathscr{B}_i}, \mathbf{R}_{\chi_i}^{\mathscr{B}_i}$), all depicted in Fig. 7.1. The correct calibration of these parameters is essential for achieving the correct transformation $\phi$.

The algorithm thresholds influence the amount of false positives detected with this system. Their values are computed as follows. The thresholds $\sigma_{c1}$ and $\sigma_{c5}$ concerns the removal of blobs or configurations with blobs that are too small. Their values are manually selected by analyzing the size of the blobs on an image acquired by the camera when observing a static multi-beacon marker.

The thresholds $\sigma_{c2}$ and $\sigma_{c3}$ concern the two pruning algorithms defined in Section 5.1.1.1. Each pruning algorithm use a metric on which $\sigma_{c2}$ and $\sigma_{c3}$ apply. The metrics are: the blob spread in the image of a specific blob configuration ($h_1$) and the spread of the blob pixel count of a specific blob configuration ($h_2$). The metric $h_1$ is defined by the standard deviation of all the distances between the blobs of that configuration. This value is normalized to the averaged pixel count from all blobs of that configuration. The metric $h_2$ is defined by the standard deviation of the pixel count from all the blobs of that configuration. The maximum values of $h_1$ and $h_2$ as a function of the range between the multi-beacon marker and the camera, depicted Fig. 7.2, are computed by simulating the multi-beacon marker

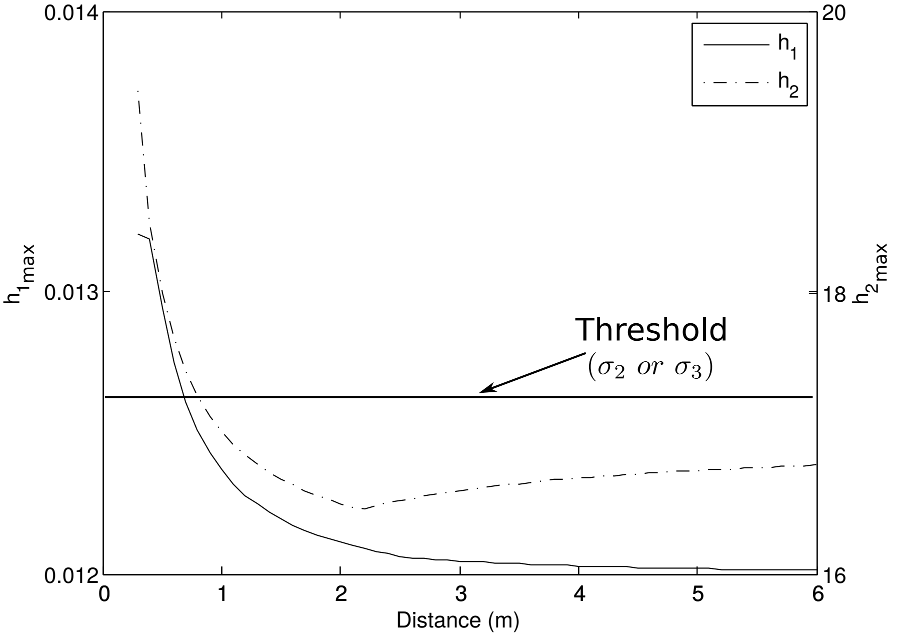


Figure 7.2: Evaluation of two metrics, $h_1$ and $h_2$, each one associated to a pruning method used in the relative pose extraction algorithm to discard in advance wrong configurations. The horizontal line represents a possible threshold that can be applied for each metric, assuming that the relative positioning system is not going to detect neighboring UAVs which are too close.

on a UAV at different 3D poses around a simulated camera. These maximum values allow the selection of $\sigma_{c2}$ and $\sigma_{c3}$, as shown in the figure.

The threshold $\sigma_{c4}$ concerns the projection errors of the solutions acquired by the P3P algorithm, described in Section 5.1.1.1. Note that these errors are in pixel positions, and can be predicted using Eq. (5.8). Therefore, $\sigma_{c4}$ is made proportional to those predictions, with a manually selected proportionality constant.

The camera intrinsic and extrinsic parameters influence how the function $\phi$ maps the image into relative poses. In this work, the $f$ parameter is divided into vertical and horizontal dimensions of the camera image ($f_x$ and $f_y$, respectively). The distortion $\mathbf{K}$ is modeled with a simple radial distortion model described in [119]. The complete set of the camera intrinsic and extrinsic parameters is denominated as $\Theta = (f_x, f_y, \mathbf{p}_0, \mathbf{K}, \mathbf{x}_{\chi_i}^{\mathscr{B}_i}, \mathbf{R}_{\chi_i}^{\mathscr{B}_i})$.

The correctness of the mapping function $\phi$ can be assessed by placing a UAV $j$ equipped with a multi-beacon marker at different poses around a UAV $i$ equipped with the camera-based system, and comparing at each pose the relative positions measured by the system ($\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i} = \phi(\Theta, image)$) with the ones measured by the MCS ($\mathbf{x}_{ij_{MCS}}^{\mathscr{L}_i}$). In this work, UAV $j$ was placed at different horizontal positions (as shown in Fig. 7.3) and at different heights, but with a constant attitude. This attitude is denominated as nominal attitude. In this way, all the beacons of the multi-beacon marker could be well observed by the camera at all the considered poses. UAV $i$ was fixed in the environment.

An optimization algorithm is then used to calibrate the values of $\Theta$ so that the error between $\hat{\mathbf{x}}_{ij}^{\mathscr{L}_i}$ and $\mathbf{x}_{ij_{MCS}}^{\mathscr{L}_i}$ is minimized for all considered poses:

$$\min_{\Theta} \sum_{l=1}^{L} ||\mathbf{x}_{ij_{MCS}}^{\mathscr{L}_i}(l) - \phi(\Theta, image(l))|| \tag{7.1}$$

where $l$ relates to one specific pose, and $L$ the total number of considered poses.

#### 7.1.1.2 Performance evaluation

The system accuracy was assessed using the model parameters computed with the calibration algorithm and the procedure described in Section 7.1.1.1. Two types of experiments were used in this assessment. The *static marker experiment* evaluated the accuracy of the relative range, bearing and elevation measurements acquired by the system, using the relative pose extraction algorithm described in Section 5.1.1.1. Additionally, this experiment is also used to observe the measuring delay and measuring frequency. The *teleoperated experiment* evaluated the accuracy of the relative velocity measurements acquired by the system. Finally, the performance achieved with this system was compared with other camera-based systems from the literature. From this comparison, a metric for analyzing the system performance was developed, which allows the choice the system design parameters according to its accuracy requirements. These experiments and results are presented below.

**Static marker experiment**

Experiment description

This experiment is similar to those performed in Section 7.1.1.1, also using the same two UAVs $i$ and $j$. An example of the different poses of UAVs $i$ and $j$ is illustrated in Fig. 7.3. The experiments were conducted with a relative range between the multi-beacon marker and the camera going from 1 *m* to 3.5 *m*. The observed multi-beacon marker was not only placed in the horizontal *xy* plane as shown in Fig. 7.3, but also at different heights. At each position, the UAV $j$ (with its deployed multi-beacon marker) was rotated around its $x$, $y$, and $z$ axes. This allowed testing the system with different relative attitudes between the camera and the multi-beacon marker. These rotations were made only for relative ranges between the camera and the multi-beacon marker going from 1 *m* to 2.5 *m*.

The relative poses between UAVs $i$ and $j$ measured by the MCS were compared with the relative poses acquired from the relative pose extraction algorithm described in Section 7.1.1. The tracking algorithm described in Section 5.1.1.2 was not turned on at this stage. Therefore, the measurements acquired by the system were expressed in the UAV $i$'s body frame. The error between the two measurements was assessed in terms of relative range, bearing, elevation and attitude.

Relative range, bearing, elevation and attitude accuracy

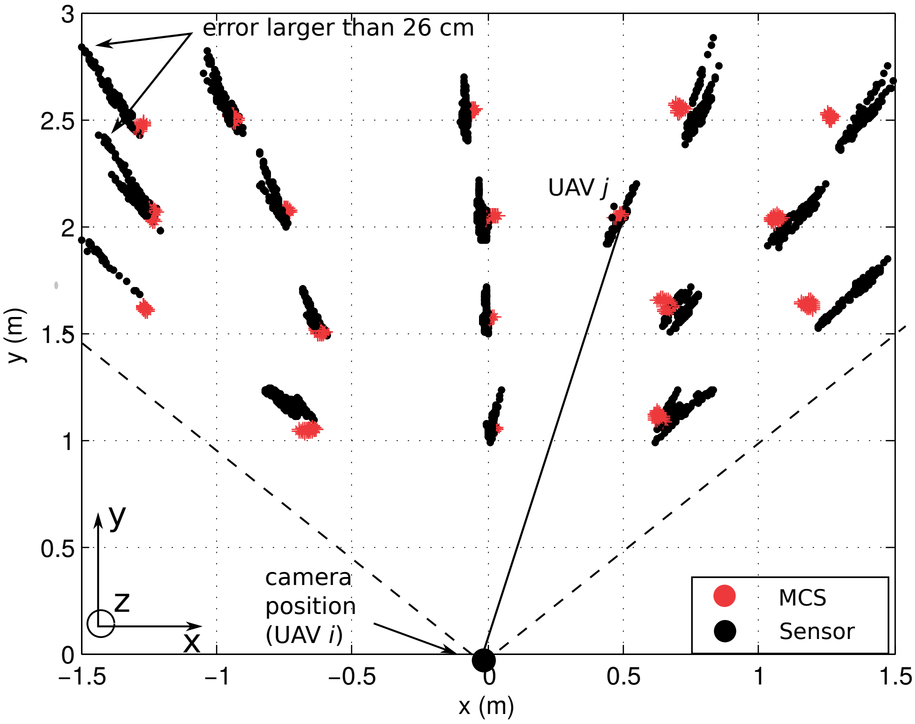


Figure 7.3: Example of an experiment used to evaluate the performance of camera-based positioning system. A UAV *j* equipped with a multi-beacon marker was placed at different positions relative to a static UAV *i* equipped with the system. At each position, UAV *j* was rotated on the spot on each axis. Note the virtual line connecting the two UAVs.

The obtained statistical data are shown in Fig. 7.4. The used dataset contains 2292 data points consisting of the errors between the measurements gathered by the relative pose extraction algorithm and the MCS. Regarding the relative range measurements, only the errors corresponding to poses of UAV *j* close to the nominal attitude were used for the statistics (1063 data points). Fig. 7.4a illustrates the norm of the 3D position error as a function of the range between the observed multi-beacon marker and the camera sensor. The results show that the maximum range errors increase with the range and are up to about 28 *cm* if the multi-beacon marker has a nominal attitude. Only 8 outliers are registered outside the 99.7% quantile of the computed data distribution, most of them with angles between 25° and 30° in the camera FOV.

Regarding the relative bearing and elevation measurements, the full dataset was used (2292 data points). The respective statistical data is shown in Figs. 7.4b and c. The figures show a relative bearing and elevation errors below 4° (for all tested ranges). The relative bearing was tested up to ±35° since the horizontal FOV of the used camera is ±45°. The relative elevation was only tested close to zero since vertical relative displacements between UAVs are not targeted given the small value for the vertical FOV of the used camera (less than ±30°). Nevertheless, it is observed that the relative elevation errors remain below 2° (for all tested ranges). Similar performance is expected for larger relative elevations.

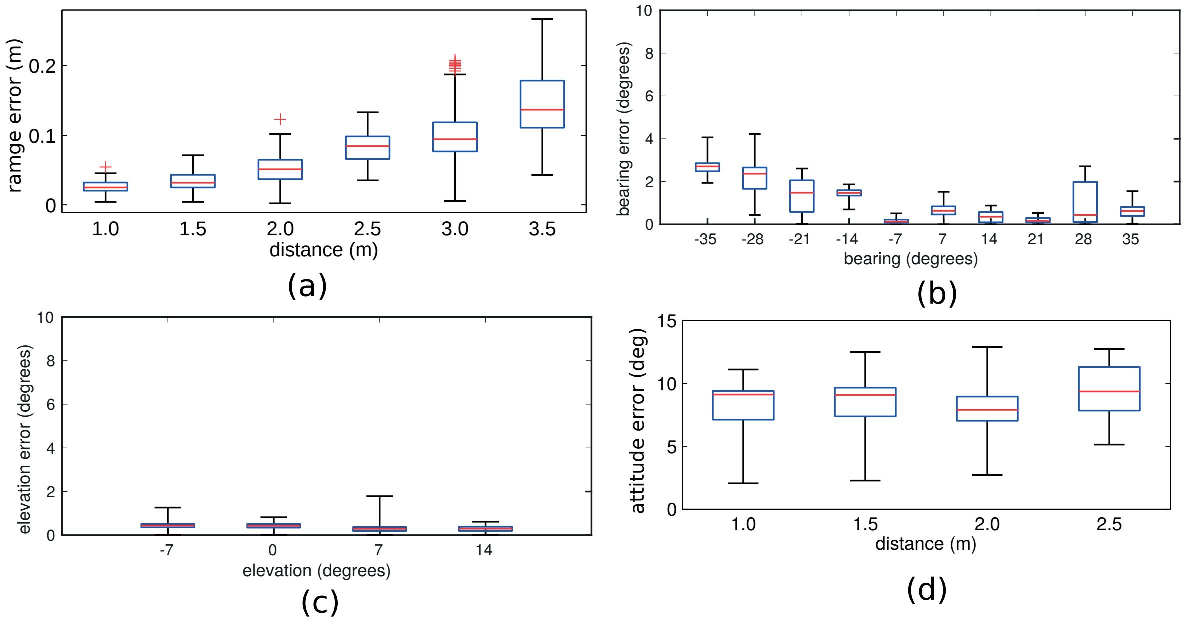


Figure 7.4: 3D position and attitude measurement errors of the tracked marker, as a function of its absolute 3D range to the static camera sensor. The relative position error is expressed in terms of relative range (a) bearing (b) and elevation (c), and the relative attitude error (d) is expressed in terms of its axis-angle form.

Regarding the relative attitude measurements, only the errors corresponding to UAV $j$ poses at a maximum range of 2.5 $m$ from the camera were used for the statistics (1229 data points), since no UAV rotations were made at ranges larger than this value. Fig. 7.4d shows the absolute angle of the attitude error expressed in the axis-angle form. The results show that the angle errors are kept below 12°, and remain roughly the same also at larger ranges. From Fig. 7.3 one can observe larger position errors than the one reported in Fig. 7.4a. This happens when the multi-beacon marker attitude is significantly far from the nominal attitude. This problem is associated to a calibration inaccuracy of either the multi-beacon marker geometry or the camera intrinsic parameters. The problem is not explored further as the maximum errors are below 10% of the range between the observed multi-beacon marker and the camera.

Sensor uncertainty model

From the previous computed position errors it is also possible to characterize the measurement uncertainty of the relative positioning system, which is defined in Eq. (5.2) as $\eta_{ij}^{\mathscr{L}_i}$. In Section 5.1.1.3 it was predicted that this model would follow a longitudinal-transversal noise model defined by Eqs. (5.6) and (5.8) (see details therein). By analyzing the positioning error data at 3 $m$ gathered in the previous dataset, it is possible to compute a longitudinal and transversal covariance of $(0.13\ m)^2$ and $(0.03\ m)^2$ respectively. This confirms the small bearing error that is assumed in this work, and therefore the correct assumption of the longitudinal-transversal noise model. The respective axes of the covariance ellipse are modeled as a linear function of the relative range between the multi-beacon marker and the camera. The parameters of this model are fitted so they have, at a 3 $m$ range, a covariance of $(0.15\ m)^2$ for the longitudinal axis, and of $(0.03\ m)^2$ for the transversal axes (close to the actual covariance values obtained for this range).

Measuring delay and measuring frequency

While acquiring the data for the experiments, it was observed that the relative pose estimation algorithm could operate with a measuring frequency of 20 $Hz$ when observing one neighbor, or 17 $Hz$ when observing two neighbors. Additionally, a comparison between ground truth and the measurements acquired by the system revealed a measuring delay of around 150 $ms$.

**Teleoperated experiment**

Experiment description

These experiments used the same two UAVs $i$ and $j$ in the same environment as in static marker experiment. However, in these experiments UAV $j$ was teleoperated in a $xz$ plane, 3 $m$ from UAV $i$ in the $y$ direction of the frame described in Fig. 7.3. UAV $j$ performed on that plane a set of vertical and horizontal movements, as described in Fig. 7.5a. Similarly to the first experiment, the UAV $i$ remained in a single static pose with its onboard camera observing the marker deployed on UAV $j$.

The relative velocities between UAVs $i$ and $j$ measured by the MCS were compared with the relative velocities acquired from the tracking algorithm described in Section 5.1.1.2. The tracking algorithm took as inputs the output of the relative pose extraction algorithm, as well as the UAV attitude measured from its onboard inertial sensors and the own thrust commands (in this case zero thrust because the UAV was static). Based on the observed measuring frequency of the relative pose estimation algorithm the tracking algorithm was activated with an operation frequency of 40 $Hz$ (setting $\Delta t$ at 25 $ms$).

Validating sensor uncertainty model

Two models are considered for the measurement noise $\eta_{ij}^{\mathscr{L}_i}$. The first model is the longitudinal-transversal model computed in the Static Marker Experiment. The second model is a simple spherical model with a diagonal covariance matrix of equal variance of $(0.15\ m)^2$. Fig. 7.5 shows the results of the velocity estimation error for one run of the experiment. The estimations acquired using the longitudinal-transversal noise model are separated from the estimations acquired using the simple spherical model. The results show that the longitudinal-transversal model drastically improves the velocity errors on the longitudinal dimension (in this case the $y$ dimension). Therefore, the use of the longitudinal-transversal model allows for accurate relative velocity measurements that enable the stabilization of a multi-UAV system without the help of any other sensing input, as shown in the experiments performed in Section 7.3.1. This analysis focuses solely on the velocity estimations since the acquired position estimations show a slightly better accuracy those shown in Fig. 7.4a, for which the observed noise was already considered sufficiently low.

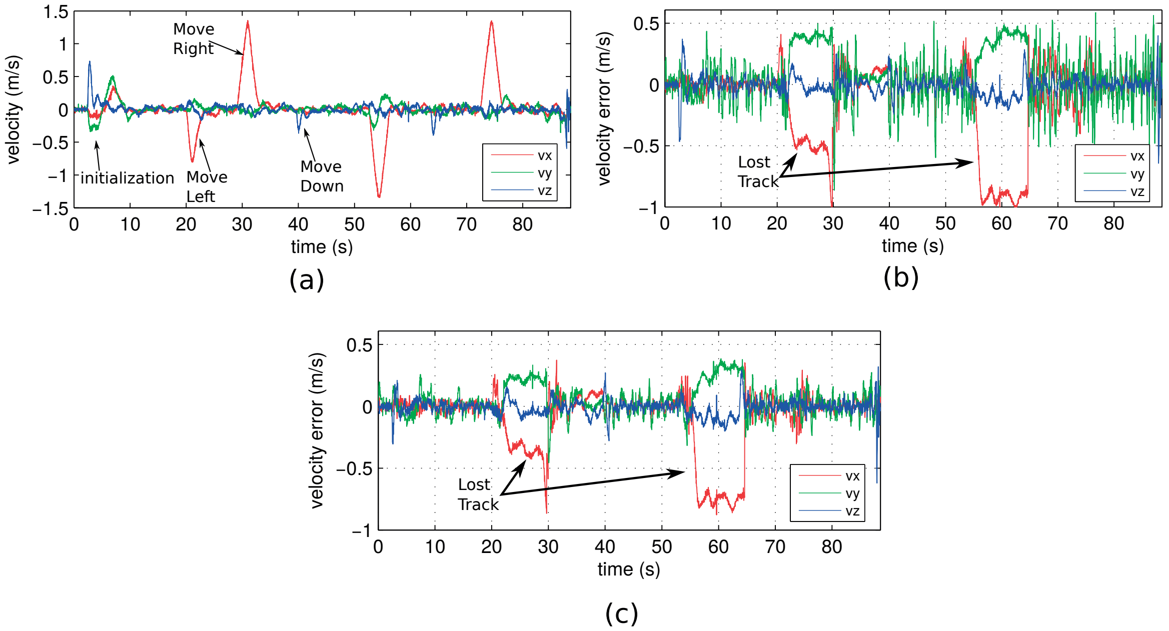

(a) (b) (c)

Figure 7.5: Performance of the relative velocity estimation of a marker at a distance of 3 *m* from the static camera-based positioning system. (a) The ground truth measurement given by the MCS. (b) Estimation error using the spherical sensor model. (c) Estimation error using the longitudinal-transversal sensor model. Note the regions where the tracking was lost.

### Comparing results

The results obtained in the static marker experiments and the teleoperated experiments are compared with works conducted in previous literature. This comparison is summarized in Table 7.1. Note that previous literature provides results using systems with different camera parameters (FOV or camera resolution). Therefore, a metric for the maximum positioning error of the system that is a function of those parameters is used for the comparison. This chosen metric is the range noise model described by Eq. (5.6). For the camera-based system of this work, this metric is used with $f$ of 220 pixels, found during calibration, and $l$ of 28 *cm* as described in Section 5.1.1.4. To compute $\delta p$ as a function of the distance to the camera sensor, a set of experiments were performed, where the multi-beacon marker was placed in front of a camera sensor at different ranges, and was shaken while its beacons were observed by the camera. The blob position of each beacon and the distance between the blobs in the image were measured. The value of $\delta p$ was set to be half of the average of the distance fluctuation observed for each blob pair, since the error was assumed to fluctuate around the correct value. This values seemed to vary little, around 2.5 pixels for the considered distances of 1 *m* to 3.5 *m*. This is because active markers produce image blobs that only change on the borders, which varies less with distance. Fig. 7.6 shows a comparison between the error predicted using Eq. (5.6) with the previous parameters, and the maximum errors of the distributions found for the accuracy performance, displayed in Fig. 7.4a, considered to be inside the 99.7% quantile.

The same metric was computed for the system used in [29], which uses a higher resolution camera, with 752x480 pixels and 90° FOV. Since this FOV is similar to the system developed in this work but using a larger camera resolution (752x480 versus 320x240), a predicted $f$ of $752/320 = 2.35$, larger than the one used in this work is computed. The values for $\delta p$ were assumed to be the same as the ones computed in this work, as an active beacon localization system was also used. However, the same experiments should be carried out on that system to be more accurate. In [29], the authors provide the circumsphere with 22 *cm* diameter for the used beacon 3D layout. The maximum values of the error were taken from their boxplots in Fig. 7a, and matched against the predicted values using Eq. (5.6) and the new values for $f$ and $l$. From Fig. 7.6, both works follow well the behavior of predicted values, although this work has measured values larger than predicted. This could be explained by the distortion of the camera that affects the value of $\theta_h$ in Eq. (5.10), with increased effect as tests are done closer from the FOV edges. As previously observed, the used datasets contained positions sufficiently far from the camera center for this to happen. From the previous results it is possible to conclude that the considered metric is a good indicator of the camera-based system accuracy limitations, even for systems with different design choices. Therefore, Eq. (5.6) and Eq. (5.10) can be used to choose the system design parameters (FOV, camera resolution, and circumsphere size) according to the accuracy requirements of the system. Moreover, the results show that the accuracy of both systems have a comparable performance within their own design limitations.

Regarding the range of the system, it was observed that the system could provide measurements reliably up to 4 *m*, using a $320x240$ pixel image and a 90° FOV. At larger ranges, the beacons start to merge with each other, precluding the detection. In [29] the system's range is larger due to larger resolution, that allow the beacons not to be merged into one at smaller ranges. In this case, the size of the beacons themselves can also be related to the maximum range of the beacon.

Regarding the measuring frequency, the results show that the system developed in this thesis achieves a lower measuring frequency during flight ($17\ Hz - 20\ Hz$) than the systems used in the previous literature (40 *Hz*). This has to do with the processing unit used on the UAV to process the images. In this thesis the Gumstix Airstorm embedded computer was used (see Section 6.2 for more details about the used Hummingbird platform). However, no information was provided in [29] about the used embedded computer. Nevertheless, it can be observed that the image size used in this thesis is 4.7 times smaller than the image size used in [29], which corresponds to an increase of the computationally efficiency by at least 4.7 times. Regarding the measuring delay of the systems, no information is provided in [29] about its value. However, its value should remain close to the one measured in this work (150 *ms*).

#### 7.1.1.3 Visibility constraints

Finally, the visibility constraints of the developed camera-based system, described in Section 5.1.1.4, are assessed. These constraints state that the relative positioning system on a UAV must allow all neighbors currently observed by the camera sensor to be able to detect this UAV. This means that the position of

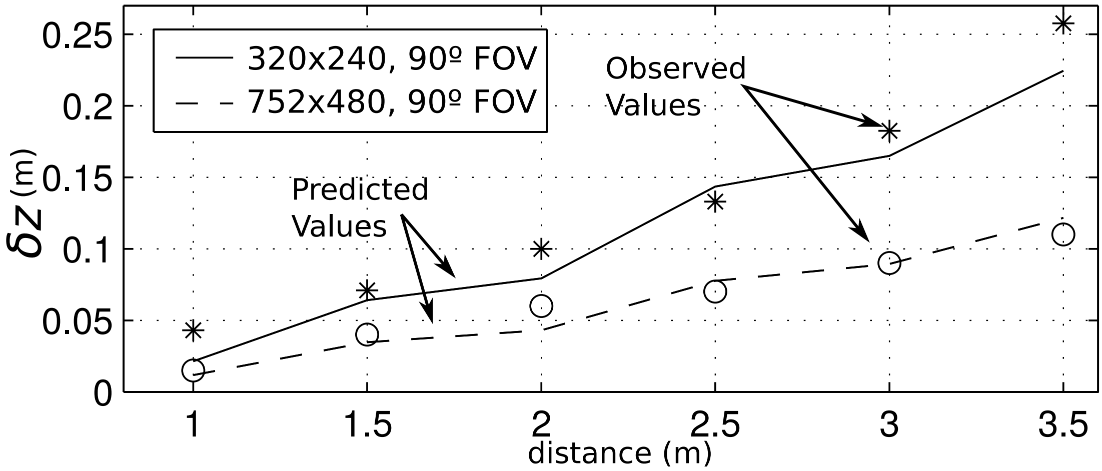


Figure 7.6: Comparison of the maximum range errors obtained using our camera and beacon setup, and the camera and beacon setup in [29]. The actual and predicted values are matched to assess the correctiveness of the chosen range accuracy metric.

the beacons must be such that the UAV can rotate itself without compromising its marker's observability from another UAV. Note that, as described in Section 5.1.1.4, the design of the multi-beacon marker already took these constraints into account. The objective of this analysis is to confirm that in fact those visibility constraints are met.

This analysis was carried out by leveraging the static marker experiments conducted in Section 7.1.1.2. In particular, this analysis leverages data points of the experiments taken when UAV $j$ was rotating around its $x$, $y$, and $z$ axes at each static position. At each static position it is noted how much UAV $j$ can rotate without compromising the detections of its onboard marker by the camera on UAV $i$. These rotations are measured with respect to the line that connects UAV $i$ to UAV $j$ (as depicted in Fig. 7.3). The visibility constraints are met if UAV $j$ can rotate more than a certain angle threshold vertically and horizontally from both sides of this line without compromising the detection. The value of this threshold is the FOV of the camera sensor used for the system, which is 45° in horizontally and 30° vertically.

In this work it was observed that UAV $j$ can rotate more than the required thresholds. When the camera is looking up to UAV $j$'s marker, UAV $j$ can horizontally rotate at least 50° from both sides of the line connecting the two UAVs. When the camera is looking down to UAV $j$'s marker, UAV $j$ can horizontally rotate more than 90° from both sides of that line. The bounds are smaller when the camera is looking up to UAV $j$'s marker because the middle beacon (the fourth beacon defined in Section 5.1.1.4) is more easily occluded. Nevertheless, since UAV $j$ could rotate more than the required thresholds, the visibility constraints of the marker were met. It is worth noticing that beyond the reported rotation limits, the marker could be still detected from most of the 3D space camera poses as inter-beacon occlusions only occur in specific poses.

Note that, in Fig. 7.5, UAV $i$'s camera loses track of the UAV $j$ at certain places. These places

|  | camera-based (320x240) | camera-based (literature) [29] (752x480) |
|---|---|---|
| FOV | 90° | 90° |
| max. range | 3.5 *m* | 5 *m* |
| accuracy errors |  |  |
| range | $< 20\ cm$ | $< 10\ cm$ |
| bearing | $< 5°$ | $< 3°$ |
| elevation | $< 2°$ | $< 3°$ |
| attitude | $< 15°$ | $< 2°$ |
|  |  |  |
| delay | 150 *ms* | ? |
| frequency | $17\ Hz - 20\ Hz$ | 40 *Hz* |

Table 7.1: Comparison between the different camera-based relative positioning systems. The displayed accuracy errors correspond to experiments conducted with ranges between the camera sensor and the target of up to 3 *m*. Note the smaller camera resolution used in this work, which explain the a smaller accuracy performance with respect to other literature, as it can be also appreciated in Fig. 7.6. Therefore, in this work accuracy was traded for computational efficiency.

corresponded to zones where the teleoperated UAV $j$ was close to the FOV edges of UAV $i$'s camera, where the distance between blobs was smaller due to the image distortion effects. It was not mentioned before that the blob detection algorithm was using a constant blob distance threshold to merge blobs that are too close to each other. Therefore, the places where UAV $i$'s camera loses track correspond to situations where the blobs of the multi-beacon marker seen by the camera were merged together by the blob detection algorithm. This could be solved if this threshold is made adaptive to the sizes of the blobs that are being merged.

### 7.1.2 Infrared-based system

To evaluate the IR-based relative positioning system performance, a set of experiments were conducted using the LBL calibration arena and the Pavilion flying arena with the UX-401 platform, described in Section 6.2. The IR-based relative positioning system was placed on each UAV as described in Section 5.1.2.5 and as depicted in Fig. 5.1.2.1. The relative positioning system has been designed to reliably operate at a range up to 3 *m* but operation can happen up to a range of 4 *m*.

The evaluation starts in Section 7.1.2.1 with the system calibration, where the calibration algorithm is shown to be able to detect and compensate deployment process inaccuracies and manufacturing tolerances of the hardware. This property allows for additional freedom in the placement of the receivers and beacons. The evaluation then proceeds with the assessment of the system accuracy in Section 7.1.2.2, considering the relative position between two UAVs, as well as their relative attitude and relative velocity. In this part of the evaluation it is shown that the accuracy of the IR-based system is comparable with the works using other IR-based positioning systems, with the advantage that it now directly can extract the

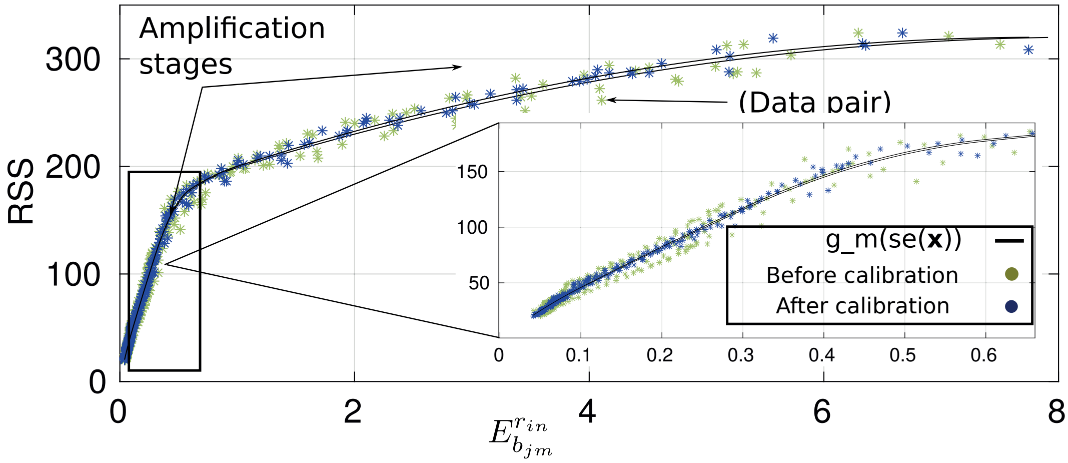


Figure 7.7: Difference between the predicted and the actual RSS of $b_{im}$ measured at a specific receiver $r_{in}$. Each data point corresponds to the actual RSS measured for a certain relative pose configuration between $b_{im}$ and $r_{in}$ measured by the MCS. The relative poses are converted to the predicted energy of $b_{im}$ absorbed by $r_{in}$ ($E_{b_{jm}}^{r_{in}}$, as calculated by Eq. (5.11)). The predicted RSS measured at $r_{in}$ as a function of the absorbed energy is given by the continuous line defining $g_{in}$. The error between these predictions and the actual RSS measured at $r_{in}$ is given by the vertical distance between each data point and the continuous line. The results are shown before and after the calibration algorithm has been applied.

relative attitude between UAVs. Moreover, the accuracy of the IR-based system is comparable with that of the camera-based system, analyzed in Section 7.1.1, but with the addition that the IR-based system has an extended FOV. Finally, this evaluation concludes in Section 7.1.2.3 by testing the extended FOV capabilities of the system with a closed-loop control experiment.

#### 7.1.2.1 Calibration

The IR-based system estimates the relative pose of neighboring UAVs expressed in the vehicle flying frame ($\mathbf{x}_{ij}^{\mathscr{L}_i}, \mathbf{R}_{ij}$). The algorithms used to acquire these estimates are presented in Section 5.1.2.2. Similarly to the previously presented camera-based system, these algorithms depend on several parameters. These parameters are the absorption coefficient function $f_{in}$, the function $g_{in}$, and the heading $\mathbf{r}_{r_{in}}^{\mathscr{B}_i}$ for each receiver $r_{in}$, as well as the parameter $C_{b_{jm}}$ that distinguishes different emission intensities of each beacon $b_{jm}$. The correct calibration of these parameters is essential for computing the correct estimates.

The procedure described in Section 5.1.2.4 was used to calibrate these parameters. The experiments needed in this procedures were conducted using two UAVs, one equipped with the four IR beacons ($j$) and the other equipped with the IR receiver station ($i$), as depicted in Fig. 5.20, in the LBL calibration arena. In these experiments, the UAVs were placed with different relative ranges, bearings and elevations between each other. The experiments were conducted for relative ranges from 0.8 $m$ to 3.0 $m$, relative bearings from 0° to 360° (full turn), and relative elevations from 0° to 75°.

To evaluate the robustness of the calibration algorithm, the experiments were conducted in the presence of misplaced receivers emulating possible deployment process inaccuracies. Fig. 7.7 illustrates the error between the predicted and the actual RSS of $b_{im}$ measured at a specific receiver $r_{in}$ obtained before and after calibration. The used receiver was deployed in the receiver station with its attitude different from its theoretical value. More concretely, its elevation was found to be of 55° instead of the theoretical 45°. These inaccuracies can lead to a large data point dispersion around function $g_{in}$, corresponding to large errors between the predicted and the actual RSS measurements. This behavior can be appreciated in the expanded part of Fig. 7.7 for the data points computed with the non-calibrated parameters. It can be observed that after the calibration algorithm has been applied the data point dispersion substantially decreases. This is due to the newly estimated elevation for the receiver $r_{in}$ of 59°, which is substantially closer to its actual value than the theoretical 45°. This shows that the deployment process does not need to be strict as inaccuracies are corrected with the developed calibration algorithm.

Finally, the homogeneity properties of the emission profile of the IR beacon were tested with a set of experiments. These experiments consist of measuring the RSS of the beacon of UAV $j$ acquired by one of the receivers of the static UAV $i$. The measurements were taken while the beacon rotates horizontally on the same place, as depicted in Fig. 7.8. Before starting its rotation, the beacon was placed with its beacon heading directly pointed to the chosen receiver. The light ray formed between the beacon and receiver should be such that its incidence angle on the receiver (angle between the receiver heading and the light ray) is zero. Afterwards, the beacon was horizontally rotated clockwise and counterclockwise on the spot (maintaining the same light ray formed between the beacon and receiver). The angle between the light ray and the new beacon heading ($\theta$ in Fig. 7.8) is then measured and related to the measured RSS, as shown by the plots of Figs. 7.8a and b. These plots show that as $\theta$ increases there is a slight RSS decay, which can reach up to 15% for $\theta$ values of 120°. This decay was attributed to reflections that beacon produces on the structure that is supporting it. When $\theta = 0$, the support structure is right behind the beacon, and all reflected light rays have directed line of sight with the receiver. As $\theta$ increases, less reflected light rays have direct line of sight with the receiver. However, this signal decay is considered small, and it was not compensated in the RSS models. Therefore, these results show that the IR beacon has a quasi-homogeneous signal intensity on all the directions where the beacon can be properly detected. These experiments were also used to dimension the value of the RSS noise term $\eta_{s_{ij}}$ defined in Eq. (5.15). Its value was set to 2% of the measured RSS signal.

Additional experiments were conducted in order to assess how these reflections impact the performance of the system in different environment conditions. It was observed that while moving the UAV in the LBL calibration arena (depicted in Fig. 6.9b) the RSS of its own beacon $b_{im}$ measured by its onboard receivers would increase when $b_{im}$ would start approaching a wall with a range less than 50 *cm*. This effect is in accordance with what was observed in systems using the same technology [92], and it can be used by the UAV itself in order to detect and avoid incoming walls. However, this effect also modifies the results of the localization algorithms. It was observed that a beacon that close to walls would also alter the UAV relative range and bearing measurements acquired by a neighboring system by 10 *cm* and 5° respectively.

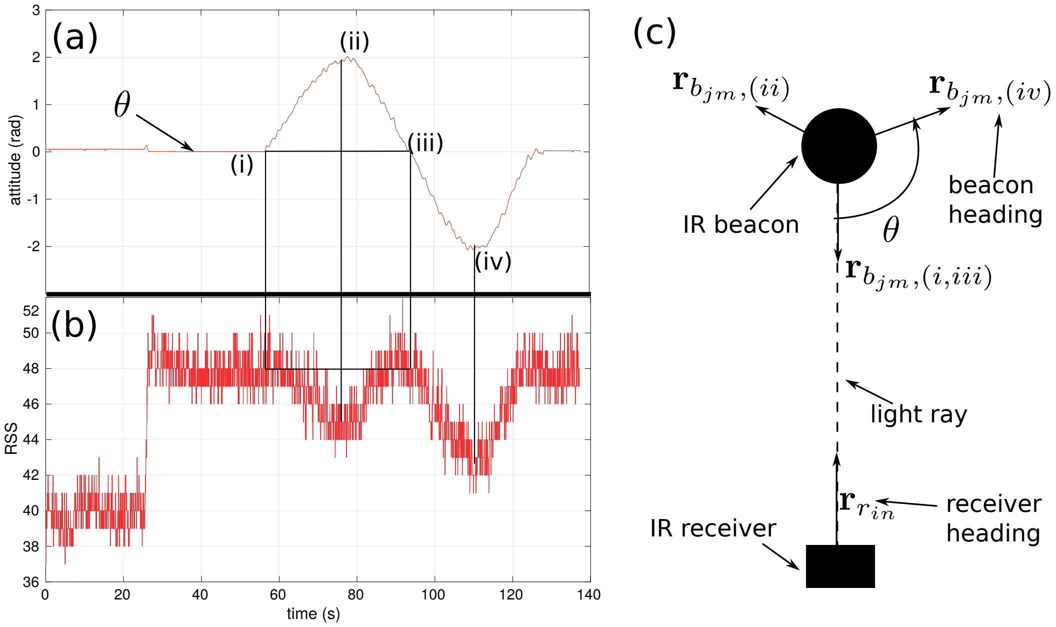


Figure 7.8: Assessing the homogeneity properties of the IR beacon emission profile. (a) Angle formed between the beacon heading and the light ray formed between the beacon and the receiver ($\theta$) during the rotation of the beacon. (b) The respective RSS measured at each $\theta$ value. (c) Illustration of the beacon rotation and the angle $\theta$.

Other experiments were also conducted in the Pavilion flying arena. This environment has a different geometry than the LBL calibration arena in the sense that the distance between the floor and the ceiling was substantially larger. When the system measured the RSS light coming from the beacons of a marker, the measurements acquired by the receivers pointing upwards (to the ceiling) were smaller than the ones acquired in the same conditions in the LBL calibration arena. This effect was due to ceiling reflections in the LBL calibration arena that did not exist in this new environment. This resulted in the system acquiring the relative UAV elevation with a value of about 15° lower than what was expected. Under these conditions, a recalibration step is advised. For this step, a small amount of RSS measurements are acquired at different relative positions between the IR-based system and the marker in the new environment. The calibration step described by Eq. (5.16) is then repeated with the acquired data in order to recalibrate the RSS gain functions of each individual receivers. It was observed that there is no need to apply the calibration step described by Eq. (5.17) to recalibrate the receiver headings. After this recalibration step, the relative positioning system accuracy was restored to the values acquired in the LBL calibration arena.

#### 7.1.2.2 Performance evaluation

The system accuracy was assessed using the model parameters computed with the calibration algorithm and the procedure described in Section 7.1.2.1. Two types of experiments were used in this assessment.

The *positioning assessment experiment* evaluated the accuracy of the relative range, bearing and elevation measurements acquired by the system, using the relative pose extraction algorithm described in Section 5.1.2.2. Additionally, this experiment is also used to observe the measuring delay and measuring frequency. The *estimation assessment experiment* evaluated the accuracy of the relative attitude and relative velocity measurements acquired by the system. Finally, the performance achieved with this system was compared with other systems from the literature, including the camera-based systems.

**Positioning assessment experiment**

Experiment description

These experiments are similar to the ones performed in Section 7.1.2.1, also using the same two UAVs *i* and *j* in the LBL calibration arena. The two UAVs are placed at different relative range, bearing and elevations. The relative ranges and elevations are changed by moving UAV *j* through a line towards UAV *i* while moving up and down. The relative bearings are changed by horizontally rotating UAV *i* on the spot. The experiments are conducted for relative ranges from 0.8 *m* to 3.0 *m*, relative bearings from 0° to 360° (full turn), and relative elevations from 0° to 75°. The experiments conducted to assess the system performance use all four beacons turned on. This creates a multi-beacon marker on UAV *j* that can be localized by UAV *i*. To evaluate the robustness of the localization algorithm, the experiments are conducted in the presence of two damaged receivers (one deactivated, and another wrongly calibrated) placed in the section of the receiving station corresponding to the bearing indicated in Fig. 7.9 (the deactivated receiver had a vertical heading of 0°, and the wrongly calibrated receiver had vertical heading of 45°).

The relative poses between UAVs *i* and *j* measured by the MCS were compared with the relative poses acquired from the localization algorithm described in Section 5.1.2.2. The tracking algorithm described in Section 5.1.2.3 was not turned on at this stage. Therefore, the measurements acquired by the system were expressed in UAV *i*'s body frame. The error between the two measurements was assessed in terms of relative range, bearing and elevation.

Relative range, bearing, and elevation accuracy

The obtained statistical data is shown in Fig. 7.9. Regarding the range measurements, the results show an error increase with the range between the marker and the sensing UAV. A small exception is seen between 1 *m* and 1.4 *m* where this error increases more than expected. This fact is attributed to the calibration algorithm that tries to reduce error at higher ranges by slightly compromising the errors at smaller ranges. However, the absolute maximum errors are equal or less than 10%. Regarding the bearing measurements, results show maximum errors of 5° in the zones corresponding to the damaged receivers. Note that when the marker bearing is such that the damaged receivers are used for the estimation (which is around the selected damaged section of the receiving station), one can observe a bearing error increase up to 15°. Regarding the elevation measurements, results show maximum errors around

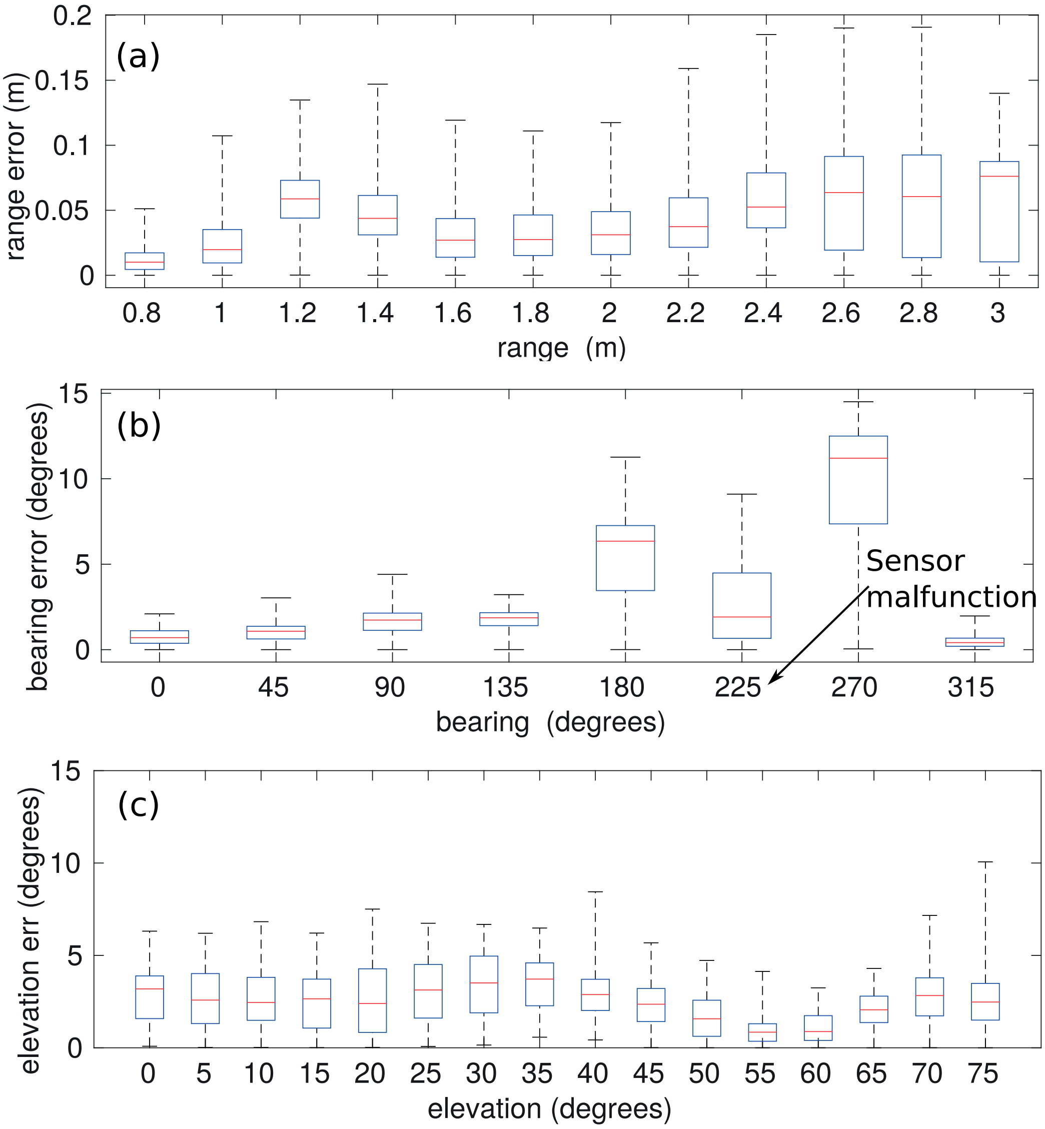


Figure 7.9: Relative range (a) and relative bearing (b) and relative elevation (c) measurement accuracy of the IR-based system measuring a marker at different static relative positions. The two malfunctioned receivers were placed at the section of the receiver station with 225° bearing with respect to the UAV's body frame.

6°, with the exception of higher elevations, which show larger maximum errors around 10°. The larger errors observed at higher elevations can be related to the fact that there is no receiver at a vertical heading of 90°.

Measuring delay and measuring frequency

Note that the measuring frequency observed by the IR-based system was of 80 $Hz$, which is close to the measuring frequency for which it was designed, 83 $Hz$, as explained in Section 5.1.2.5. Additionally, a comparison between the ground truth and the measurements acquired by the system revealed a measuring delay smaller than 50 *ms* (a precise measurement of this delay was not possible since its value was too small for being measured reliably).

**Estimation assessment experiment**

Experiment description

These experiments also used the same two UAVs $i$ and $j$ in the LBL calibration arena. UAV $i$ is static in the environment while UAV $j$ moves around. The trajectories made with UAV $j$ include a 360° rotation of the UAV, in order to assess the accuracy of the relative attitude measurements, and segments with different velocity profiles, in order to assess the accuracy of the relative velocity measurements.

Similarly to the previous set of experiments, the relative poses between UAVs $i$ and $j$ measured by the MCS were compared with the relative poses acquired by the localization algorithm described in Section 5.1.2.2 with the tracking algorithm described in Section 5.1.2.3 turned on. This enabled the error assessment to be made not only on the relative position and relative attitude, but also on the relative velocity. The tracking algorithm takes as inputs the output of the relative pose extraction algorithm, as well as the UAV attitude measured from its onboard inertial sensors and the self thrust commands (in this case zero thrust because the sensing UAV is static). The tracking algorithm was activated with an operation frequency of 40 $Hz$ (setting $\Delta t$ at 25 *ms*), the same as the one used in the developed camera-based system. The measurements produced by the tracking algorithm are expressed in UAV $i$'s flying frame.

Relative attitude and relative velocity accuracy

Fig. 7.10a illustrates one trajectory performed with UAV $j$ containing a 360° rotation of the UAV (at around 70 $s$ of the experiment) and a relative elevation section up to 40° (at around 30 $s$ and 60 $s$ of the experiment). The absolute maximum relative elevation error observed in this experiment is of 3°, which is within bounds of the accuracy measured in the previous experiment. Regarding the relative attitude measurements, Fig. 7.10b shows that the marker rotation is fully captured by the IR-based system. Fig. 7.10c, shows a constant estimated attitude error of about 10° and an absolute maximum error of 23.5°. The constant error is most likely related to the previously measured beacon signal decay (see Fig. 7.8), which was not compensated. Finally, note that during marker rotation the relative 3D position remained accurately estimated. Note that the relative attitude measurements were not possible to be acquired with the IR-based systems used in previous literature without extra communication overhead

Fig. 7.11 illustrates the error between the relative velocity measurements provided by the system and the respective measurements provided by the MCS, during the trajectory depicted in Fig. 7.10. The error of the velocity measurements acquired by the system for the three axes of UAV $i$ flying frame is kept under 0.4 $m/s$, as shown in Fig. 7.11b. Note that several velocity intensities were applied to UAV $j$ during its trajectory as depicted in Fig. 7.11a. These results are comparable with the relative velocity accuracy of the camera-based system developed in this work, for which the results are presented in Fig. 7.5.

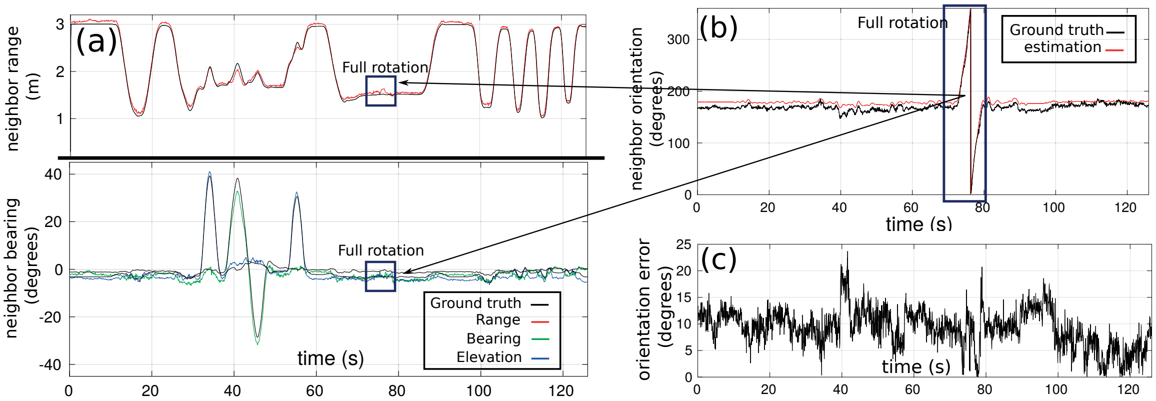


Figure 7.10: Relative pose measurements of a moving marker acquired by the IR-based system at a static position. (a) Range (red), relative bearing (green) and relative elevation (blue) estimations compared to the ground truth (black). s(b) Relative attitude (red) compared to the ground truth (black). (c) Measurement error with respect to the ground truth of the data in (b). Note the time period when the marker full rotation was performed (black boxes).

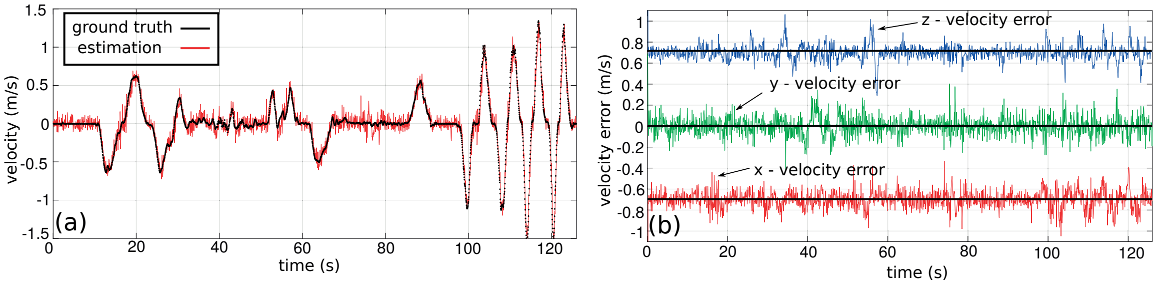


Figure 7.11: Relative velocity measurements of a moving marker acquired by the IR-based system at a static position. (a) Relative velocity measured in the $x$ axis of the UAV flying frame, (red) compared to the ground truth (black). (b) Relative velocity measurement error in the $x$ (red), $y$ (greed), and $z$ (blue) axes of the UAV flying frame. Note the horizontal continuous black lines defining the zero error line for each axis.

### Comparing results

The previous results are compared with the ones obtained for works conducted in previous literature. This comparison is summarized in Table 7.2. Note that the table only includes systems that provide three dimensional measurements, namely [92]. One can observe that the accuracy of the IR-based system developed in this thesis is comparable to the one reported in [92]. Regarding the relative range accuracy the absolute maximum errors are equal or less than 10%, which is comparable to other works that use IR-based systems [86, 92]. Regarding the bearing measurements, results show maximum errors of 5° for the zones that did not require the usage of the damaged receivers. These errors are similar to those obtained in [86, 92], which is expected since both these systems use the same $45^{circ}$ heading spacing between horizontal receivers. For zones requiring the usage of the damaged receivers, the maximum bearing errors are 15°, showing that the adjacent sections are still able to compensate for this failure situation (with expected accuracy degradation). Regarding the elevation measurements, results show maximum errors around 6°, with the exception of higher elevations, which show larger maximum errors around 10°. These errors are slightly higher than in [92], which is expected since the IR-based system contained additional receivers with a heading of ±90°. Finally, note that an advantage of the developed IR-based system is that it now can directly extract the relative attitude between UAVs.

Regarding the range of the system, it was observed that the system could provide measurements reliably up to 4 *m*. This value is smaller comparing with [92] mainly due to the number of cascade amplifiers used in the reception side (as explained in Section 5.1.2.2), which is less than the ones used in [92]. This was chosen for the simplicity of the hardware design, but can easily be expanded with minimum extra weight in order to increase the sensor range.

Regarding the measuring frequency, the results show that the system developed in this thesis has a lower measuring frequency (80 *Hz*) than the systems used in the previous literature (200 *Hz*). This has to do with the fact that the system developed in this thesis uses four independent beacons to identify each UAV, and that six different UAVs can be simultaneously detected by the system. In previous literature [86, 92], each vehicle was identified as a single emission source. If this also is considered for the developed system (and in this case no attitude measurements could be acquired), the same six UAVs could be identified with four times less beacons. According to Section 5.1.2.5, this would allow to reshape the $N_s$ to a value four times smaller than the one that is being used (60), and boost the frequency towards 333 *Hz*. In fact, if only one beacon is being tracked, $N_s$ could be reshaped to a value of 3 (theoretically it could be 2, but an additional slot is introduced for safety) and further boosting the frequency towards 1.66 *KHz*, which is 1.66 times larger than the maximum value predicted for the systems used in the literature (1 *KHz* as reported in [92]). The frequency could also be further increased by tuning $N_c$ and $T$, described in Section 5.1.2.5. The shortest value for $T$ is already chosen according to the responsiveness of the used electronic filters. However, $N_c$ could be decreased since that less information needs to be precessed with less beacons. However, this parameter remained always the same in this work. Regarding the delay of the systems, no information is provided in [92] about its value. However, this value should remain close to the one measured in this work ($< 50$ *ms*).

The accuracy of the IR-based system is comparable with the one of the camera-based system presented in this work, for which the results are presented in Section 7.1.1. Systems with higher camera resolutions, such as (752x480) in [15, 29], have greater accuracy but at the expense of higher computational cost. This prevents them from having larger FOVs on resource-constrained robots such as small-scale UAVs. The proposed IR-based system not only is shown to have a 360° FOV, but it also provides higher measuring frequencies than the camera-based systems. This system currently runs at 80 *Hz*, which is substantially higher than the maximum of 40 *Hz* in the previous described camera-based systems. Additionally, a comparison between the ground truth and the measurements acquired by the system revealed a measuring delay for the developed IR-based system of less than 50 *ms*. This value is much lower than the one acquired for the developed camera system (150 *ms* as seen in Section 7.1.1.2) given the larger measuring frequencies of the IR-based system.

|  | IR-based | IR-based (literature) [92] |
|---|---|---|
| FOV | 360° | 360° |
| max. range | 4 $m$ | 12 $m$ |
| accuracy errors |  |  |
| range | $< 20\ cm$ | $< 20\ cm$ |
| bearing | $< 5°$ | $< 5°$ |
| elevation | $< 10°$ | $< 3°$ |
| attitude | $< 20°$ | unavailable |
|  |  |  |
| delay | $< 50\ ms$ | ? |
| frequency | $80\ Hz - 1.66\ KHz$ | $200\ Hz - 1\ KHz$ |

Table 7.2: Comparison between the different IR-based relative positioning systems. The displayed accuracy errors correspond to experiments conducted with ranges between the sensor and the target of up to 3 $m$. The developed system can compute the relative attitude of neighboring vehicles and potentially provide larger measuring frequencies than the systems in the literature.

#### 7.1.2.3 Assessing extended FOV

The 360° horizontal FOV capabilities of the system were assessed with an experiment with the same two UAVs $i$ and $j$ that were described in the previous experiments. However, this type of experiment was conducted in the Pavilion flying arena. In this experiment, UAV $i$ starts by autonomously taking off up to a certain height, and then it has to maintain a desired range (in this case it was 2.26 $m$ as shown in Fig. 7.12) to the static UAV $j$ while rotating on the spot at the specified height. The flying UAV had an onboard sonar to control its height and an onboard OF sensor to control its velocity in the environment. The relative range to the static UAV was controlled using the measurements from the IR-based system. The closed-control algorithm deployed on UAV $i$ is the baseline formation controller, described in Section 6.1.

As described in Section 6.2.2, the ad-hoc MCS installed in the environment used in this experiment can only acquire accurate measurements of the relative range and elevation between the two UAVs. Therefore, in this experiment, only the data from the IR-based system is displayed to show how the measured relative bearing evolves during the rotation experiment, as well as the measured relative attitude. The results are displayed in Fig. 7.12. Fig. 7.12a shows the estimated range stabilizing around the desired value, meaning that UAV $i$ stabilized around UAV $j$. Fig. 7.12b shows the IR-based system onboard UAV $i$ tracking the marker of UAV $j$ at all possible bearings, illustrating the desired 360° FOV of the system. Also, note that the marker relative attitude measurements are aligned with the bearing behavior. This makes sense since UAV $i$ is rotating on the spot while tracking the static UAV $j$. Finally, note the period when the measurements were acquired using the damaged receiver section. According to the relative bearing results shown in Fig. 7.9, these sensor malfunction sections produce slightly larger relative bearing errors between the relative bearings of 180° and 270°. By observing Fig. 7.12b, slightly

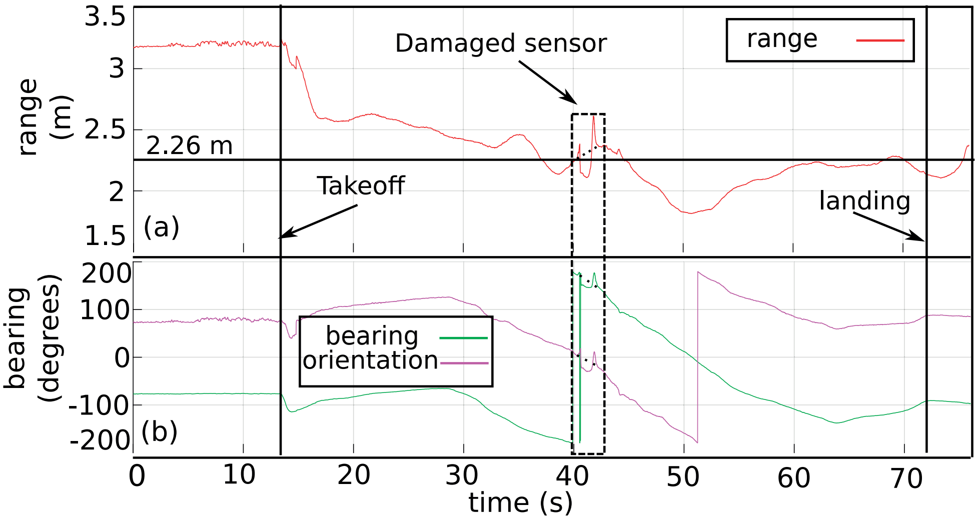


Figure 7.12: Relative pose measurements of the IR-based system during a closed-loop control experiment evolving the entire FOV of the sensor. In this experiment, the UAV with the sensor part of the IR-based had to fly at a desired relative range (2.26 *m*) away from a second (static) UAV equipped with the multi-beacon marker. (a) Marker range, (b) relative bearing, and relative attitude measurements acquired by the IR-based system. Note the takeoff and landing events and the period when the malfunctioning receivers might have been used to acquire the measurements.

larger errors can be detected during the period between 40 *s* and 43 *s*, corresponding to relative bearings close to 180°. In this period, range fluctuations up to 20 *cm* also occur but with no apparent consequences to the closed-loop control algorithm deployed on UAV *i*.

## 7.2 Formation control performance

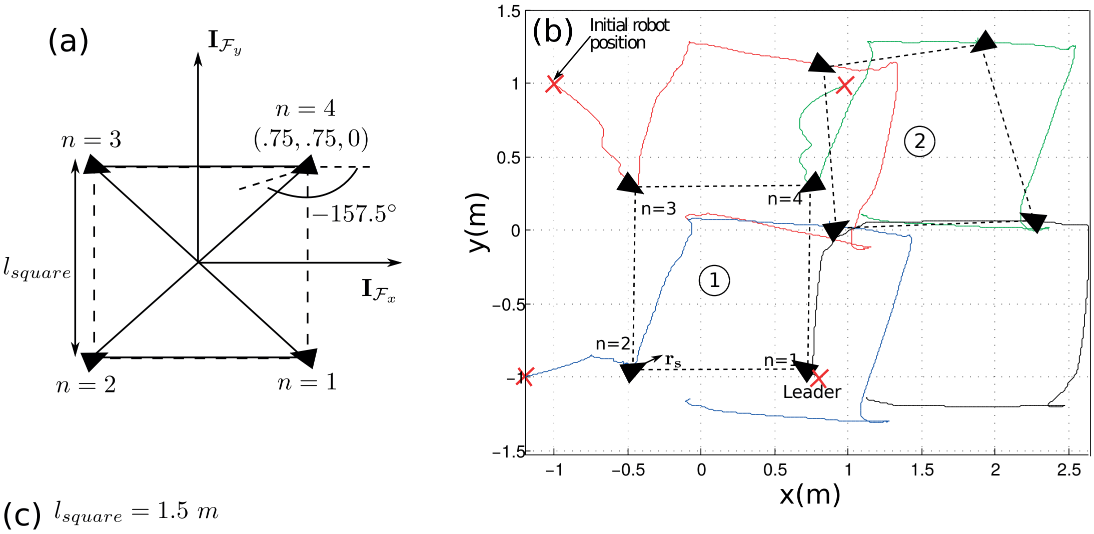
The relative positioning systems developed in this thesis are used to allow a group of UAVs to move in formation using exclusively onboard sensors and control algorithms. This section assesses the performance of the implemented sensing and control stack developed in this thesis, with focus on the formation control algorithm with all the additional terms that tackle the problems described in Section 5.2. The assessment is conducted in simulation, leveraging the high-fidelity simulator described in Section 6.2.2. In this environment, the sensors (the camera-based and the IR-based sensors, as well as the OF and height sensors) and the UAV actuation are simulated, as described in Section 6.2.1. It is worth noticing from Tables 7.1 and 7.2 that the maximum range of the relative positioning systems developed in this work varies from 3 *m* to 4 *m*. Therefore, in these experiments a maximum range of 3 *m* was considered for any used system.

The analysis of the formation control algorithm in simulation starts in Section 7.2.1 by assessing the behavior of the baseline formation control algorithm described in Section 4.2.2 with the FOV constrained camera-based system. Note that at this stage the formation is steered in the environment using a leader-follower approach. With this setup, it is shown that severe convergence issues can occur with the UAV formation, as it is not fully controllable with the desired geometric configuration. This fact shows how the sensing constraints, such as their FOV constraints, can severely limit the possible geometric configurations achievable by the formation. In Section 7.2.2, the same experiments are conducted using the formation control algorithm with the control terms that directly consider the sensing constraints, developed in Section 5.2.1. The results show that the modified algorithm is able to overcome the convergence issues observed in the previous section. However, strong distortions on the formation geometry caused by the movement of the formation using a leader-follower approach are observed. These distortions limit the maximum velocity of the leader, and therefore the reactiveness of the entire formation. A third phase of the analysis conducted in Section 7.2.3 assesses how the addition of the novel formation steering algorithm developed in Section 5.2.2 is able to reduce these distortions, allowing an increase of the maximum velocity of the formation, and therefore its reactiveness. A final phase of the analysis conducted in Section 7.2.4 assesses the robustness of the novel formation control algorithms in the presence of different noise and environment conditions (including obstacles), and its scalability with respect to the increasing number of robots.

### 7.2.1 Baseline leader-follower formation control

The simulation experiments to evaluate the baseline formation control algorithm presented in Section 4.2.2 used four UAVs in a square formation on the $xy$ plane of the world frame, described in Figs. 7.13a and c. The size of the desired square was $l_{square} = 1.5\ m$, and the desired height of the formation was 1 $m$. The desired relative height between all the UAVs was set to zero. The formation was steered in the environment using a leader-follower approach. In these experiments, the leader moved with a certain velocity in the $x$ and $y$ dimensions to random positions, as shown in Fig. 7.13b (the velocity in the $z$ dimension was set to zero). The desired horizontal speed for the leader was of 0.5 $m/s$. All UAVs were running the formation control algorithm defined by Eqs (4.12), (4.13) and (4.11). The desired velocities ($\mathbf{v}_{i_h,d}^{\mathscr{L}_i}$ and $v_{i_z,d}^{\mathscr{L}_i}$) were set to zero for all UAV followers. The desired height for all UAVs ($z_{i,d}^{\mathscr{L}_i}$) was set to the desired height of the formation (1 $m$). The OF and height sensors simulated on the UAVs provide the sensory feedback to control ($z_{i,d}^{\mathscr{L}_i}, \mathbf{v}_{h_i,d}^{\mathscr{L}_i}$) using the previous equations. Note that, for this experiment the $\psi_i$ bias errors of the simulated platform (presented in Section 6.2.1) were set to zero for each UAV.

The relative inter-vehicle localization of neighboring UAVs, required to run the previous algorithm, was measured by each UAV using the simulated camera-based system. With that into consideration, the most complete sensing and control graphs ($\mathscr{G}_S$ and $\mathscr{G}_F$) that could be achieved within the FOV constraints of the simulated camera (which had a 90° horizontal FOV) were chosen, as depicted by the continuous lines connecting each UAV in Fig. 7.13a. Note that the graph edges are bidirectional according to the bidirectional sensing assumptions considered in Section 4.2.2. Additionally, the desired UAV attitudes ($\psi_{fi,d}$), shown in Fig. 7.13c, were computed in order to direct the sensing camera on board each UAV to its optimal direction, as defined in Section 5.2.1. Finally, the initial UAV positions in the experiments were set so that all necessary FOV constraints of the camera-based system were initially met. This consisted of taking off all UAVs to the desired formation height and rotating them to attitudes where they



(c) $l_{square} = 1.5\ m$

| | $n=1$ | $n=2$ | $n=3$ | $n=4$ |
|---|---|---|---|---|
| $x^{\mathcal{F}}_{fi,d}$ | $0.75\ m$ | $-0.75\ m$ | $-0.75\ m$ | $0.75\ m$ |
| $y^{\mathcal{F}}_{fi,d}$ | $-0.75\ m$ | $-0.75\ m$ | $0.75\ m$ | $0.75\ m$ |
| $z^{\mathcal{F}}_{fi,d}$ | $1.0\ m$ | $1.0\ m$ | $1.0\ m$ | $1.0\ m$ |
| $\psi^{\mathcal{F}}_{fi,d}$ | $157.5°$ | $22.5°$ | $-22.5°$ | $-157.5°$ |

Desired formation geometric configuration

| | $n=1$ | $n=2$ | $n=3$ | $n=4$ |
|---|---|---|---|---|
| $n=1$ | $0\ m$ | $1.5\ m$ | $2.26\ m$ | - |
| $n=2$ | $1.5\ m$ | $0\ m$ | - | $2.26\ m$ |
| $n=3$ | $2.26\ m$ | - | $0\ m$ | $1.5\ m$ |
| $n=4$ | - | $2.26\ m$ | $1.5\ m$ | $0\ m$ |

Desired inter-vehicle ranges

| | $n=1$ | $n=2$ | $n=3$ | $n=4$ |
|---|---|---|---|---|
| $n=1$ | $0°$ | $22.5°$ | $-22.5°$ | - |
| $n=2$ | $-22.5°$ | $0°$ | - | $22.5°$ |
| $n=3$ | $-22.5°$ | - | $0°$ | $22.5°$ |
| $n=4$ | - | $22.5°$ | $-22.5°$ | $0°$ |

Desired inter-vehicle bearings

Figure 7.13: Performance of the baseline formation control algorithm implementing a leader-follower approach in simulation. (a) The desired formation configuration defined in the formation frame. (b) Evolution of the UAV trajectories performed in one experimental run. Each UAV is represented by a black triangle, and each color illustrates a trajectory of a specific UAV. The triangle orientation is the same as the sensor direction ($\mathbf{r}_{s_i}$). (1) and (2) represent snapshots during the experiment, when the formation is respectively stable and distorted. (c) Details of the desired configuration for the chosen sensing and control graphs for $l_{square} = 1.5\ m$.

could observe all their neighbors (in this case two) with their onboard sensing camera. The proposed algorithm would only run after all UAVs have achieved their initial positions.

During each run of the simulation, the pose of each UAV was tracked and the relative ranges between pairs of UAVs were computed for analysis. The error between the desired actual values for these quantities was then computed. Fig 7.14 shows the progress of all the horizontal relative range errors between all pairs of UAVs. The results show that with the baseline formation control and the previously defined $\mathcal{G}_S$ and $\mathcal{G}_F$ in this experiment, relative range errors never converge to zero. This is because $\mathcal{G}_F$ is not rigid (see discussion in Section 3.6 about rigidity). Therefore, the set of desired inter-vehicle constraints corresponds to multiple formation geometries. This makes the formation not fully controllable to the desired geometric configuration with the baseline control algorithm. This fact produces an unrecoverable situa-

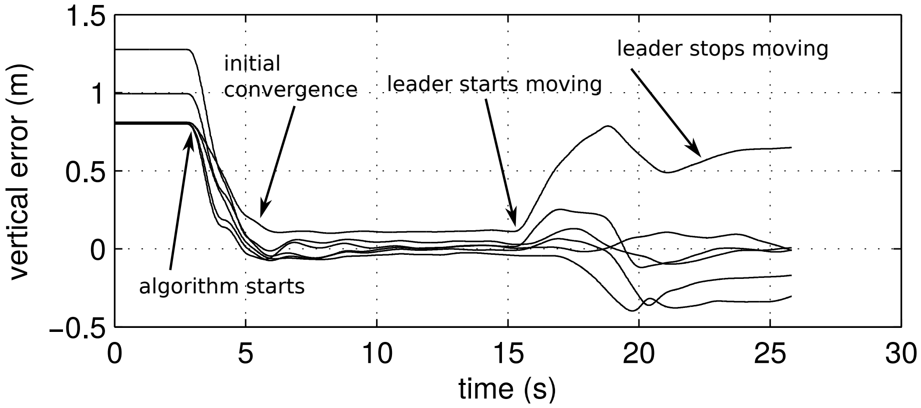


Figure 7.14: Errors between the desired and the actual horizontal range between two UAVs, during one simulation run when using the baseline formation control algorithm implementing a leader-follower approach. Each line on the plot corresponds to the progress of the horizontal relative range error between a pair of UAVs.

tion as soon as the leader starts moving for the first time. When the leader moves, the motion lag between the leader and the followers produce distortions on the geometry of the system. Since the formation can not be fully controlled towards the desired geometric configuration, most of the time the formation is not able to reduce those distortions. After a while, the FOV constraints are no longer compatible with the current relative ranges which causes the UAVs to lose their neighbors. In these conditions, the first low level safety trigger described in Section 6.1.1 is activated, forcing the UAVs to hover in the same position once they lose the leader. Only the leader will keep moving with the desired velocity, until the motion command stops. This example shows how the sensing constraints, such as the FOV constraints, can severely limit the possible geometric configurations achievable by the formation.

### 7.2.2 Including field of view constraints

To remove the controllability issues observed in the results of Section 7.2.1 and enforce the sensing constraints when multiple neighbors are being observed, the baseline formation control algorithm is adapted as described in Section 5.2.1. This adaptation corresponds to adding control terms that directly consider the sensing constraints of each UAV. In this work, the horizontal FOV constraint of each sensor is controlled by modifying the attitude controller defined in Eq. (4.11) to the one defined in Eq. (5.20), and by completing the horizontal control law defined in Eq. (4.13) with the term defined in Eq. (5.21). To test the stability, rigidity, and correct maintenance of the sensor FOV constraint of this modified formation control algorithm, the same experiments of the previous sections were conducted, with the same sensing and control setup for each UAV, and the same uncertainty conditions. The only difference is that the formation control algorithm is now that described above.

During each run of the simulation, the pose of each UAV was tracked and the relative ranges and bearings between pairs of UAVs were computed for analysis. The error between the desired and the

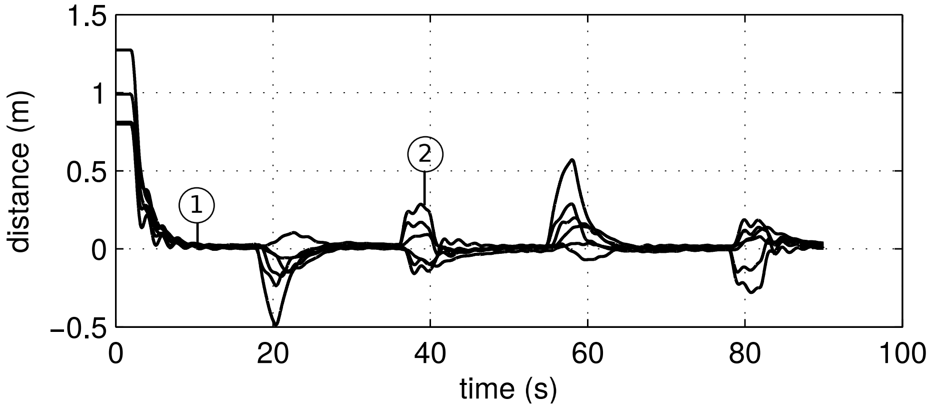


Figure 7.15: Errors between the desired and the actual horizontal range between two UAVs, during one simulation run when using the formation control algorithm with FOV constraint control implementing a leader-follower approach. This simulation run corresponds to the trajectories presented in Fig. 7.13a. The two snapshots described on that figure are highlighted here using circles.

actual values for these quantities was then computed. In this section the inter-edge apertures, as defined in Section 5.2.1, were also computed, in order to assess the controllability of the FOV constraints. Fig 7.15 shows the progress of all the relative horizontal range errors between all pairs of UAVs. Unlike the results in the previous section, the current results show that the modified formation control algorithm using the previously defined $\mathscr{G}_S$ and $\mathscr{G}_F$ in this experiment is able to fully control the formation to the desired geometric configuration, since all relative range errors converge to zero. The control graph $\mathscr{G}_F$ becomes rigid since, with the additional control terms, each UAV also indirectly controls the ranges between the neighbors themselves (see explanation of Fig 5.22). Therefore, the dashed lines in the formation definition in Fig. 7.13a are added to $\mathscr{G}_F$ and the formation becomes fully controllable.

The results show that the formation always converges to the right configuration, even after the perturbation caused by leader movements, showing the stability properties that were discussed in Section 5.2.1.1. Note that, when the leader moves, the geometric configuration is distorted (also observed in Fig. 7.13), and with higher intensity on the $y$ axis (observed by the higher relative range errors). The distortions happen because only the leader has the knowledge of the desired velocity and the followers simply follow the leader. The following behavior lags behind because of the perception-to-action loop delays of the UAVs. The distortion is larger on the $y$ axis, because that is the direction where the leader $n = 1$ is aligned with the follower $n = 4$ (numbers described in Fig. 7.13), that only relies on other followers for information. This will generate a larger motion lag between the leader and that follower, creating a larger relative range and bearing errors. This issues limits the maximum velocity of the leader, as it's speed can not be larger than a value that is able to generate a distortion that breaks the sensing constraints of the formation. Therefore, the reactiveness of the formation becomes limited by the size of these distortions.

Fig. 7.16 shows the relative bearings and inter-edge apertures for the neighborhood observed by the leader $n = 1$ and the follower $n = 2$ (numbers described in Fig. 7.13). The results show similar

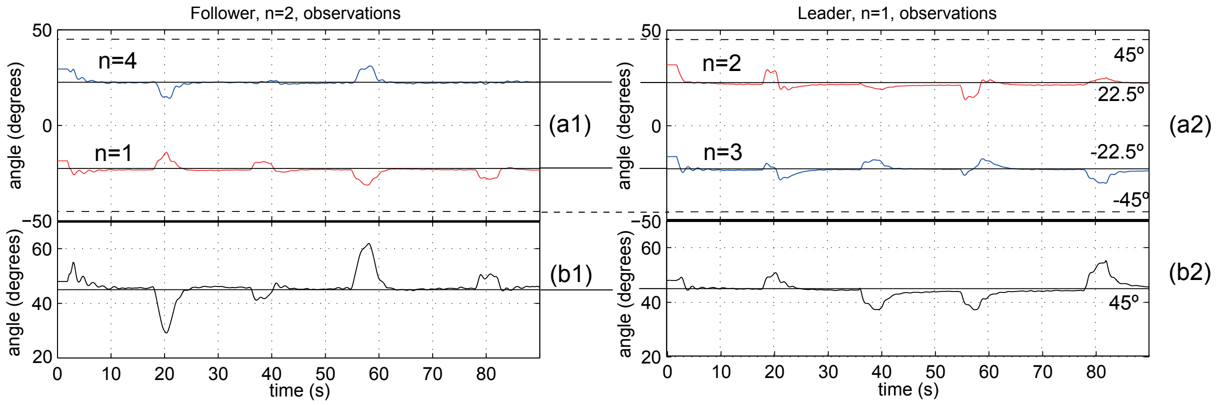


Figure 7.16: Controlled relative bearings (a) and inter-edge aperture (b) values, observed from UAVs $n = 2$, and $n = 1$ (numbers described in Fig. 7.13a). Desired relative bearing and inter-edge apertures depicted using horizontal lines, with corresponding values shown on the right. The neighbor to which each relative bearing line corresponds, is indicated in text on the left. The FOV limits are shown with dashed lines.

perturbations on the relative bearing and inter-edge aperture values. However, Fig. 7.16a, shows the convergence of the relative bearings, in both leader and follower cases, to symmetric values with respect to the sensor direction ($\mathbf{r}_{s_i}$), meaning that the modified attitude controller was able to optimize the sensor FOV. Additionally, Fig. 7.16b confirms, for the leader and follower cases, the convergence of the inter-edge aperture to the desired values. The inter-edge aperture distortions are also higher for movements on the $y$ axis, for the same reasons as before. These results, in combination with results presented in Fig. 7.15, show the correct behavior of the modified horizontal controller, and the correct maintenance of the FOV constraints. This allows each UAV to control multiple neighbors inside a limited FOV sensor, without one of them leaving the sensing area. Moreover, with the addition of edges in $\mathscr{G}_F$ that could not be controlled in the previous section, one can conclude that this modified algorithm allows for a larger number of feasible geometric configurations for the formation.

### 7.2.3 Formation steering with virtual structure

The distortions observed in the results of Section 7.2.2 were caused because the formation was steered in the environment using a leader-follower approach. To remove those distortions, this work switches to a virtual structure approach to steer the formation. The considered algorithm is described in Section 5.2.2; it is able to operate solely using the relative inter-vehicle localization information acquired by each UAV. To test the formation steering algorithm, experiments similar to the ones conducted in previous sections were performed, with the same uncertainty conditions. In these experiments, the same square geometric configuration with four UAVs was used. However, in order to be able to disregard the FOV constraints discussed in the previous section, the relative positioning system on board each UAV is switched to the IR-based system with a 360° FOV. With that into consideration, fully connected graphs can be considered for $\mathscr{G}_S$ and $\mathscr{G}_F$. Therefore, the controllability problems observed in Section 7.2.1 due to the usage of the baseline formation control are no longer an issue in this case.

These experiments use the same baseline controller considered in Section 7.2.1. Note that the algorithm with FOV constraint control considered in Section 7.2.2 could also be used in case of the camera-based positioning system is considered. The steering algorithm translates the desired virtual structure motion to the desired UAV motion by means of the desired horizontal velocity ($\mathbf{v}^{\mathscr{L}_i}_{i_h,d}$) and desired height ($z^{\mathscr{L}_i}_{i,d}$) of each UAV. These quantities are defined in Eqs (4.12), (4.13) of the baseline control algorithm implemented on each UAV. The OF and height sensors simulated on the UAVs provide the sensory feedback to control ($z^{\mathscr{L}_i}_{i,d}, \mathbf{v}^{\mathscr{L}_i}_{h_i,d}$) using the previous equations. The virtual structure is defined by the position of the formation center and the attitude of the formation frame (see Section 4.2.2 for details about the definition of this frame). The desired motion of the virtual structure is defined in terms of the translation of its 3D position (i.e. the position of the formation frame center), or its angular velocity (i.e. the angular velocity of the vehicle rotation around the formation center). These motion directives are implemented by means of coordinated formation motion commands, defined in Section 5.2.2 ($\mathbf{u}_{vf_h}$, $u_{zf}$, $u_{wf_z}$). These motion commands are translated to the desired motion of each UAV ($\mathbf{v}^{\mathscr{L}_i}_{i_h,d}$, $z^{\mathscr{L}_i}_{i,d}$) using Eq. (5.25). The previous algorithm requires the knowledge of the virtual structure by each UAV. The novel consensus algorithm, described in Section 5.2.2.1, allows each UAV to locally acquire this information, relying only on the relative inter-vehicle localization of its neighbors and the desired formation geometric constraints.

The experiments conducted in the previous sections controlled the leader UAV by giving it desired velocities expressed in its flying frame. These experiments did not move the formation to any particular place in the environment. However, an autonomous teleoperation system, external to the formation, could have been implemented in order to move the leader to specific positions in the environment with velocity commands. In this section, such system is implemented for controlling the pose of the formation frame. The autonomous system measures, at each time step of the simulation, the average of all UAV positions. Those measurements are used in the estimation algorithm proposed in Section 5.2.2.2 in order to produce an estimate of the formation frame pose in the environment ($\hat{\mathbf{x}}^{\mathscr{W}}_{\mathscr{F}}, \hat{\mathbf{R}}^{\mathscr{W}}_{\mathscr{F}}$). Note that $\mathbf{R}^{\mathscr{W}}_{\mathscr{F}}$ define the three axes of the formation frame expressed in the environment frame ($\mathbf{I}^{\mathscr{W}}_{\mathscr{F}_x}, \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_y}, \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_z}$). The error between these estimates and the desired pose in the environment ($\mathbf{x}^{\mathscr{W}}_{\mathscr{F},d}, \mathbf{R}^{\mathscr{W}}_{\mathscr{F},d}$) can then be used in a control algorithm that generates the desired motion commands for the virtual structure. In this work, the Algorithm 2 is used, which switches between translation control or attitude control of the formation, depending on whether the attitude error is sufficiently large (controlled by $th1$). The output of this algorithm are the coordinated formation motion commands ($\mathbf{u}_{vf_h}$, $u_{zf}$, $u_{wf_z}$), which are then broadcast to all UAVs simultaneously and then translated to local UAV motion commands, as previously described. Note that in Algorithm 2, *formationPoseEstimation* corresponds to the formation frame pose estimation algorithm of Section 5.2.2.2 , $angle(\mathbf{x}, \mathbf{y})$ is a function that returns the angle between vectors $\mathbf{x}$ and $\mathbf{y}$, and *BroadcastFormationMotionCommand* relates to the broadcasting of the coordinated formation motion commands to all UAVs simultaneously. In simulation, the communication network that broadcasts the motion commands to the UAVs is simulated as an idealized network.

In the experiments, the formation is moved to a set of desired poses in the environment, as shown in

Fig. 7.17. The desired height of the formation was again set to 1 $m$. As before, each experiment begins by moving the UAVs to initial random positions in the environment. The baseline formation control algorithm would then be activated so that the UAVs converge to the desired configuration. After a period of time, the autonomous external system that steers the formation is activated, moving the formation to the desired poses in the environment ($g_0$ to $g_4$ in Fig. 7.17) by sending coordinated formation motion commands to the UAVs, according to Algorithm 2. The amplitude of the motion commands was limited to 1.0 $m/s$ for the horizontal velocity, and to 0.3 $rad/s$ for the angular velocity.

**Algorithm 2** Formation steering control law with virtual structure

```
procedure FormationSteering
    $(\hat{\mathbf{x}}^{\mathscr{W}}_{\mathscr{F}}, \hat{\mathbf{R}}^{\mathscr{W}}_{\mathscr{F}}) \leftarrow formationPoseEstimation$;
    if $angle(\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}, \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_x,d}) > th1$ then
        $\mathbf{u}_{vf_h} \leftarrow (0,0)$;
        $u_{zf} \leftarrow z^{\mathscr{W}}_{\mathscr{F},d}$;
        $u_{wf_z} \leftarrow -k_w\, angle(\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}, \mathbf{I}^{\mathscr{W}}_{\mathscr{F}_x,d})$;
    else
        $\mathbf{u}_{vf_h} \leftarrow k_v \left[ (x^{\mathscr{W}}_{\mathscr{F},d}, y^{\mathscr{W}}_{\mathscr{F},d}) - (\hat{x}^{\mathscr{W}}_{\mathscr{F},d}, \hat{y}^{\mathscr{W}}_{\mathscr{F},d}) \right]$;
        $u_{zf} \leftarrow z^{\mathscr{W}}_{\mathscr{F},d}$;
        $u_{wf_z} \leftarrow 0$;
    $BroadcastFormationMotionCommand(\mathbf{u}_{vf_h}, u_{zf}, u_{wf_z})$;
```

During each run of the simulation, the pose of each UAV was tracked and the relative ranges were computed for analysis. The error between the desired and the actual values for the previous quantity was then computed. Fig. 7.18a shows the progress of all the relative horizontal range errors between all pairs of UAVs. The results show that the distortions occurred during formation motion are much smaller than in the ones achieved in the previous sections using a leader-follower approach to steer the formation. These results are achieved even though the maximum formation speeds (1 $m/s$) are twice as large the ones used in the previous sections. Small distortions are still visible, caused by sensing and actuation inaccuracies modeled in simulation, but easily kept under control by the formation control algorithm. This shows that the formation steering algorithm does not compromise the stability properties of the formation control algorithm discussed in Section 5.2.1.1. Moreover, the results confirms a larger reactiveness of the formation using a virtual structure approach, which can now be implemented solely using relative inter-vehicle localization measurements with the novel algorithm developed in this work (without extra inter-vehicle communication).

During each run of the simulation, the positions of all UAVs in simulation are also averaged in order to acquire information about the formation center. The information about the formation frame attitude ($\mathbf{I}^{\mathscr{W}}_{\mathscr{F}_x}$) can also be acquired by computing $\psi_f = \psi_i - \psi_{fi,d}$ for each UAV, and then average this value for all UAVs. This information is then compared with the results acquired by the formation frame pose estimation algorithm of Section 5.2.2.2, as shown in Figs. 7.18b and c. The results show that the formation frame pose estimates closely follow the ground truth, validating the proposed estimator. Note that this

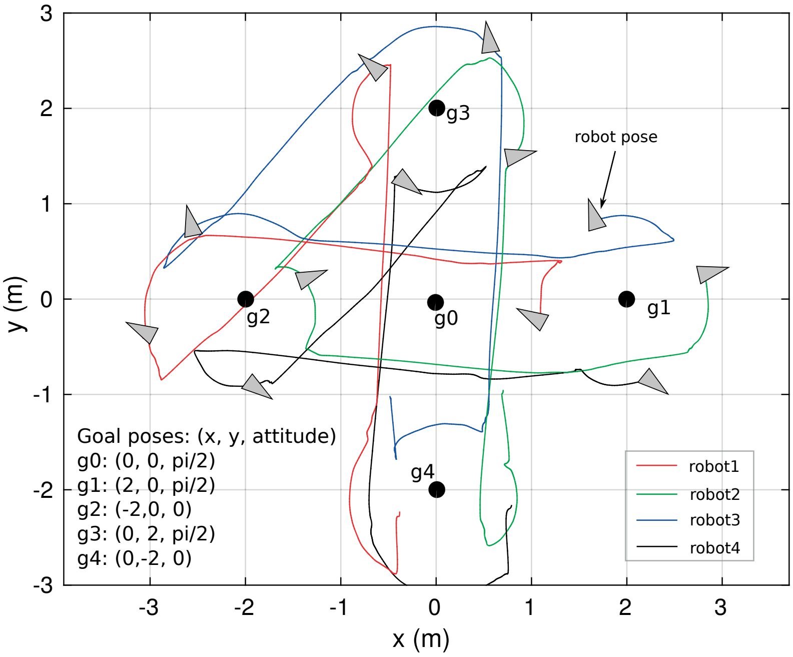


Figure 7.17: Evolution of the UAV trajectories performed in one experiment run. Each UAV is represented by a black triangle, and each color illustrates a trajectory of a specific UAV. The triangle orientation is the same as the sensor direction ($\mathbf{r}_{s_i}$). (1), (2) and (3) represent snapshots during the experiment. The external system has to guide the formation through a sequence of goals $g_0$ to $g_4$ consisting of a formation position in the $xy$ plane, and a formation attitude.

estimator did not require the individual UAV positions, and only single coordination motion commands were sent to the entire formation. Additionally, if the communication network that sends the motion commands fails, the worst case scenario is the formation stopping in place. Therefore, the external teleoperation system is relieved from controlling and estimating individual UAVs, avoiding unnecessary inconsistencies on the UAV positions or communication problems that could result in catastrophic failures. This allows the external system to be scalable in terms of the number of UAVs.

### 7.2.4 Robustness, scalability, and cluttered environments

The experiments conducted in Sections 7.2.1 to 7.2.3 show how the improved formation control algorithm has a better performance than the baseline formation controllers, for a specific simulation run, with a specific uncertainty, and for a team of four UAVs moving along an empty arena. This section presents experiments that evaluate the robustness of the algorithm to different levels of noise and clutter in the environment, and how it scales with the number of UAVs.

These experiments were divided into two scenarios. In *Scenario 1*, depicted in Fig. 7.19b, the formation had to perform a set of translation and rotation maneuvers at a certain speed in an empty arena. In this scenario, formations of two to six UAVs were considered. In *Scenario 2*, depicted in Fig. 7.19c, the formation had to go through an obstacle course while overcoming several challenges (bottlenecks,

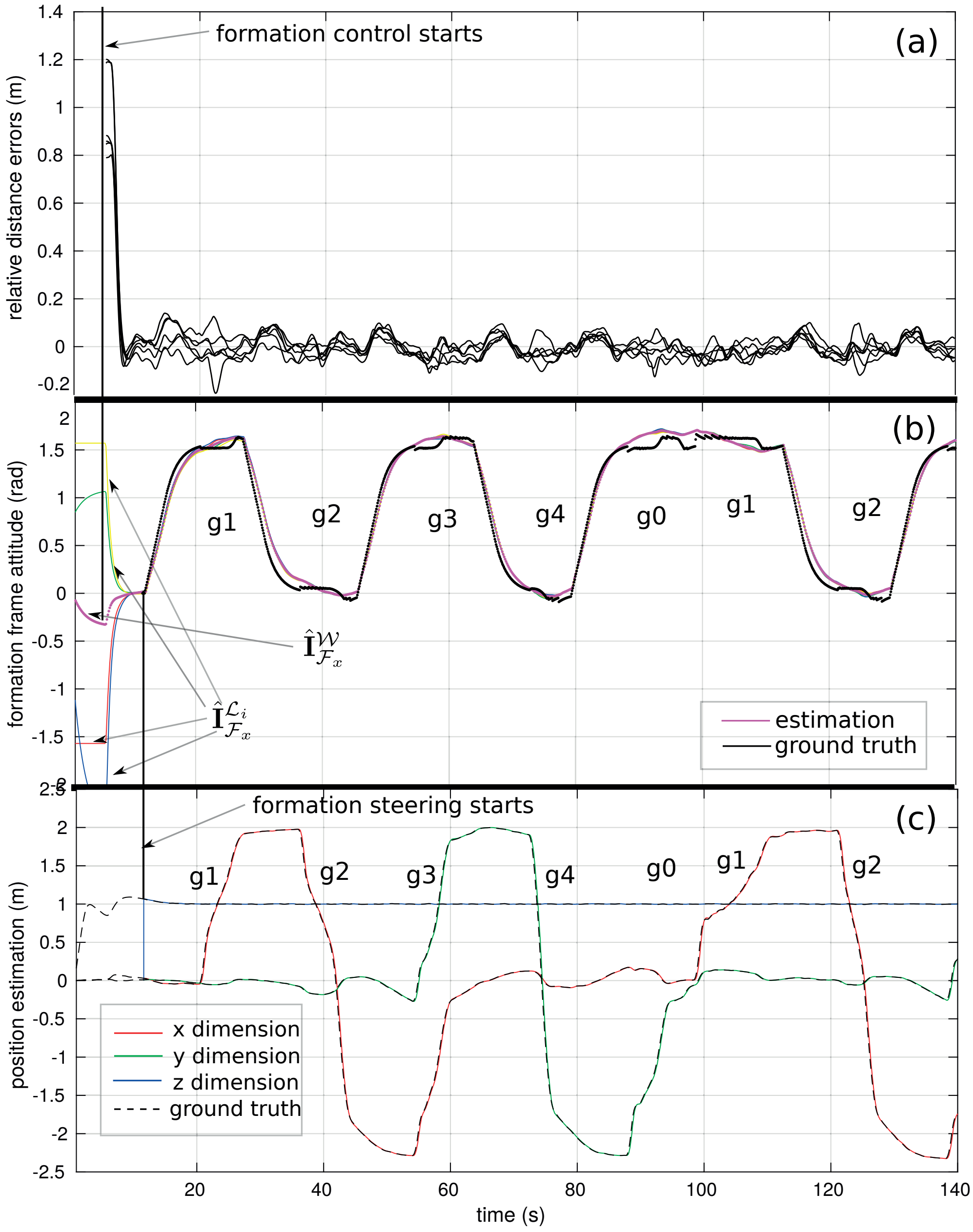


Figure 7.18: Performance of the formation control algorithm with a virtual structure approach. (a) Errors between the desired and the actual range between two UAVs, during one simulation run. This simulation run corresponds to the trajectories presented in Fig. 7.17. (b) Estimates of the attitude of the formation frame (defined by the $x$ axis of the formation frame, $\mathbf{I}_{\mathscr{F}_x}$), acquired by the external system $\hat{\mathbf{I}}^{\mathscr{W}}_{\mathscr{F}_x}$, and by each UAV $\hat{\mathbf{I}}^{\mathscr{L}_i}_{\mathscr{F}_x}$. (c) Estimates of the position of the formation center acquired by the external system.

curves, and cluttering objects). In this scenario, a formation of six UAVs was considered. For both scenarios, each UAV was equipped with an IR-based system with a 360° FOV, as in Section 7.2.3.

The geometric configurations used for the UAV formation in these scenarios are shown in Fig. 7.19a.

Similarly to the previous sections, the desired formation geometry lies on the $xy$ plane of the world frame with a desired height of 1 $m$. The desired distances between two pairs of UAVs was set to 1.7 $m$. One immediate problem that arises from increasing the number of UAVs in the formation is the maximum range of the relative positioning system, which was set to 3 $m$ for all simulation experiments done in this work (see beginning of Section 7.2). In these conditions, $\mathscr{G}_F$ and $\mathscr{G}_S$ become only partially connected when more than three UAVs are used. In Fig. 7.19a, a dash line connecting UAVs $i$ and $j$ means that $j \in \mathscr{N}_i$ and vice-versa (i.e. bidirectional edge added to $\mathscr{G}_S$). For these configurations, $\mathscr{G}_F$ was made the same as $\mathscr{G}_S$. Note that some edges that do not belong in $\mathscr{G}_S$ will belong to $\mathscr{G}_F$ if the UAVs are actively enforcing the FOV constraints of their onboard relative positioning systems, using the control algorithm presented in Section 5.2.1. However, those edges are not shown in Fig. 7.19a for simplicity. Note that, for all configurations in Fig. 7.19a, the graph $\mathscr{G}_F$ is rigid, meaning that if the UAVs control the inter-vehicle constraints defined in the considered $\mathscr{G}_F$ graphs, the formation will converge to a single geometry (see discussion in Section 3.6 about rigidity).

As illustrated in Fig. 7.19a, for each geometric configuration, the formation heading is defined as the $x$-axis of the formation frame ($\mathbf{I}_{\mathscr{F}_x}$), and the formation front, $L1$, is defined as the first UAV that is encountered when going in the direction opposite from the formation heading. The number of hops between UAVs $i$ and $j$ corresponds to the minimum number of edges used to connect $i$ and $j$ on a given graph. The number of sensing and control hops relate respectively to the $\mathscr{G}_S$ and $\mathscr{G}_F$ graphs. The group L(k) in the formation is defined as the group of UAVs that have $k-1$ sensing hops to the formation front.

In *Scenario 1*, each experimental run consisted of performing multiple translation and rotation maneuvers with the formation. In order to gather statistical data, several experimental runs (about ten) were conducted for several combinations of uncertainty profiles and number of UAVs. The uncertainty profile is defined by a certain $\psi_b$ bias noise and the other sensing and actuation noise already defined in Section 6.2.1. Different uncertainty profiles were achieved by varying the $\psi_b$ bias from $0^\circ$ to $\pm 15^\circ$. The formation configuration used for a specific number of UAVs was taken from the configurations depicted in Fig. 7.19a. In *Scenario 2*, each experimental run consisted of a round tour of the formation through the environment. Multiple experimental runs (about ten) were conducted using one uncertainty profile, with a $\psi_b$ bias of $\pm 10^\circ$,and with the formation of six UAVs depicted in Fig. 7.19a, in order to assess how the algorithm behaves in the presence of obstacles.

The experiments on both scenarios were conducted independently for two versions of the control algorithm: a version using a leader-follower approach, as in Section 7.2.2, and a version using a virtual structure approach, as in Section 7.2.3. Both versions included the additional control terms (described in Section 5.2.1) responsible for enforcing the FOV constraints of the relative positioning systems on board each UAV. For comparative purposes, the parameters of both versions of the controller (mainly $k_p$, $k_v$, $k_\psi$, $k_\alpha$, $k_{p_1}$, $k_{v_1}$, and $\mathbf{L}$) were initially set to the same values. In particular, $\mathbf{L}_{ij} = \frac{1}{N_i}, \forall j \in \mathscr{N}_i, \forall i$. For the leader-follower approach, the UAV corresponding to the formation front ($L1$ depicted in Fig. 7.19a) was chosen as the formation leader. A teleoperator was responsible to move the formation by sending commands (either to the leader UAV or to all the UAVs of the formation) through the offboard processing unit, as

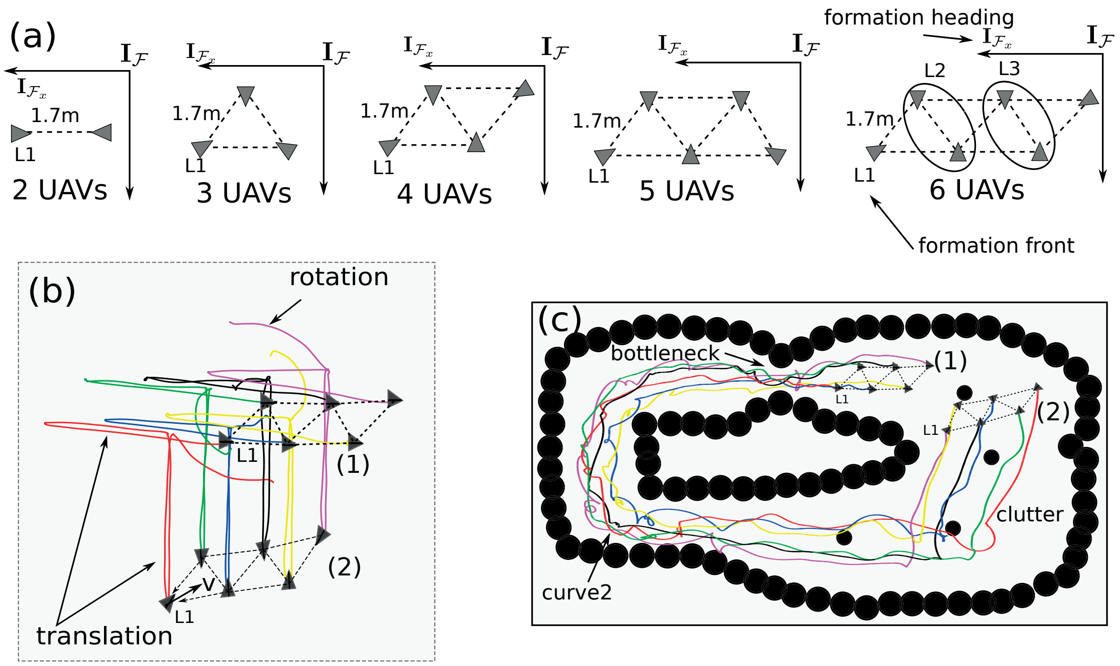


Figure 7.19: Simulation scenarios used to study the robustness and scalability of the formation control algorithms. (a) The different formations geometries used, depending on the number of UAVs. Each triangle corresponds to a UAV, and the triangle's direction corresponds to the UAV's desired horizontal heading in the formation. Note the definition of the formation front (L1) and formation heading ($\mathbf{I}_{\mathscr{F}_x}$). (b) Scenario 1: empty arena. (c) Scenario 2: cluttered environment. For each scenario, note the two snapshots taken at a specific time instance of an experimental run. For each snapshot, the formation front is indicated in order to give an idea of the formation heading. Each color illustrates a trajectory of a specific UAV. The $z$ axis is not displayed for illustration simplicity.

described in Section 6.1.1. Recall that, in simulation, the communication links between the offboard processing unit and the onboard processing unit of each UAV are modeled by a perfect network, as described in Section 6.2.1. The commands were sent with an intensity so that the UAVs would move at $1\ m/s$. This meant sending linear speed commands ($\mathbf{v}^{\mathscr{L}_i}_{i_h,d}$ for the leader-follower approach, and $\mathbf{u}_{vf_h}$ for the virtual structure approach) of $1.0\ m/s$, and an angular speed ($\mathbf{u}_{wf_z}$ for the virtual structure approach) of $0.6 rad/s$.

To assess the performance of the control algorithm for each experimental run, the notion of formation integrity was used. The formation integrity measures the matching between the desired and current geometric configuration of the formation, and it combines two metrics: the maximum error between the desired and current relative range between two neighboring robots (i.e. with $\mathbf{L}_{ij} \neq 0$), denoted as $M1$; the percentage of runs where the robots break formation and do not recover, denoted as $M2$. Breaking the formation means that the UAVs leave the area of detection of the relative positioning systems of their neighbors, precluding the formation controllers from operating. Note that metric $M1$ can be computed at each time instance of the experimental run (the maximum range error between all pairs of neighbors), or in the end of the experimental run (the maximum range error between all pairs of neighbors at any time instance).

When running the experiments, it became clear that the leader-follower approach was not achieving the desired UAV speeds mentioned above (1 $m/s$), and it had a much smaller formation integrity. This happens because only the leader knows the desired motion, in contrast to the virtual structure approach. This disadvantage grows as the number of sensing hops between the leader and the other UAVs grows. Therefore, the leader-follower approach was made more competitive by changing the version of the formation control algorithm dedicated to it. In the new version, a hierarchy of multiple leaders was considered (similarly to [24]), as depicted in Fig. 7.19a, in order to better cope with multiple hops between UAVs. All $\mathbf{L}_{ij}$ representing connections between a UAV $i$ from L(k) to a UAV $j$ from L(k-1) were changed to one, and the weight $\mathbf{L}_{1j}$ was changed to zero, so that the leader ($i = 1$) could move faster.

The simulation results of the virtual structure approach and the new version of the leader-follower approach are shown in Fig. 7.20. In Scenario 1 (Figs. 7.20a and b), one can clearly observe that the virtual structure approach always achieves a higher formation integrity, regardless of the number of UAVs in the formation or the uncertainty profile. In this scenario, it was also observed that the leader could not move towards the formation (direction $v$ in Fig. 7.19b) without breaking it. Therefore, the leader was always forced to move away from the formation. For the virtual structure approach, motion in all directions was achieved. Finally, the rotation behavior could only be achieved in the virtual structure approach, since moving the leader to rotate the formation would almost always result in motion towards the formation, eventually breaking it.

The superior performance of the virtual structure approach was also clearly visible in Scenario 2 (Figs. 7.20c). In particular, when the formation is passing through *curve2* and the *clutter*, the leader-follower approach performs worst. In *curve2* the teleoperator is able to rotate the formation in the virtual structure approach, leading to a much higher formation integrity while maneuvering through the curve. In *clutter*, the UAVs occasionally lose connections due to occlusions. In the leader-follower approach, when the UAVs lose the leader they also lose the desired direction. In the virtual structure approach, only one sensing connection to any robot is needed to establish a desired direction (see Eq. (5.26) and its explanation), and, therefore, occasional occlusions are less important. During the *bottleneck* periods both approaches performed worst since the formation had to be distorted in order to pass through the small entrance. Note that the metric $M2$ measured in Scenario 2 was of 25% and 41.5% when using a virtual structure and a leader-follower approach, respectively. This means that the virtual structure approach had almost two times less failures than the leader-follower approach.

## 7.3 Real system deployment

This section analyses the performance of the developed UAV sensing and control stack presented in Section 6.1 during a set of real UAV formation control tasks subjected to sensing and actuation inaccuracies. The differences between the observed performance and the performance obtained in simulation are also

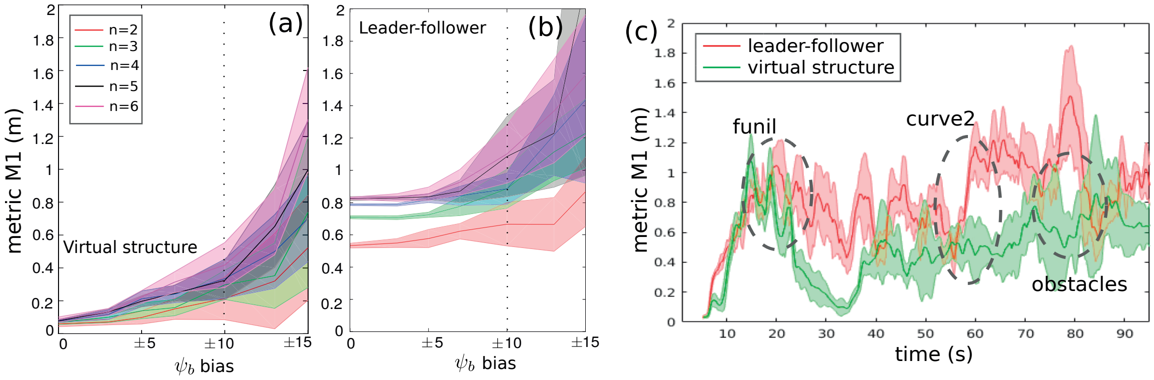


Figure 7.20: Results on the assessment of the robustness and scalability of the formation control algorithms. The plots show the mean and standard deviation for metric $M1$ of the formation integrity (the lower the better) achieved by both versions of the control algorithm, for different noise conditions, in Scenarios 1 (a-b) and 2 (c).

analyzed. The developed relative camera-based and IR-based positioning systems (developed in Section 5.1 and tested in Section 7.1), along with the control stack, are deployed on the real UAVs. The control stack includes the formation control algorithms (developed in Sections 4.2.2 and 5.2, and tested in simulation in Section 7.2) and the localization algorithms that accompany the used relative positioning systems (also developed in Section 5.1). The UAV platforms and the real environments used in this work are described in Section 6.2. The Hummingbird platform was used with the Maillefer flying arena. The UX-401 platform was used with the LBL calibration arena and the Pavilion flying arena. The LBL calibration arena was only used to calibrate the relative positioning systems. The Pavilion flying arena had a larger volume, where it was possible to fly the UAVs, but it had no embedded MCS. An ad-hoc MCS was mounted instead.

The analysis of the of the developed UAV sensing and control stack starts in Section 7.3.1 by assessing the behavior of the baseline formation control algorithm described in Section 4.2.2 with the FOV constrained camera-based system. Note that at this stage the formation is steered in the environment using a leader-follower approach. Two UAVs are used, one being a leader and another a follower. With this setup, it shown that the follower UAV is able to follow the leader using just the relative positioning system. However, the motion lag between the leader and the follower, as observed in simulation in Section 7.2.2, can still be observed. Finally, the results also show that, with no additional sensors to stabilize its velocity in the environment (such as an OF sensor), the UAV is strongly affected by inaccurcies from actuation and from the relative positioning systems. In Section 7.3.2, experiments are conducted with three UAVs, one leader and two followers, with the FOV constrained camera-based system, in order to test cases with multiple neighbors inside the limited FOV of the camera sensor. These experiments are conducted using the formation control algorithm with the control terms that directly consider the sensing constraints, developed in Section 5.2.1. The results show that the modified algorithm allows the UAV to optimize the FOV of its camera when observing two neighboring UAVs, maintaining the desired inter-edge aperture between the neighbors. Note that these results are obtained in the presence of sensing and actuation inaccuracies of real system. The previous observed effects caused by sensing and actuation inaccuracies are still observed in these experiments. A third phase of the analysis conducted in Section 7.3.3 assesses how the addition of the novel formation steering algorithm developed in Section 5.2.2 is able to reduce distortions seen in the leader-follower approach, potentially allowing an increase of the formation reactiveness. Additionally, it also assess how the insertion of the OF sensor is able to substantially reduce the previously observed effects caused by sensing and actuation inaccuracies. In these experiments, the previous FOV constraints are removed by using the IR-based system instead of the camera-based system.

### 7.3.1 Leader-follower formation control with the camera-based system

The sensing and control stack was initially tested with a set of experiments involving two UAVs maintaining a relative range and height between each other, while the formation was moved using the baseline formation control algorithm with a leader-follower approach (see preliminary results in simulation of similar experiments in Sections 7.2.1 and 7.2.2). These experiments used the Hummingbird platform with the Maillefer flying arena. The camera-based system developed in this work was chosen as the relative positioning system on board the UAVs for this experiment. The leader only contained the active beacon marker and the follower only contained the camera sensor to localize the marker. Therefore, only the follower UAV run the formation control algorithm defined by Eqs (4.12), (4.13) and (4.11) in order to maintain the range to the leader. The leader was teleoperated to the desired positions using the safety controller described in Section 6.1, with the help of the MCS measurement feedback provided through the available wireless communication network. Also, with the help of the MCS measurement feedback, the leader run the attitude control defined in (4.11) to keep the active marker turned towards the follower. No other sensors (either height or OF) were used besides the relative positioning system. The relative positioning system was run at 40 $Hz$, which corresponds to the value set in Section 7.1.1.2. The formation control algorithm was run at the same frequency.

In each experiment, the follower had the objective of following the leader at a range of 1.5 $m$ and a relative height of 0 $m$. Also, similarly to what was done in Section 7.2.1, the desired relative attitudes between the UAVs were set in order to direct the sensing camera of the follower and the active marker of the leader towards each other, optimizing the FOV of the sensing camera, in the sense of the optimal sensor direction defined in Section 5.2.1. The leader UAV had the objective of completing a square trajectory, as depicted in Fig. 7.21, through teleoperation. Each experiment started by taking off and move the two UAVs to their initial positions, using the safety controller and the MCS feedback. Note that the initial positions had to ensure that the marker could be observed by the camera sensor, similarly to what was described in Section 7.2.1. After the UAVs have achieved their initial positions, the actual experiment would start by cutting all the connections between the follower and the MCS while activating the formation control for the leader (only in attitude) and the follower. At certain periods of time, the leader would be teleoperated towards a new position. The maximum leader speed recorded during the

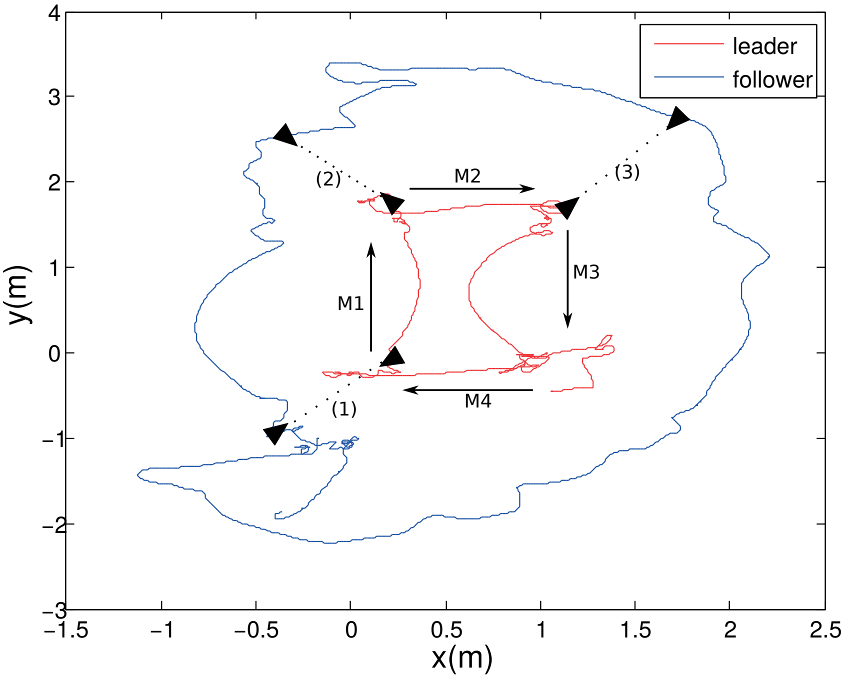


Figure 7.21: Evolution of both leader and follower trajectories during one particular experiment run of the formation control algorithm using a leader-follower approach with two UAVs, observed from top view. Each UAV is represented by a black triangle, and each color illustrates a trajectory of a specific UAV. The triangle orientation is the same as the sensor front ($\mathbf{r}_{s_i}$). (1), (2) and (3) represent snapshots during the experiment. Note the four maneuvers performed by the leader in order to achieve the desired square trajectory. Also note the behavior of the follower, following the leader from the outside the desired square.

experiments using the MCS was of 2 $m/s$.

During each experiment run, the pose of each UAV was tracked and the relative ranges between pairs of UAVs were computed for analysis. The error between the desired and the actual relative range between pairs of UAVs throughout the experiment run was then computed. For this experiment, the attitude ($\psi_i$) of each UAV was also tracked, in order to assess the behavior of the attitude control part of the formation control algorithm. Fig. 7.22 shows the progress of the horizontal and vertical relative range errors between the two UAVs. The results show that the follower is able to maintain the desired range of 1.5 $m$ to the leader only using the camera-based system as sensory feedback. However, the results shows that the follower has to catch up with the leader every time the leader moves (see the horizontal error growing when the leader moves, in Fig. 7.22a). In fact, the leader could only move for a small amount of time before it would have to stop and wait for the follower to catch up. This behavior is related to the motion lag between the leader and the follower previously seen in the simulation results of Section 7.2.2. The vertical relative range was kept always close to the desired values, as shown in Fig. 7.22b. This makes sense, since no vertical motion was performed to the formation.

By analyzing the attitudes of each UAV and the trajectories made by them, as for example shown in

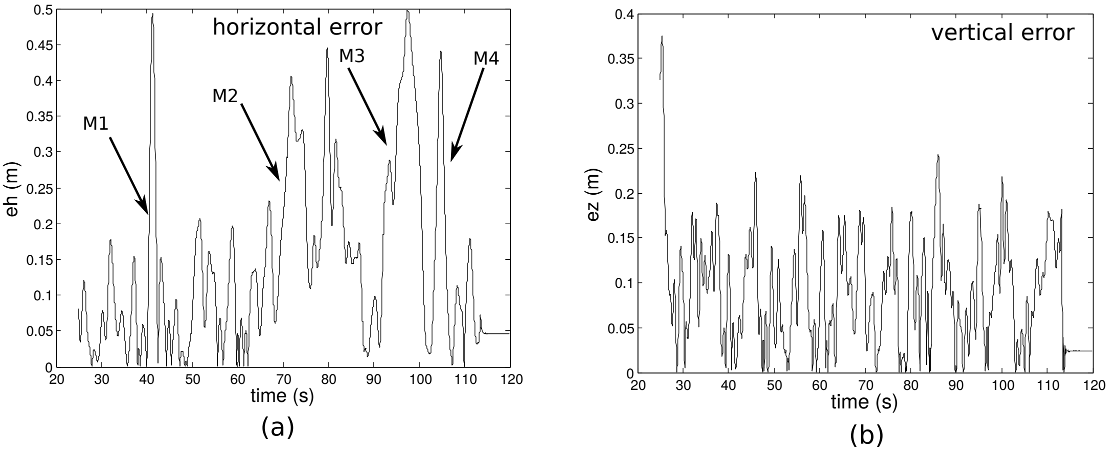


Figure 7.22: Error of the relative horizontal (a) and vertical (b) range between two UAVs with respect to the desired values, during one run of the formation control algorithm using a leader-follower approach with two UAVs. Note the periods of time where the leader performed its maneuvers. These maneuvers are visually depicted in Fig. 7.21.

Figs. 7.23 and 7.21 respectively, it is possible to observe that the follower is rotating around the leader, following it from outside of the square (as seen in Fig. 7.21). This is attributed to the fact that there is a slightly calibration bias of the relative positioning sensor, that triggers this rotation. Given that there is no other sensor to help the UAV to stop in place (such as an OF sensor to allow velocity control), this rotation continues throughout the experiment. Nevertheless, the results displayed in Fig. 7.23 show that the attitude control part of the formation control algorithm works correctly, as the relative attitude between the two UAVs is around the desired value (180°). This means that they are turned towards each other, optimizing the FOV of the sensing camera on board the follower UAV.

Finally, to assess the overall performance of the formation control algorithm, the previous experiment was repeated several times and a statistical analysis of the relative range error with respect to the desired values was conducted. The results of this statistical analysis are shown in Fig. 7.24. These results show that the relative horizontal and vertical range error closely follows a normal distribution. The distribution of the horizontal error is averaged around 10 *cm* and it has a standard deviation of 16.2 *cm*, which correspond respectively to 6.67% and 10.8% of the desired range of 1.5 *m*. The average is non-zero due to the previous discussed motion lag between the leader and the follower. In fact, this value is not larger since the leader was forced to stop in order to let the follower catch up. This motion lag is caused by the leader-follower approach chosen to move the formation in these experiments. These errors will cause distortions on the formation geometric configuration, as already observed in simulation results of Section 7.2.2. The reactiveness of the formation becomes limited by the size of these distortions. The distribution of the vertical error is measured in terms of the relative elevation between the UAVs. This distribution is averaged around 0° and with a standard deviation of 4.7°. The average of this distribution would have been non-zero if there would have been vertical motion of the leader in the conducted

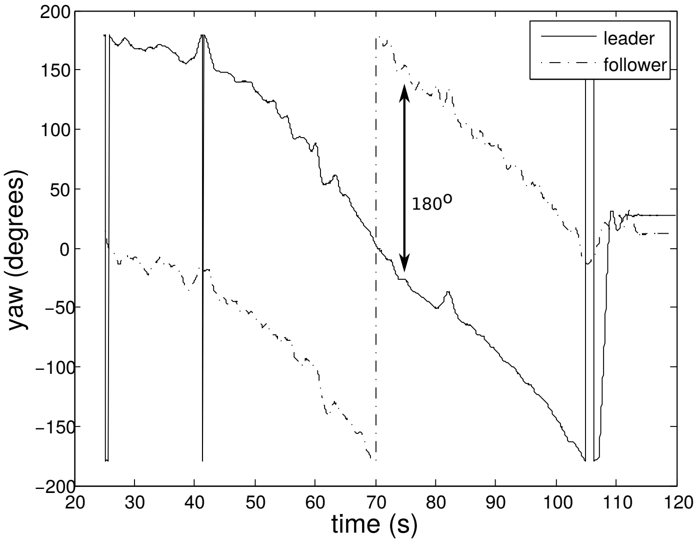


Figure 7.23: Evolution of the UAV attitudes during one run of the formation control algorithm using a leader-follower approach with two UAVs.

experiments.

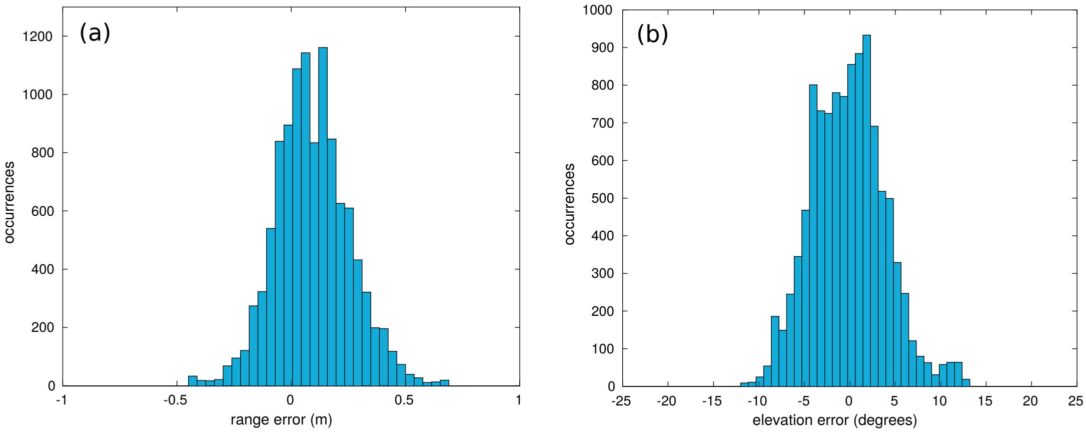


Figure 7.24: Statistics for the relative horizontal (a) and vertical (b) range errors between two UAVs with respect to the desired values, during several runs of the formation control algorithm using a leader-follower approach with two UAVs. Note that the vertical error is represented by the relative elevation between the two UAVs.

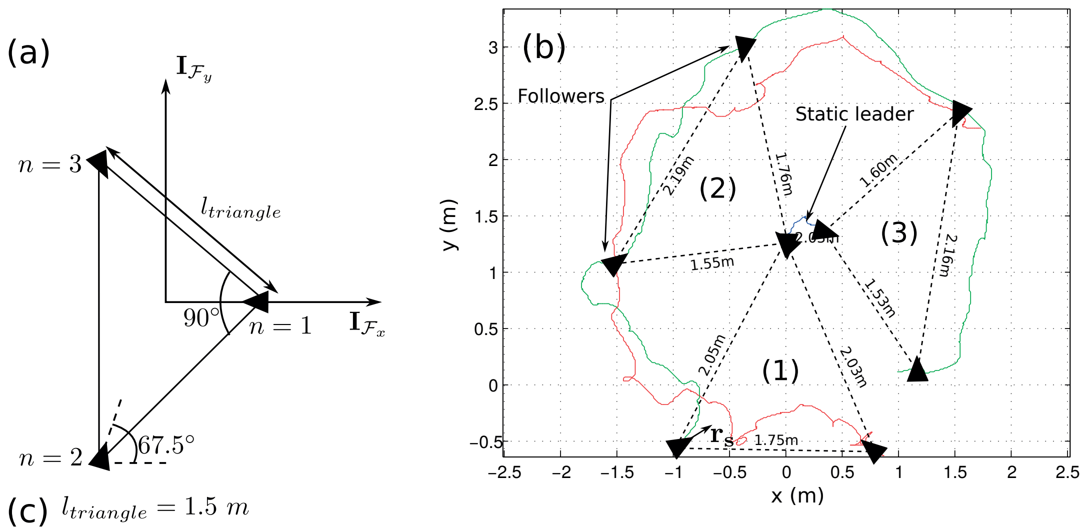
### 7.3.2 Managing multiple neighbors with the camera-based system

The sensing and control stack was then tested in a situation where each UAV had to manage multiple neighbors inside the FOV of its relative positioning sensor. This was done by performing a set of experiments similar to the previous section, with the same UAV platform and testing environments. In these experiments three UAVs (one leader, and two followers) were used. The camera-based system developed in this work (now fully deployed on the three UAVs) was chosen as the relative positioning system on board the UAVs for this experiment. The two followers run the formation control algorithm as in the previous section, but with the modifications that allow each UAV to control its sensing constraints (the detailed description of these modifications are in the simulation experiments in Section 7.2.2). In contrast with the previous section, the leader remains static in the environment, and it was manually rotated in order to keep its active beacon marker turned to the followers. No other sensors (either height or OF) were used besides the relative positioning system. The relative positioning system was run at 40 *Hz*, which corresponds to the value set in Section 7.1.1.2. The formation control algorithm was run at the same frequency. An integrator was also included in the height controller of each follower, in order to allow the flying UAV followers to compensate any height differences between themselves and the static leader.

In each experiment, the chosen triangular structure in Fig. 7.25 with $l_{triangle} = 1.5\ m$ was chosen as the target formation configuration for the UAVs. The desired height of the formation was 1 *m*, and the desired relative height for all UAVs was set to zero. Also, similarly to what was done in the experiments of the previous section, the desired relative attitudes between the UAVs were set in order to direct the sensing camera of the followers towards their local neighborhoods, optimizing the FOV of the sensing cameras in the sense of the desired sensor direction defined in Section 5.2.1. In each experiment, the leader UAV was placed on a fixed position in the environment. The experiment started by taking off and move the two UAV followers to their initial poses, using the safety controller and the MCS feedback. Note that the initial poses had to ensure that all cameras on board the UAV followers were capable of observing the neighborhood of the UAV (the leader and the other follower), similarly to what was performed in the experiments of the previous section. After the UAVs have achieved their initial poses, the actual experiment would start by cutting all the connections between the followers and the MCS while activating their formation controllers.

During each experimental run, the pose of each UAV was tracked and the relative ranges and bearings between the two UAVs were computed for analysis. The error between the desired and the actual relative range and bearing between the two UAVs throughout the experiment run was then computed. In this section the inter-edge apertures, as defined in Section 5.2.1, were also computed, in order to assess the controllability of the FOV constraints. Figs. 7.26c and d show the progress of the measured and the actual horizontal and vertical relative range of the neighbors observed by one of the flying UAV followers. These results show that both followers are successfully maintaining the desired relative ranges between their neighbors. The maximum relative range error observed in the experiments was 30 *cm* for the horizontal range, which is similar to the statistical results observed in the experiments of the previous section (see histogram of Fig. 7.24a and consider only the negative side, which is not influenced by the motion lag between UAVs). The vertical range errors were slightly larger, of about 40 *cm* with respect to the desired height. These errors are attributed to the integrator used in the height controller.

Figs. 7.26a and b show the relative bearings and inter-edge apertures for the neighborhood observed by one of the flying UAV followers. The results show that the relative bearings converge to symmetric



(c) $l_{triangle} = 1.5\ m$

|  | $n = 1$ | $n = 2$ | $n = 3$ |
|---|---|---|---|
| $x^{\mathcal{F}}_{fi,d}$ | $0.53\ m$ | $-0.53\ m$ | $-0.53\ m$ |
| $y^{\mathcal{F}}_{fi,d}$ | $0.0\ m$ | $-1.06\ m$ | $1.06\ m$ |
| $z^{\mathcal{F}}_{fi,d}$ | $1.0\ m$ | $1.0\ m$ | $1.0\ m$ |
| $\psi^{\mathcal{F}}_{fi,d}$ | 0.0° | 67.5° | −67.5° |

Desired formation geometric configuration

|  | $n = 1$ | $n = 2$ | $n = 3$ |
|---|---|---|---|
| $n = 1$ | $0\ m$ | $1.5\ m$ | $1.5\ m$ |
| $n = 2$ | $1.5\ m$ | $0\ m$ | $2.12\ m$ |
| $n = 3$ | $1.5\ m$ | $2.12\ m$ | $0\ m$ |

Desired inter-vehicle ranges

|  | $n = 1$ | $n = 2$ | $n = 3$ |
|---|---|---|---|
| $n = 1$ | 0° | 45° | −45° |
| $n = 2$ | −22.5° | 0° | 22.5° |
| $n = 3$ | 22.5° | −22.5° | 0° |

Desired inter-vehicle bearings

Figure 7.25: Performance of the formation control algorithm with FOV constraint control in real experiments. (a) The desired formation configuration defined in the formation frame. (b) Evolution of the UAV trajectories performed in one experiment run. Each UAV is represented by a black triangle, and each color illustrates a trajectory of a specific UAV. The triangle orientation is the same as the sensor direction ($\mathbf{r}_{s_i}$). (1), (2) and (3) represent snapshots during the experiment, where (1) corresponds to the initial follower poses. (c) Details of the desired configuration for the chosen sensing and control graphs for $l_{triangle} = 1.5\ m$.

values with respect to $\mathbf{r}_{s_i}$, meaning that the altered attitude controller optimizes the sensor FOV when observing multiple neighbors. Additionally, Fig. 7.26b confirms the convergence of the inter-edge aperture to the desired values. This, in combination with the results from Figs. 7.26c and d, suggests the correct behavior of the formation control algorithm with FOV constraint control.

Finally, the results show a rotation behavior of the UAV followers around the leader, similarly to what was observed in the previous section (see Figs. 7.25 and 7.23). This rotation behavior was attributed to sensing and actuation inaccuracies of each UAV. An example of a sensing inaccuracy is the sensor bias on the horizontal plane shown in Fig. 7.26a, observed by the differences between the tracked relative ranges by the MCS and relative positioning sensor. These biases create unwanted forces competing with the formation control algorithm. Since the algorithm forces are tangent to the edge between UAVs, the

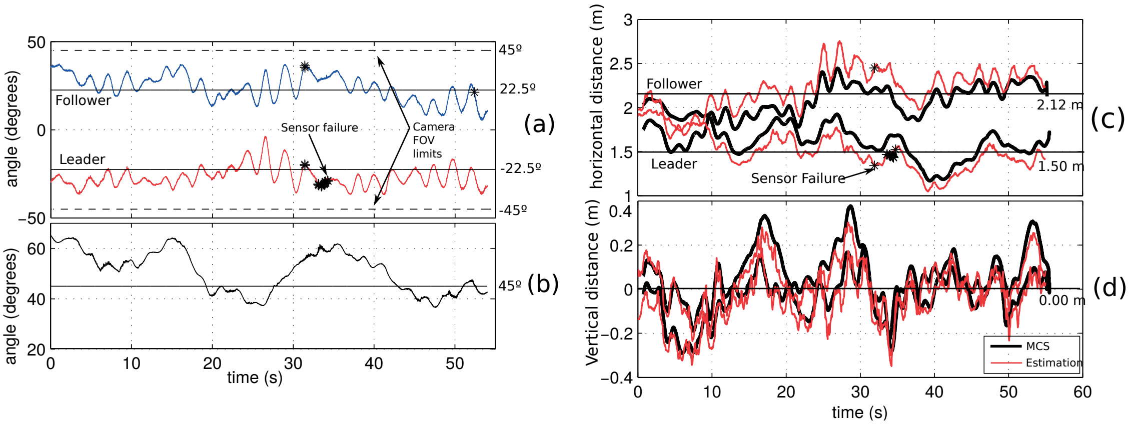


Figure 7.26: Neighbor bearings (a), aperture (b), and relative ranges observed in the local frame of one of the UAV followers, during one real experiment using the formation control algorithm with FOV constraint control. The relative range is divided into horizontal (c) and vertical (d) components. Desired bearings, apertures, and ranges are depicted on the respective plot using horizontal lines, with their values written on the right. The neighbor to which each bearing or horizontal range line corresponds is indicated in text on the left of the line. Vertical ranges of both neighbors are indistinguishable since they are both close to zero. The aperture is related to both neighbors. All values are tracked using the MCS, but on the range plots, the values estimated from the sensor data are also shown. Black dots on the lines signal moments where the follower stopped receiving sensor data of the respective neighbor for more than 100 *ms*.

algorithm becomes weak on the radial axis, allowing relatively small forces to still be able to generate rotation movements. The biases can be different for each UAV, and it could happen that they generate rotations in different directions. This would lead to a steady increase of the follower aperture observed by the leader. In this case, the leader would perform inter-edge aperture control (see Eq. (5.21)), going backwards to maintain the desired value. If these biases are constant, the system would move backwards until the end of the experiment. In the next section, an OF sensor will be able to substantially reduce this rotation behavior, by allowing velocity control of the UAV in the environment.

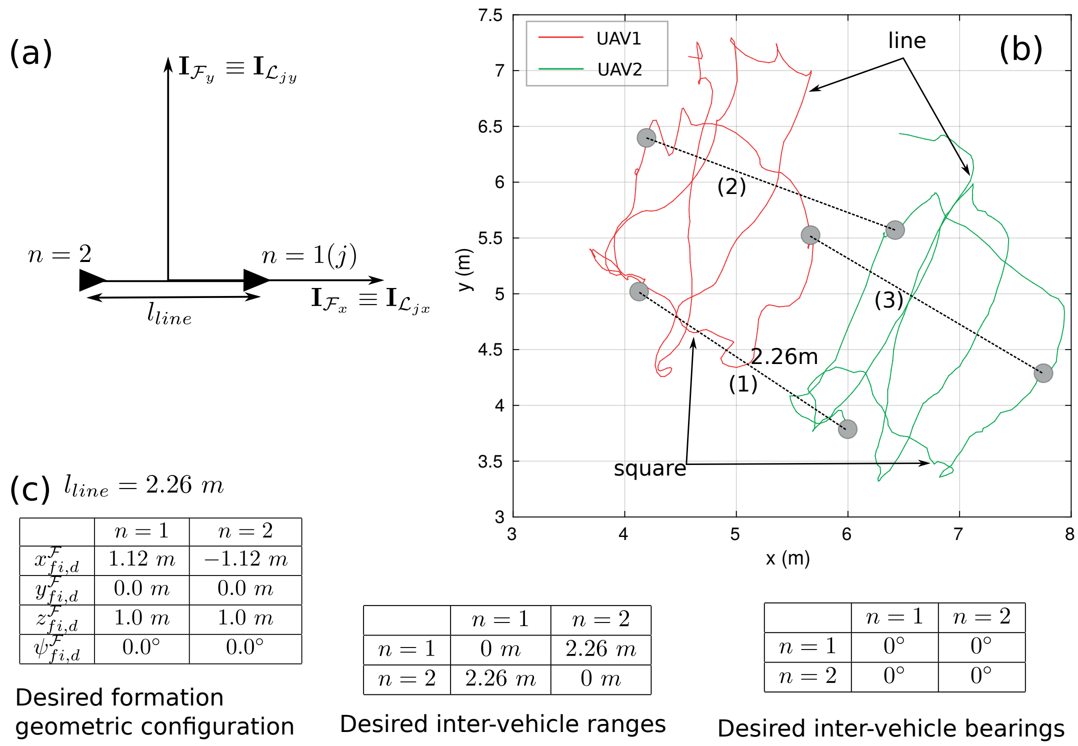
### 7.3.3 Formation steering with the infrared-based system

Finally, the sensing and control stack was tested in a situation where the formation moved in the environment using the virtual structure approach developed in Section 5.2.2. These experiments used the UX-401 platform. In order to disregard the FOV constraints discussed in the previous sections, the chosen onboard relative positioning system was the IR-based system, given its 360° FOV. The system was calibrated in the LBL calibration arena with the method described in Section 7.1.2.1. The UAVs were then flown in the Pavilion flying arena. During the experiments, the pose of each UAV was tracked using the mounted ad-hoc MCS described in Section 6.2.2. However, as explained in that section, the UAVs have to maintain a fixed attitude in the environment so as to allow this MCS to measure the relative bearing between them. In these experiments, the attitude of both UAVs was set to the magnetic north direction. The UAVs controlled their attitudes to this direction using sensory feedback from the magnetometer sensor present in the onboard IMU.

The conducted experiments were similar to the real experiments conducted in Section 7.3.1, with also two flying UAVs. Only one UAV ($i$) had a top IR receiver station pointing upwards (as depicted in Fig. 5.19) and was able to sense the other UAV ($j$ or $n = 1$ in Fig. 7.27a). However, in the current experiments the entire formation, not just the leader, was moved at the same time using the virtual structure approach. The maneuvers were achieved by sending a set of coordinated formation motion commands to the UAVs, defined in Section 5.2.2, with the help of a wireless communication network. Given that in these experiments both UAVs had to be facing the north direction, it was decided only to provide translation commands to the formation. Only horizontal velocity-commands ($\mathbf{u}_{vf_h} \neq 0$) were considered in the experiments. Given the absence of a reliable MCS, the motion commands were generated by a human teleoperator using the sensory feedback provided by his own eyes, instead of the autonomous system implemented in the simulation experiments of Section 7.2.3. The motion commands were translated on each UAV by implementing Eq. (5.25). Since UAV $j$ could not sense its neighborhood, the formation frame used in this equation ($\mathbf{I}_{\mathscr{F}}$) was replaced by UAV $j$'s flying frame ($\mathbf{I}_{\mathscr{L}_j}$). Therefore, UAV $j$ can immediately translate the motion commands without additional sensor measurements. UAV $i$ measures $\mathbf{I}_{\mathscr{L}_j}$ using the attitude measurement provided by its onboard IR-based relative positioning sensor. This would not be possible using other IR-based systems reported in the literature since they could not provide relative attitude measurements.

The translation of the motion commands using Eq. (5.25) gives the desired height and horizontal velocities ($z_{i,d}^{\mathscr{L}_i}, \mathbf{v}_{h_i,d}^{\mathscr{L}_i}$) for the UAVs to follow. The UAVs control these quantities by implementing the baseline formation control algorithm, defined in Eqs. (4.12) and (4.13). Only UAV $i$ implements the part of that algorithm that maintains the relative positioning between neighbors, defined in Eqs. (4.9) and (4.10), since only it has the capability of acquiring inter-vehicle localization information from its neighbors. OF and height sensors were mounted on UAVs in order to provide the sensory feedback to control ($z_{i,d}^{\mathscr{L}_i}, \mathbf{v}_{h_i,d}^{\mathscr{L}_i}$) using the previous equations. Additionally, these sensors help to avoid the drifts and rotations observed in the previous experiments, due to sensing and actuation inaccuracies. Note that the relative positioning system on board UAV $i$ was run at 80 $Hz$, which corresponds to the value set during the analysis of the IR-based system conducted in Section 7.1.2. The formation control algorithm on both UAVs was run at 40 $Hz$, frequency used in the previous real experiments.

In these experiments, the linear structure in Fig. 7.27a with $l_{line} = 2.26$ $m$ was chosen as the target formation configuration for the UAVs. The desired height of the formation was 1 $m$, and the desired relative height for all UAVs was set to zero. In contrast with the real experiments conducted in the previous sections, no optimal sensing directions were considered nor did the UAVs have to go to any specific initial positions. In the current experiments, the two UAVs take off at the same time with their formation control algorithm already initialized and they stabilize in the environment using their OF and height sensors. This was possible because the IR-based system has a 360° horizontal FOV. In fact, it would not have been possible to control the system in other way given the lack of a reliable MCS to



| | $n = 1$ | $n = 2$ |
|---|---|---|
| $x^{\mathcal{F}}_{fi,d}$ | $1.12\ m$ | $-1.12\ m$ |
| $y^{\mathcal{F}}_{fi,d}$ | $0.0\ m$ | $0.0\ m$ |
| $z^{\mathcal{F}}_{fi,d}$ | $1.0\ m$ | $1.0\ m$ |
| $\psi^{\mathcal{F}}_{fi,d}$ | $0.0°$ | $0.0°$ |

Desired formation geometric configuration

| | $n = 1$ | $n = 2$ |
|---|---|---|
| $n = 1$ | $0\ m$ | $2.26\ m$ |
| $n = 2$ | $2.26\ m$ | $0\ m$ |

Desired inter-vehicle ranges

| | $n = 1$ | $n = 2$ |
|---|---|---|
| $n = 1$ | $0°$ | $0°$ |
| $n = 2$ | $0°$ | $0°$ |

Desired inter-vehicle bearings

Figure 7.27: Performance of the formation control algorithm using a virtual structure approach during real experiments with two UAVs. (a) The desired formation configuration defined in the formation frame. Note that this frame has the same direction than the flying frame of UAV $n = 1$. (b) Evolution of the UAV trajectories performed in one experiment run. Each UAV is represented by a black circle, and each color illustrates a trajectory of a specific UAV. (1), (2) and (3) represent snapshots during the experiment. (c) Details of the desired configuration for the chosen sensing and control graphs for $l_{line} = 2.26\ m$.

teleoperate the individual UAVs to the correct initial positions. This shows the advantage of the extended FOV capabilities of the IR-based system over the camera-based system. Given some period of time after which the formation have stabilized at the desired height, the human operator steered the formation as a virtual structure in order to achieve a square or a line trajectory, as depicted in Fig. 7.27b. The maximum horizontal speed of the motion commands was set to 0.65 $m/s$. No vertical motion commands were issued, similarly to the real experiments initially conducted in Section 7.3.1.

During each experimental run, the ground truth for the position of each UAV was acquired using the ad-hoc MCS. From the acquired data, the relative ranges, bearings and elevations between the two UAVs were computed. The error between the desired and the actual values for these quantities was then analyzed. Fig. 7.28 shows, for one experiment run, the relative range, bearing, and elevation between the two UAVs measured by the relative sensor and the MCS. The sensor measurements are shown to closely follow the ground truth values, except for the relative bearing in the first seconds of the experiment. This happens because the ground truth for the relative bearing is computed assuming that the UAVs are always pointing to the north direction, as previously explained. However, at the beginning of the experiment the

UAVs can be placed with any attitude. It is only a few seconds after the UAVs have taken off that their attitude converges towards the north direction. This makes the ground truth for the relative bearing to take a while to align with the values measured by the relative sensor, as shown in the Fig. 7.28b.

Figs. 7.28a and c show that the formation controller in closed-loop with the developed relative positioning system is able to maintain the relative range and elevation between between UAVs around the desired values. However, as observed in Fig. 7.28a, when motion commands are issued to the formation (vehicle motion) the range error increases. This error should not exist as the UAVs receive the motion commands at the same time, resulting in a simultaneous actuation in the same direction. However, sensing and actuation inaccuracies can cause these errors, as seen in the simulation experiments of Section 7.2.3. In simulation, these errors were smaller most likely because the modeled sensing and actuation noise was much smaller than that seen in reality (for example, the propeller thrust and OF noise models were not captured realistically in simulation, and they can seriously degrade the velocity control performance in the desired direction). These inaccuracies generate distortions to the formation configuration that can grow fast. These distortions might easily break the constraints of limited relative positioning systems, precluding the UAVs from recovering the desired configuration. Such issues can in particular occur with a camera-based relative localization system, where such distortions might even lead to to situations in which one of the observed neighbors is getting out of the FOV. When the infrared-based relative localization system is leveraged, the UAV can always track its neighbors thanks to its larger FOV while the onboard formation controller reduces the error back to reasonable values, as shown in Fig. 7.28a. This example again shows the advantage of the extended FOV capabilities of the IR-based system over the camera-based system.

The results from Fig. 7.28b show that the relative bearing between the UAVs does not vary much. Also, Fig. 7.27b shows that both UAVs produced similar trajectories (in contrast with the experiments conducted in Section 7.3.1, where the follower UAV seemed to rotate around the leader UAV). These results suggest that the rotations caused by sensing and actuation inaccuracies were substantially reduced. The cause of this reduction is due to the use of the OF sensor, which allows the UAVs to control their velocity in the environment.

Finally, to assess the overall performance of the formation control algorithm, the previous experiments were repeated several times and a statistical analysis of the relative range error with respect to the desired values was conducted. The results of this statistical analysis are shown in Fig. 7.29. These results show that the relative horizontal and vertical range error closely follows a normal distribution. The distribution of the horizontal error is averaged around 0 *cm* and it has a standard deviation of 27.2 *cm*, which correspond respectively to 0% and 12% of the desired range of 2.26 *m*. The distribution of the vertical error is measured in terms of the relative elevation between the UAVs. This distribution is averaged around 2° and with a standard deviation of 5°. The standard deviations of the previous distributions are close to the ones obtained for the distributions acquired in Section 7.3.1 (when related in percentage to the desired horizontal range of each experiment). This makes sense since the developed camera-based system (used in the real experiments of that section) and IR-based system (used in the current exper-

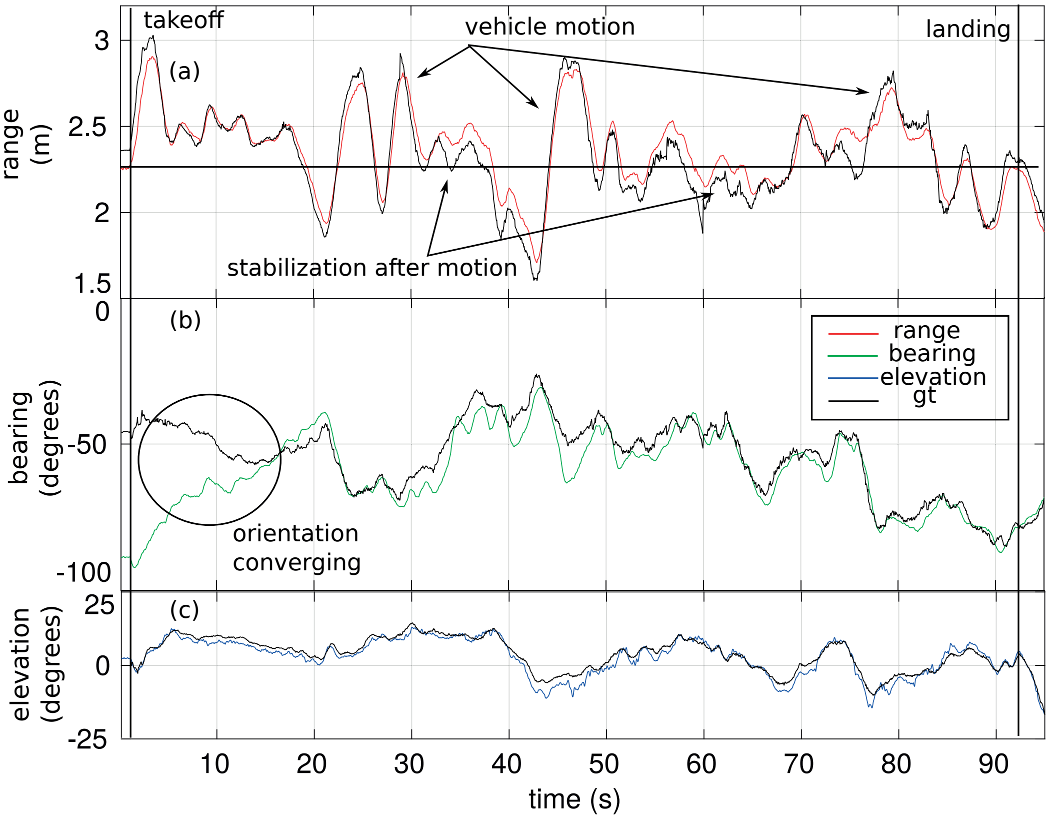


Figure 7.28: Neighbor relative range (a), bearing, (b), and elevation (c) measurements acquired by the IR-based relative system onboard one of the UAVs compared to the ground truth during an experiment run with the two UAVs. Note the takeoff and landing events.

iments) were shown to have similar accuracy performance (see Section 7.1 for accuracy performance details of the developed relative positioning systems).

However, note that the non-zero average shown in the horizontal error distribution in Section 7.3.1 (see Fig. 7.24) could be removed in the current experiments. Recall that the non-zero error average was related to the motion lag produced by the leader-follower approach in steering. Therefore, the average being removed in these experiments show that using a virtual structure approach to steer the formation removes the motion lag between the UAVs, as also confirmed in the simulation experiments of Section 7.2.3. Finally, note that the average of the vertical error distribution is also non-zero. However, no motion commands in the vertical direction were issued. Therefore, this average is attributed to small biases between the actual and desired height of each UAV throughout the experiments. These biases can be easily generated by small dissimilarities in the thrust of the UAVs, often influenced by different discharge curves of the onboard batteries.

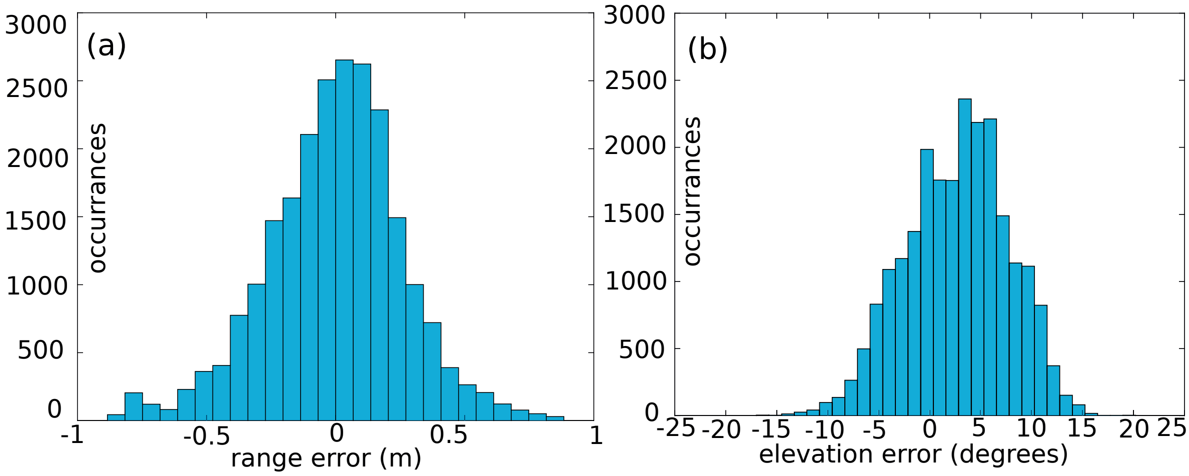


Figure 7.29: Statistics for the relative horizontal (a) and vertical (b) range errors between two UAVs with respect to the desired values, during several runs of the formation control algorithm using a virtual structure approach with two UAVs. Note that the vertical error is represented by the relative elevation between the two UAVs.

## 7.4 Conclusions and discussion

This section compares the two types of relative positioning systems used in this work, in Section 7.4.1, and discusses the results acquired for the formation control system, in Section 7.4.2.

### 7.4.1 Relative localization systems

In this work, two relative positioning systems were developed, each one operating with a different technology. Both systems enable a UAV to acquire 3D relative pose and ID measurements of multiple neighboring UAVs within the sensor FOV. The systems are designed in the way that allow bidirectional sensing between two neighboring UAVs using similar systems. This means that if a UAV $i$ is able to the detect another UAV $j$, then UAV $j$ is also able to detect UAV $i$. The specifications of these systems are summarized in Table 7.3. For more information about these specifications the reader is referred to Sections 7.1.1 and 7.1.2.

The developed camera-based system places a camera and a multi-beacon marker on each UAV $i$. The camera is able to provide measurements that allow the acquisition of the 3D relative pose of UAV $j$'s marker. As discussed in Section 5.1.1.1, this marker has a set of localization and ID beacons. This allowed generating multiple marker IDs without using different geometrical configurations, an option which would require a careful choice of marker positions to prevent possible ID misclassification, and would increase the computational complexity of the corresponding classification algorithms with the number of IDs involved (note that this was indeed reported in previous literature presenting this type of systems). Additionally, the development of an accurate noise model through Eqs. (5.6), (5.8) and (5.10) allowed for an implementation of an algorithm that was able to track accurately the position and velocity

|  | Camera-based (320x240) | IR-based | Camera-based (literature) [29] (752x480) | IR-based (literature) [92] |
|---|---|---|---|---|
| Power | $2\ W$ | $7\ W$ | NA | $10\ W$ |
| Weight | $60\ g$ | $110\ g - 200\ g$ | NA | $245.2\ g - 400\ g$ |
| FOV | $90^\circ$ | $360^\circ$ | $90^\circ$ | $360^\circ$ |
| Max. range | $3.5\ m$ | $4\ m$ | $5\ m$ | $12\ m$ |
| Accuracy errors |  |  |  |  |
| Range | $< 20\ cm$ | $< 20\ cm$ | $< 10\ cm$ | $< 20\ cm$ |
| Bearing | $< 5^\circ$ | $< 5^\circ$ | $< 3^\circ$ | $< 5^\circ$ |
| Elevation | $< 2^\circ$ | $< 10^\circ$ | $< 3^\circ$ | $< 3^\circ$ |
| Attitude | $< 15^\circ$ | $< 20^\circ$ | $< 2^\circ$ | unavailable |
|  |  |  |  |  |
| Delay | $150\ ms$ | $< 50\ ms$ | ? | ? |
| Frequency | $17\ Hz - 20\ Hz$ | $83\ Hz - 1.66\ KHz$ | $40\ Hz$ | $200\ Hz - 1\ KHz$ |

Table 7.3: Comparison between the different relative positioning systems. The displayed accuracy errors correspond to experiments conducted with ranges between the sensor and the target of up to 3 $m$.

of neighboring UAVs, despite the low resolution of the deployed camera. This tracking algorithm was then shown in Section 7.3.1 to be able to stabilize the UAVs without the help of additional sensors. Moreover, as discussed in Section 7.1.1.2, the corresponding underlying model can be applied in order to chose the system design parameters (FOV, camera resolution, circumsphere size, etc.) according to the accuracy requirements of the relative positioning system. For example, this model could answer questions such as the following: if the system requires a larger FOV than the one currently used, how much larger could the FOV be in order to still allow the system to achieve a certain accuracy without increasing the computational requirements (i.e., keeping constant the camera resolution)?

The developed IR-based system places multiple receivers and a multi-beacon marker both on board each UAV $i$. The set of receivers of UAV $i$ is able to provide measurements that allow the acquisition of the 3D relative pose of UAV $j$'s marker. The developed system achieved several hardware enhancements with respect to the systems of the previous literature using the same technology. Firstly, as observed in Table 7.3, the maximum weight and power of the system developed in this thesis (composed of the four beacons and the two receiver stations), 200 $g$ and 7 $W$ respectively, is less than the ones reported in the literature [92], requiring a weight that can go up to 400 $g$ and a power of 10 $W$. Additionally, the development of small omni-directional IR beacons simplified the design of the emission sources on each UAV, and also allowed the use of several emission sources for each UAV. It was then shown that multi-emission sources on each UAV enable the development of a positioning system that could acquire the attitude measurements of neighboring UAVs (this feature was not present in previous IR-based systems [86, 92]). Finally, note that the communication algorithms of the sensor were directly implemented on the IR channel (instead of [92] that synchronized all the nodes of the system using a RF channel parallel to the IR channel). Since the RF emissions have larger ranges than the IR emission, the system has the potential of becoming scalable with the number of vehicles. For now, a communication algorithm based on a TDMA communication protocol is implemented, as in [92], which compromises the scalability properties of the system. However, CSMA (implemented in [86]) or adaptive TDMA protocols could also be considered to reintroduce this scalability.

The developed IR-based system also achieved several software enhancements. Firstly, the developed localization algorithm considers each receiver and each beacon separately, as discussed in Section 5.1.2. Secondly, the developed calibration algorithm compensates for wrong receiver placement on board the UAV, as observed in the results of Section 7.1.2.1. These two enhancements combined have three consequences on the IR-based system. The first consequence is that, together with the use of multiple beacons, the system is able to acquire the relative attitude of neighboring UAVs. The second consequence is that the receiver placement can be arbitrary, and the placement procedure can be easily adapted to any 3D geometry without the need of extra supporting structures. The third consequence is that the mechanical design of the system does not need to be strict, since placement errors of the receivers can be corrected after the system deployment through calibration.

The two developed relative positioning systems can be compared with respect to their mechatronic implications and their localization capabilities. Regarding the mechatronic implications, it can be easily observed from Table 7.3 that the camera-based system consumes less energy, given its low weight and power requirements (only 60 $g$ and 2 $W$ respectively) compared with the IR-based system (up to 200 $g$ and 7 $W$ respectively). This is a very important advantage of the camera-based system given that the UAVs are always power-constrained. The most important factor is the weight of the system, since the power of both systems is well below the power needed by the propellers (which is usually higher than 40 $W$). For example, the Hummingbird quadrotors used for the experiments in Section 7.3.1 have a 500 $g$ mass (including battery). Placing the full IR-based system will increase the mass of the system up to 40 % (and therefore consuming more energy and allowing less flight time), while placing the camera-based system will only increase it to 12 %.

Regarding the localization capabilities, it is clear from Table 7.3 that the IR-based system is able to provide an accuracy comparable with the one of the camera-based system presented in this work. Systems with higher camera resolutions, such as (752x480) in [15,29], have greater accuracy but a several times larger computational cost. This prevents them from having larger FOVs on resource-constrained robots such as small-scale UAVs. The proposed IR-based system not only is shown to have a 360° FOV, but it also provides higher measuring frequencies than the camera-based systems. This system currently runs at 80 $Hz$, which is substantially higher than the maximum reported frequency of 40 $Hz$ in the previous described camera-based systems. Larger measuring frequencies also mean less sensing delay (50 $ms$ for the IR-based system compared to the 150 $ms$ for the camera-based system). The measuring frequency of the IR-based system can be further increased according to the number of neighbors and beacons considered for each UAV, as discussed in Section 7.1.2.2. The current system is designed for detecting six UAVs, each with four beacons. If each UAV is described by a single beacon the theoretical measuring frequency of the systems increases to 333 $Hz$, and if the system would only require to detect one single IR beacon the theoretical measuring frequency of the system increases to 1666 $Hz$. However, this value might be unrealistic given the communication limitations between the system and the onboard computer that runs the localization algorithm (which is running probably not based on a RT operating system). Note that these possible measuring frequencies are larger than the ones predicted for the work using the system in [92].

The last point gives the IR-based system a very important advantage when it comes actually to perform multi-UAV interaction. Not only the sensors have larger measuring frequencies (allowing for more reactive maneuvers of the neighbors) but also they have an extended FOV and an accuracy comparable with the camera-based system. As observed in the results of Section 7.2.1, sensor FOV constraints can severely limit the number of possible geometric configurations achievable by the formation and can also reduce the reactiveness of the entire formation. Additionally, as observed in Section 7.3.3, sensing and actuation inaccuracies generate distortions to the formation configuration that can grow fast. These distortions might easily break the constraints of limited relative positioning systems, precluding the UAVs from recovering the desired configuration. For example, the neighbors observed by an UAV using the developed camera-based system (with FOV constraints) might leave the FOV of the camera sensor when these distortions occur. However, the developed IR-based system has 360° FOV and does not suffer from the previous problem. Note that it is also possible to reduce the FOV of the IR-based system to a 180° FOV (which is still substantially larger than the one used by the camera-based system) by using just one receiver station, in order to save weight. For this last example, the system implemented with one receiver station is about 110 $g$, which on the Hummingbird quadrotor means an increase of 20% of the mass of the system, a design solution generating an overweight of the UAV closer to that of camera-based technology.

### 7.4.2 Formation control

The relative positioning systems developed in this thesis are used to allow a group of UAVs to move in formation using exclusively onboard sensors and control algorithms. The formation control problem is addressed by making use of a graph-based formation control algorithm, relying on relative inter-vehicle localization measurements from sensors on board each UAV. In this work, adaptations are made to the formation control algorithms reported in previous literature in order to tackle limitations that arise from the use of noisy, FOV limited relative localization.

The first limitation is related to the sensing constraints of the relative positioning systems (mainly the FOV constraints). These constraints severely limit the number of possible geometric configurations achievable by the formation and can also reduce the reactiveness of the entire formation. To tackle this limitation, a well-established formation control algorithm reported in previous literature is modified in order to natively consider the FOV constraints of the onboard sensor, so they will be respected during operation. The simulation results presented in Section 7.2.2 show that the altered algorithm is able to overcome the severe convergence issues observed when the FOV constraints of the used camera-based system are not considered. The real experiments presented in Section 7.3.2 show that the altered algorithm allows the UAV to optimize the FOV of its camera when observing multiple neighbors, in the presence of sensing and actuation inaccuracies of real systems.

The second limitation is related to the steering of the formation to the desired goal in the environment. From the simulation results in Section 7.2.2, it was observed that moving the formation with a leader-follower approach generate strong distortions on the formation geometry. These distortions are caused by motion lag between the leader and the followers, limiting the maximum velocity of the leader, and therefore the reactiveness of the entire formation. The virtual structure approach developed in this work to move the formation was shown to be able to reduce these distortions (see Section 7.2.3). The real experiments presented in Section 7.3.3 show, that with sensing and actuation inaccuracies of real devices, the formation steered by the novel algorithm is able to reduce the effects of the motion lag caused by a leader-follower approach. The novelty of this algorithm is that it relies exclusively on relative inter-vehicle localization, and no extra communication between the UAVs or features external to the formation (which was the case of previous literature) are required.

It is worth noting that the real experiments reported in Section 7.3 show that it is possible to achieve formation control, and more generally multi-UAV coordination, using exclusively the onboard relative positioning systems and other auxiliary onboard sensing. For example, the OF sensor was used to substantially remove unwanted formation rotation behaviors, caused by inaccuracies on relative sensing and actuation, and to provide sensory feedback for the UAV velocity control when applying the formation steering algorithms (see experiments in Section 7.3.3). Additionally, it is shown that the proposed steering algorithms allow a centralized system (in this work an external system was used, but it could also be a set of UAVs of the team) to steer the multi-UAV system as a whole in the environment without requiring the knowledge of the specific absolute positions of each UAV of the system. Therefore, this overall control approach substantially removes the complexity of the centralized systems which would require complex localization, communication and planning algorithms in order to achieve the behaviors for each UAV (as in [3, 110]).

Finally, it is worth recalling that the set of experiments conducted in this work was performed using bidirectional sensing and control between the UAVs, as referred in Section 4.2.2. Such assumptions can also limit the number of geometric configurations achievable by the formation. For example, in this work, only formations that place the UAVs on the convex hull of the target formation shape were considered, so that problems associated to occlusions, and to the FOV limitations of the sensors could be easily be solved. However, the developed positioning systems and control algorithm can also allow additional unidirectional connections between UAVs. This would relax the bidirectional constraints assumed in this work, and therefore increase the number of possible formation configurations.

### 7.4.3 Videos and multimedia

Additional videos concerning the work conducted in this thesis can be found in:

https://www.dropbox.com/sh/99oc9ykmgfn4jc5/AAC24LO0kNgKbKrHIQsQuvgGa?dl=0

# Chapter 8

# Conclusions

The goal of this thesis is to contribute in the field of aerial robotics by proposing solutions to some challenges posed by the design of teams of multiple flying vehicles operating in the absence of global positioning systems, such as GNSS-based technology. Specifically, this work addresses the requirements of UAVs to detect the pose and velocity of their team members in a three dimensional environment while having inherent energy and weight design constraints. For this purpose, two different positioning systems that allow each UAV to obtain the relative pose of other team members have been proposed. These relative positioning systems focus on maximizing the number of detected team members while remaining accurate and light enough to allow their deployment on the UAVs and ensure reliable spatial coordination between multiple UAVs.

The first relative positioning system is based on computer vision. A camera-based system has been developed in a way that allows its deployment on multiple UAVs without the need for choosing the unique geometrical configurations of markers for each vehicle. Using different geometrical configurations would require a careful choice of marker positions to prevent possible ID misclassifications, and would increase the computational complexity of the used classification algorithms with the number of IDs (as confirmed by previous contributions). On the contrary, this new marker ID allows its deployment and usability in a way that is scalable with the number of UAVs in the team. Additionally, the development of an accurate sensor model of the system was conducted, allowing the choice of system design parameters (FOV, camera resolution, circumsphere size, etc.) according to the accuracy requirements of the system.

The second relative positioning system leverages infrared technology. It features several hardware and software enhancements with respect to the systems of the previous literature using the same technology. Regarding the hardware enhancements, the weight of the system developed in this work is at least two times lighter than the ones reported in the literature, and it also requires less power during its operation. Additionally, the development of small omni-directional IR beacons simplifies the mechanical layout of the emission sources on each UAV, and also allows the use of several emission sources for each UAV. Thanks to the presence of multiple emission sources on each UAV, our relative positioning system is able to acquire the attitude measurements of neighboring UAVs (this feature was not present in previous infrared-based systems). Finally, the developed infrared devices allows also for explicit communication, a key feature for further promoting scalability in the number of vehicles. Regarding the software enhancements, the developed localization and calibration algorithms for this relative positioning system allow for: the extraction of the relative attitude of neighboring UAVs; the placement of IR receivers in arbitrary poses on the UAV, making the placement procedure easily adapted to any 3D geometry without the need of extra supporting structures; a simplification of the system deployment process since placement errors on the IR receivers can be corrected through calibration.

Moreover, this thesis proposes novel formation control and formation steering algorithms with the goal of maximizing the reactiveness of the multi-UAV team in a way that is scalable with the number of used team members. The novelty consists of improvements made on the formation control algorithms reported in previous literature in order to tackle limitations that arise from the use of relative localization. The first limitation is related to the reduced number of inter-vehicle interactions originated from the sensing constraints of the relative positioning systems (mainly the FOV constraints). The problem leads to a low number of possible formation configurations and can also reduce the reactiveness of the entire system. To tackle this limitation, this work alters a typical formation control algorithm reported in previous literature in order to directly control the FOV constraints of the onboard sensor, so they will be kept during formation operation. The results show that this algorithm allows the system to achieve additional formation configurations with the same limited sensors, and it enables each UAV to optimize the FOV of its sensor when observing multiple neighbors. The second limitation is related to the problem of moving the formation with only local relative localization information. It was observed that moving the formation with a leader-follower approach generate strong distortions on the formation geometry. These distortions are caused by motion lag between the leader and the followers, limiting the maximum velocity of the leader, and therefore the reactiveness of the entire formation. An external system moving the formation as a whole using a virtual structure approach is shown to reduce these distortions. This implementation has been achieved without requiring extra communication between the UAVs or features external to the formation (which was the case of previous literature). Furthermore, the external system does not require the specific absolute positions of each UAV of the system. Therefore, this overall control approach substantially removes the complexity of the external systems, which would require complex localization, communication and planning algorithms in order to achieve the behaviors for each UAV (as in [3, 110]).

## 8.1 Potential applications

This work presents a set of sensing and control solutions designed to enable the coordination of multiple autonomous UAVs operating in indoor environments or in environments where GNSS-based technology is not available. The UAV's ability to obtain the relative inter-vehicle localization of its neighbors independently from any external systems or its localization in the environment provides an approach that can be deployed and operated in a simpler way in uninstrumented arbitrary environments.

One interesting advantage of using multiple UAVs is that the resulting team has the ability to provide extended control capabilities for manipulating or carrying objects. For example, they can be used to remove debris from collapsed buildings in search and rescue missions. Given that the geometry of the debris cannot be specifically controlled, their inertial properties might be such that one UAV might not be enough to handle them in the desired way (that does not compromise safety or the integrity of nearby structures). Another possible environment in which such operations could be useful would be that of spacecrafts on board space stations orbiting earth, such as the ones described in [74]. In these environments, these UAVs can be used to carry objects between different facilities of the station, helping astronauts in their daily chores. If multiple UAVs are placed on different sides of a large object, they would be able to instantly provide thrust in different directions, a possibly way more flexible solution than that involving a single UAV and providing the right thrust direction by changing its pose serially. Therefore, in these scenarios multiple UAVs can optimize how the system handles the object, saving fuel and operation time.

Another interesting advantage of using multiple UAVs, as discussed in Chapter 1, is that the geometry of these systems can be exploited to minimize the impact of individual UAV limitations. For example, multiple UAVs can carry heavier objects for largest distances (either in construction sites or in search and rescue scenarios) which allows them to be deployable on larger scale missions. Another example is represented by mapping and aerial surveillance tasks. When using small UAVs with low FOV cameras, by combining the sensory input of all UAV from the team it is possible to improve the FOV of the combined system. In this way mapping and surveillance is conducted in a more efficient way, and the system is able to plan its operation in a more optimal way.

One interesting note regarding the previous discussion is that, as the desirable number of UAVs of the team increases, so does the complexity of the multiple UAV coordination. A key requirement is that the system remains scalable. For example, the number of communication links required for a single UAV should not grow linearly with the number of UAVs of the formation, or the unit coordinating the system (usually an external system communicating with the UAVs) should not have to plan every single action of each agent, in order to not compromise the system reactiveness. According to Section 2.2.2, in a behavior-based approach, each agent interacts with its local neighborhood and the environment. The locality of these interactions are able to be implemented in a distributed fashion, which helps maintaining the system reactiveness while remaining scalable with the number of UAVs. Additionally, it creates more flexible systems when faced with unpredictable and unprepared environments, since simple behaviors (avoid obstacles and other robots, maintain distance to neighbors, move closer to the objective, etc.) typically hold.

The use of onboard relative positioning systems help implement these distributed behaviors since UAVs can extract information directly from their neighbors with minimal or no communication overhead. Moreover, some of these sensors can allow a rough localization of nearby obstacles. For example, the infrared-based positioning system developed in [92] is able to roughly detect obstacles (and their bearing) by sensing reflections produced by close obstacles. These properties can also be leveraged in our hardware, as discussed in Section 7.1.2.1). This sensing information enables the UAV to perform obstacle avoidance without the use of additional sensors dedicated to obstacle detection, and thus saving the already constrained energy and weight of the vehicle. Finally, the results show an implementation of an hierarchical coordination approach, which retains scalability without compromising adaptability and reactiveness on a multi-UAV system, following similar goals as in [120].

Leveraging the last idea regarding behavior-based systems, a group of UAVs using relative positioning systems could potentially allow modeling of group behaviors seen in nature, such as the phototaxis behavior among insects for example [116]. These systems can allow a better understanding of these organisms, which could result in the development of new efficient bio-inspired robotic controllers.

Last but not least, a growing interest in employing swarms of UAVs for art and entertainment applications have been observed over the recent years [99]. Following on this line, swarm of UAVs using the positioning systems developed in this work could potentially be used to obtain distributed algorithms needed for displaying large objects and animations in the air and furthermore controlling the swarm to react, and according to the different lighting displays. Additionally, either the developed camera-based or infrared-based positioning systems could provide a simple way of interaction between human operators and UAVs, allowing inexperienced users to operate and control the UAVs through means of active markers manipulated by hand (emitting either in the infrared or visible light spectrum).

## 8.2 Future directions

While many future directions were suggested in the previous section to reach the potential application propositions, this section presents possible future research efforts that aim at improving specific solutions presented in this thesis.

Firstly, improvements could be made on the hardware of the developed relative positioning systems. Regarding the camera-based system, the proposed sensor model can be further used to better chose the parameters of the camera sensor (mainly the camera FOV and resolution). The results in Section 7.2.1 show that the used FOV of 90° for the camera sensor allows only for the implementation of a very limited set of formations topologies. We believe that the system considerations and camera model presented in this thesis help better capture design choices for such relative positioning system, in particular how to trade off additional FOV while maintaining an acceptable accuracy. Still regarding the camera-based system, a new 3D beacon layout that would allow the detection of the multi-beacon marker from any view point should be investigated. Although the visibility constraints for bidirectional connectivity between UAVs was comfortably met for this sensor (see results in Section 7.1.1.3), relaxing those constraints can also allow additional unidirectional connections between UAVs, and therefore the number of possible formation configurations.

Regarding the infrared-based system, it was observed that its major drawback is still the weight of the system. It is worth saying that the weight of the mechanical structure holding the IR receivers has not been optimized, and substantial improvements are still possible and should be considered for future work. However, it was also discussed that the FOV of the system could be reduced in order to allow a lower system weight. Moreover, recall that the developed localization and calibration algorithms allow each IR receiver to be placed at any pose on the UAV. Additionally, note that the number of IR receivers define the weight of the system (apart from a bias weight related to the minimum support structure required to include the microcontrollers that process the sensor information). Therefore, according to the maximum weight allowed for the system (related to the maximum number of IR receivers) it is still possible to optimize the configuration of IR receivers on the UAV in order to still provide the desired accuracy and FOV. These possible modifications and optimizations suggest that the system weight can still be substantially reduced.

Finally, an interesting line of research would be to build on top of the results shown for the previously developed algorithms to control and steer the formation in the environment. This line of research is relevant since this approach is able to retain the scalability of the multi-UAV system without compromising its adaptability or reactiveness. The results show a group of UAVs moving between different defined waypoints. Although a group of four UAVs was used in simulation, only two UAVs were used in the real experiments. An immediate goal is to increase the number of UAV in the real experiments to at least four and achieve similar coordinated maneuvers in a reliable manner. Additionally if, in order to reduce weight or increase FOV, the relative positioning system performance degrades, it becomes interesting to analyze how this degradation can affect the performance of the formation control algorithm. The same analysis can also be conducted for UAVs with larger actuation noise. In these cases, the sensor FOV constraints should be considered in the formation control algorithms, as discussed in Section 5.2.1.1. This will allow a proper study of possible formation control deadlocks, and additional experiments with different variations of the proposed formation control algorithms (e.g., using a non-zero $K_{\perp}$ gain, as presented in Section 5.2.1.1). Moreover, the formation control algorithm should be tested in a scenario with a larger number of UAVs, including team members that do not directly with each other, in order to test the developed solutions in a complete distributed system. Finally, navigation in more complex environments with multiple obstacles should be considered to test the robustness of the approach. In this scenario, it can also be studied how using predictive control for defining inter-vehicle interactions and formation motion could optimize the behavior of the UAVs, according to their dynamic constraints and the geometric constraints imposed by the formation and the environment.

# Bibliography

[1] *AscTec Hummingbird with AutoPilot User's Manual reference*.

[2] M. Achtelik, Tianguang Zhang, K. Kuhnlenz, and M. Buss. Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors. In *2009 International Conference on Mechatronics and Automation*, pages 2863–2869, Aug 2009.

[3] F. Augugliaro, A.P. Schoellig, and R. D'Andrea. Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In *Proceedings of the IEEE International/RSJ 2012 International Conference on Intelligent Robots and Systems (IROS)*, pages 1917–1922, October 2012.

[4] Hari Balakrishnan, Dorothy Curtis, Erik Demaine, Michel Goraczko, Allen Miu, David Moore, Michael Newman, Bodhi Priyantha, Adam Smith, Ken Steele, et al. The cricket indoor location system. 2005.

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

[6] M. Basiri, F. Schill, D. Floreano, and P. U. Lima. Audio-based localization for swarms of micro air vehicles. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4729–4734, 2014.

[7] M. Basiri, F. Schill, P. Lima, and D. Floreano. On-board relative bearing estimation for teams of drones using sound. *IEEE Robotics and Automation Letters*, 1(2):820–827, July 2016.

[8] M. Basiri, F. Schill, P. U. Lima, and D. Floreano. Robust acoustic source localization of emergency signals from micro air vehicles. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4737–4742, 2012.

[9] R. W. Beard, J. Lawton, and F. Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, Nov 2001.

[10] Alberto Bemporad and Claudio Rocchi. Decentralized hybrid model predictive control of a formation of unmanned aerial vehicles. *IFAC Proceedings Volumes*, 44(1):11900–11906, 2011.

[11] J. Bisson, F. Michaud, and D. Letourneau. Relative positioning of mobile robots using ultrasounds. In *Proceedings of the IEEE International/RSJ 2003 International Conference on Intelligent Robots and Systems (IROS)*, pages 1783–1788, October 2003.

[12] Martin Bossert. *Channel coding for telecommunications*. John Wiley & Sons, Inc., 1999.

[13] S. Bouabdallah, P. Murrieri, and R. Siegwart. Design and control of an indoor micro quadrotor. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4393–4398, April 2004.

[14] Samir Bouabdallah, Pierpaolo Murrieri, and Roland Siegwart. Towards autonomous indoor micro vtol. *Autonomous Robots*, 18(2):171–183, 2005.

[15] A. Breitenmoser, L. Kneip, and R. Siegwart. A monocular vision-based system for 6d relative robot localization. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 79–85, Sept 2011.

[16] A. Canepa, Z. Talebpour, and A. Martinoli. Automatic calibration of ultra wide band tracking systems using a mobile robot: A person localization case-study. In *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, Sept 2017.

[17] J. B. Carruthers and J. M. Kahn. Modeling of nondirected wireless infrared channels. In *Communications, 1996. ICC '96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on*, volume 2, pages 1227–1231 vol.2, Jun 1996.

[18] J. B. Carruthers and J. M. Kahn. Angle diversity for nondirected wireless infrared communication. In *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, volume 3, pages 1665–1670 vol.3, Jun 1998.

[19] A. Censi, J. Strubel, C. Brandli, T. Delbruck, and D. Scaramuzza. Low-latency localization by active led markers tracking using a dynamic vision sensor. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 891–898, Nov 2013.

[20] V. Cichella, I. Kaminer, E. Xargay, V. Dobrokhodov, N. Hovakimyan, A.P. Aguiar, and A.M. Pascoal. A lyapunov-based approach for time-coordinated 3d path-following of multiple quadrotors. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 1776–1781, December 2012.

[21] M. Cocetti, L. Sabattini, C. Secchi, and C. Fantuzzi. Decentralized control strategy for the implementation of cooperative dynamic behaviors in networked systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5902–5907, Nov 2013.

[22] M. Cutler, B. Michini, and J. P. How. Lightweight infrared sensing for relative navigation of quadrotors. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1156–1164, May 2013.

[23] Titterton. D. and J. Weston. *Strapdown Inertial Navigation Technology*. 2005.

[24] J.P. Desai, J. Ostrowski, and V Kumar. Controlling formations of multiple mobile robots. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation*, pages 2864–2869, May 1998.

[25] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distante. A new algorithm for ball recognition using circle hough transform and neural classifier. *Pattern Recognition*, 37(3):393 – 408, 2004.

[26] D. Dumitriu, S. Marques, P.U. Lima, J. C. Bastante, J. Araujo, L.F. Penin, A. Caramagno, and B. Udrea. Optimal guidance and decentralised state estimation applied to a formation flying demonstration mission in gto. *IET Control Theory Applications*, 1(2):532–544, March 2007.

[27] M. Egerestedt. Graph-theoretic methods for multi-agent coordination. In *Workshop on Robotics and Mathematics*, September 2007.

[28] M. Egerstedt and Xiaoming Hu. Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation*, 17(6):947–951, August 2001.

[29] M. Faessler, E. Mueggler, K. Schwabe, and D. Scaramuzza. A monocular pose estimation system based on infrared leds. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 907–913, May 2014.

[30] R. Falconi, S. Gowal, and A. Martinoli. Graph based distributed control of non-holonomic vehicles endowed with local positioning information engaged in escorting missions. In *2010 IEEE International Conference on Robotics and Automation*, pages 3207–3214, May 2010.

[31] R. Falconi, S. Gowal, J. Pugh, and A. Martinoli. Graph-based distributed control for non-holonomic vehicles engaged in a reconfiguration task using local positioning information. In *Proceedings of the 2nd International Conference on Robot Communication and Coordination*, pages 1–6, March 2009.

[32] S. H. Fang, T. N. Lin, and P. C. Lin. Location fingerprinting in a decorrelated space. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):685–691, May 2008.

[33] J.A. Fax and R.M. Murray. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, 49(9):1465–1476, September 2004.

[34] Pedro V. Fazenda and Pedro U. Lima. Non-holonomic robot formations with obstacle compliant geometry. *IFAC Proceedings Volumes*, 40(15):439 – 444, 2007. 6th IFAC Symposium on Intelligent Autonomous Vehicles.

[35] R. Fierro, P. Song, A. Das, and V. Kumar. Cooperative control of robot formations. In Robert Murphey and PanosM. Pardalos, editors, *Cooperative Control and Optimization*, volume 66 of *Applied Optimization*, pages 73–93. Springer US, 2002.

[36] R. Forster. Manchester encoding: opposing definitions resolved. *Engineering Science and Education Journal*, 9(6):278–280, Dec 2000.

[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, Special Issue on 3D Exploration, Mapping, and Surveillance*, 31:1504–1525, 2012.

[38] S. García, M. E. López, R. Barea, L. M. Bergasa, A. Gómez, and E. J. Molinos. Indoor slam for micro aerial vehicles control using monocular camera and sensor fusion. In *2016 International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 205–210, May 2016.

[39] S. Gezici, Zhi Tian, G.B. Giannakis, Hisashi Kobayashi, A.F. Molisch, H.V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, July 2005.

[40] Saeed Shiry Ghidary, Takahiro Tani, Toshi Takamori, and Motofumi Hattori. A new home robot positioning system (hrps) using ir switched multi ultrasonic sensors. In *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, volume 4, pages 737–741, 1999.

[41] D. Goldin and J. Raisch. Controllability of second order leader-follower systems. In *IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pages 223–238, 2010.

[42] S. Gowal, R. Falconi, and A. Martinoli. Local graph-based distributed control for safe highway platooning. In *Proceedings of the IEEE International/RSJ 2010 International Conference on Intelligent Robots and Systems (IROS)*, pages 6070–6076, October 2010.

[43] S. Gowal and A. Martinoli. Real-time optimization of trajectories that guarantee the rendezvous of mobile robots. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3518–3525, Oct 2012.

[44] S. Gowal, A. Prorok, and A. Martinoli. Two-phase online calibration for infrared-based inter-robot positioning modules. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3313–3319, Sept 2011.

[45] J. Graefenstein, A. Albert, and P. Biber. Radiation pattern correlation for mobile robot localization in low power wireless networks. In *2009 IEEE International Conference on Robotics and Automation*, pages 3545–3550, May 2009.

[46] Kexin Guo, Zhirong Qiu, Wei Meng, Lihua Xie, and Rodney Teo. Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments. *International Journal of Micro Air Vehicles*, 9(3):169–186, 2017.

[47] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, April 1950.

[48] David Hoeller, Anton Ledergerber, Michael Hamer, and Raffaello D'Andrea. Augmenting ultra-wideband localization with computer vision for accurate flight. *IFAC-PapersOnLine*, 50(1):12734–12740, 2017.

[49] G. Hoffmann, S. Waslander, and C. Tomlin. Quadrotor helicopter trajectory tracking control. In *AIAA Guidance, Navigation and Control Conference*, August 2008.

[50] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys. An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications. In *2013 IEEE International Conference on Robotics and Automation*, pages 1736–1741, May 2013.

[51] Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an rgb-d camera. In *Robotics Research: The 15th International Symposium ISRR*, pages 235–252. Springer, 2017.

[52] Haomiao Huang, G.M. Hoffmann, S.L. Waslander, and C.J. Tomlin. Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3277–3282, May 2009.

[53] Donghwa Jeong and KIJU Lee. Directional rss-based localization of multi-robot applications. In *Proceedings of WSEAS conf. on signal processing, robotics, and automation*, 2013.

[54] Meng Ji and M. Egerstedt. Connectedness preserving distributed coordination control over dynamic graphs. In *Proceedings of the 2005 American Control Conference*, pages 93–98, June 2005.

[55] Meng Ji and M. Egerstedt. Information flow and cooperative control of vehicle formations. *IEEE Transactions on Robotics*, 23(4):693–703, August 2007.

[56] J. M. Kahn and J. R. Barry. Wireless infrared communications. *Proceedings of the IEEE*, 85(2):265–298, Feb 1997.

[57] Juho Kannala and Sami S Brandt. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 28(8):1335–1340, 2006.

[58] Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, 1993.

[59] Alonzo Kelly. Precision dilution in triangulation based mobile robot position estimation. In *Intelligent Autonomous Systems*, volume 8, pages 1046–1053, 2003.

[60] I. Kelly and A. Martinoli. A scalable, on-board localisation and communication system for indoor multi-robot experiments. *Sensor Review*, 24(2):167 – 180, August 2004.

[61] H.K. Khalil. *Nonlinear systems*. Prentice Hall PTR, 2002.

[62] L. Kneip, D. Scaramuzza, and R. Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *CVPR 2011*, pages 2969–2976, June 2011.

[63] A. Ledergerber and R. D'Andrea. Ultra-wideband range measurement model with gaussian processes. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1929–1934, Aug 2017.

[64] Taeyoung Lee, M. Leoky, and N.H. McClamroch. Geometric tracking control of a quadrotor uav on se(3). In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5420–5425, December 2010.

[65] N.E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proceedings of the 40th International Conference on Decision and Control*, pages 2968 – 2973, December 2001.

[66] H. Li, L. Almeida, Z. Wang, and Y. Sun. Relative positions within small teams of mobile units. In *Mobile Ad-Hoc and Sensor Networks*, volume 4864 of *Lecture Notes in Computer Science*, pages 657–671. 2007.

[67] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128x128 120 db 15 us latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuit*, 43(2), 2008.

[68] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, November 2007.

[69] D. Mellinger and V. Kumar. Minimum snap trajectory generation and control for quadrotors. In *Proceedings of 2011 International Conference on Robotics and Automation (ICRA)*, pages 2520–2525, May 2011.

[70] 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.

[71] Nathan Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The grasp multiple micro-uav testbed. *Robotics Automation Magazine, IEEE*, 17(3):56–65, August 2010.

[72] Eduardo Montijano, Eric Cristofalo, Dingjiang Zhou, Mac Schwager, and Carlos Sagüés. Vision-based distributed formation control without an external positioning system. *IEEE Transactions on Robotics*, 32(2):339–351, 2016.

[73] Iñaki Navarro and Fernando Matía. A framework for the collective movement of mobile robots based on distributed decisions. *Robotics and Autonomous Systems*, 59(10):685–697, 2011.

[74] Simon Nolet. The spheres navigation system: from early development to on-orbit testing. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, page 6354, 2007.

[75] P. Ogren, M. Egerstedt, and Xiaoming Hu. A control lyapunov function approach to multiagent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, October 2002.

[76] P. Ogren, E. Fiorelli, and N.E. Leonard. Cooperative control of mobile sensor networks:adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, 49(8):1292–1302, August 2004.

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

[78] R. Olfati-Saber and R. M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 3, pages 2965–2971 vol.3, 2002.

[79] Luis Oliveira and Luis Almeida. *RoboCup 2014: Robot World Cup XVIII*, chapter RF-based Relative Position Estimation in Mobile Ad-Hoc Networks with Confidence Regions, pages 383–394. 2015.

[80] D. Panagou and V. Kumar. Maintaining visibility for leader-follower formations in obstacle environments. In *IEEE International Conference on Robotics and Automation*, pages 1811–1816, 2012.

[81] D. Panagou and K.J. Kyriakopoulos. Cooperative formation control of underactuated marine vehicles for target surveillance under sensing and communication constraints. In *IEEE International Conference on Robotics and Automation*, 2013.

[82] A. Pant, P. Seiler, and K. Hedrick. Mesh stability of look-ahead interconnected systems. *IEEE Transactions on Automatic Control*, 47(2):403–407, Feb 2002.

[83] G. Pereira, Aveek K. Das, V. Kumar, and M. Campos. Formation control with configuration space constraints. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2755 – 2760, 2003.

[84] Amanda Prorok and Alcherio Martinoli. Accurate indoor localization with ultra-wideband using spatial models and collaboration. *The International Journal of Robotics Research*, 33(4):547–568, 2014.

[85] J. Pugh and A. Martinoli. Relative localization and communication module for small-scale multi-robot systems. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 188–193, May 2006.

[86] J. Pugh, X. Raemy, C. Favre, R. Falconi, and A. Martinoli. A fast onboard relative positioning module for multirobot systems. *IEEE/ASME Transactions on Mechatronics*, 14(2):151–162, April 2009.

[87] W. Ren and E. Atkins. Second-order consensus protocols in multiple vehicle systems with local interactions. In *Guidance, Navigation, and Control Conference*, 2005.

[88] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4):25–34, 1987.

[89] R. Ritz, Muller M.W., M. Hehn, and R. D'Andrea. Cooperative quadrocopter ball throwing and catching. In *Proceedings of the IEEE International/RSJ 2012 International Conference on Intelligent Robots and Systems (IROS)*, pages 4972–4978, October 2012.

[90] F. Rivard, J. Bisson, F. Michaud, and D. Letourneau. Ultrasonic relative positioning for multi-robot systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 323–328, May 2008.

[91] J. F. Roberts, T. S. Stirling, J. C. Zufferey, and D. Floreano. 2.5d infrared range and bearing system for collective robotics. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3659–3664, Oct 2009.

[92] J.F. Roberts, T. Stirling, J. C. Zufferey, and D. Floreano. 3-d relative positioning sensor for indoor flying robots. *Autonomous Robots*, 33(1-2):5–20, August 2012.

[93] S. Roelofsen, D. Gillet, and A. Martinoli. Reciprocal collision avoidance for quadrotors using on-board visual detection. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4810–4817, Sept 2015.

[94] Salmah, Sutrisno, E. Joelianto, A. Budiyono, I. E. Wijayanti, and N. Y. Megawati. Model predictive control for obstacle avoidance as hybrid systems of small scale helicopter. In *2013 3rd International Conference on Instrumentation Control and Automation (ICA)*, pages 127–132, Aug 2013.

[95] Angel Santamaria-Navarro, Joan Sola, and Juan Andrade-Cetto. High-frequency mav state estimation using low-cost inertial and optical flow measurement units. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1864–1871, 2015.

[96] Martin Saska, Tomas Baca, Justin Thomas, Jan Chudoba, Libor Preucil, Tomas Krajnik, Jan Faigl, Giuseppe Loianno, and Vijay Kumar. System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization. *Autonomous Robots*, 41(4):919–944, 2017.

[97] D. Scaramuzza, M. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M.W. Achtelik, M. Chl, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, H. Gim, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier. Vision-controlled micro flying robots: from system design to autonomous navigation and mapping in gps-denied environments. *IEEE Robotics & Automation Magazine*, 21(3):26–40, 2014.

[98] F. Schiano, A. Franchi, D. Zelazo, and P. R. Giordano. A rigidity-based decentralized bearing formation controller for groups of quadrotor uavs. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5099–5106, Oct 2016.

[99] M. Schoch, J. Alonso-Mora, R. Siegwart, and P. Beardsley. Viewpoint and trajectory optimization for animation display with aerial vehicles. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4711–4716, May 2014.

[100] Shaojie Shen, Nathan Michael, and V. Kumar. Autonomous multi-floor indoor navigation with a computationally constrained mav. In *Proceedings of 2011 International Conference on Robotics and Automation (ICRA)*, pages 20–25, May 2011.

[101] Shaojie Shen, Yash Mulgaonkar, Nathan Michael, and Vijay Kumar. Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. *Robotics: Science and Systems*, 1, 2013.

[102] Henrique Silva. *Controlo de Forma cão de Veículos Aéreos não Tripulados*. Instituto Superior Técnico, 2012.

[103] Jorge M Soares, A Pedro Aguiar, António M Pascoal, and Alcherio Martinoli. A graph-based formation algorithm for odor plume tracing. In *Distributed Autonomous Robotic Systems*, volume 112, pages 255–269. Springer, 2016.

[104] Joan Sola. Quaternion kinematics for the error-state Kalman filter, 2017.

[105] G. Stacey, R. Mahony, and P. Corke. A bondgraph approach to formation control using relative state measurements. In *European Control Conference*, pages 1262–1267, 2013.

[106] Maja Stella, Mladen Russo, and Dinko Begusic. RF localization in indoor environment. *Radioengineering*, 21(2):557–567, 2012.

[107] D. Swaroop and J. K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–357, March 1996.

[108] Nikolas Trawny and Stergios I Roumeliotis. Indirect Kalman filter for 3d attitude estimation. *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep*, 2:2005, 2005.

[109] R. Tron, J. Thomas, G. Loianno, J. Polin, V. Kumar, , and K. Daniilidis. Vision-based formation control of aerial vehicles. In *Robotics: Science and Systems*, 2014.

[110] M. Turpin, N. Michael, and V. Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1-2):143–156, August 2012.

[111] BE Tweddle, J McClellan, G Vulikh, J Francis, and DW Miller. Relative vision based navigation and control for the mars sample return mission: Capturing the orbiting sample.

[112] Maja Varga, Meysam Basiri, Gregoire Heitz, and Dario Floreano. Distributed formation control of fixed wing micro aerial vehicles for area coverage. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 669–674, 2015.

[113] G. Vásárhelyi, Cs. Virágh, G. Somorjai, N. Tarcai, T. Szorényi, T. Nepusz, and T. Vicsek. Outdoor flocking and formation flight with autonomous aerial robots. In *International Conference on Intelligent Robots and Systems*, pages 3866–3873, 2014.

[114] C. Verginis, C. Bechlioulis, D. Dimarogonas, and K. Kyriakopoulos. Decentralized 2-d control of vehicular platoons under limited visual feedback. In *IEEE International Conference on Intelligent Robots and Systems*, 2015.

[115] Xiaorui Xi and Eyad H Abed. Formation control with virtual leaders and reduced communications. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 1854–1860. IEEE, 2005.

[116] JING Xiang-Feng and LEI Chao-Liang. Advances in research on phototaxis of insects and the mechanism [j]. *Entomological Knowledge*, 3:002, 2004.

[117] Mahboubi Z., Kolter Z., Wang T., Bower G., and Ng A. Y. Camera based localization for autonomous uav formation flight. American Institute of Aeronautics and Astronautics, 2011.

[118] R. Zhang and H.H.T. Liu. Vision-based relative altitude estimation of small unmanned aerial vehicles in target localization. In *Proceedings of the American Control Conference (ACC)*, pages 4622–4627, June 2011.

[119] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 666–673, 1999.

[120] Hu Zhi-wei, Liang Jia-hong, Chen Ling, and Wu Bing. A hierarchical architecture for formation control of multi-uav. *Procedia Engineering*, 29:3846–3851, 2012.

# Appendices

# Appendix A

# Kalman Filter Implementation

In this thesis, the Kalman Filter framework [58] is used to estimate the vehicle self position in the environment and the relative position between vehicles. The quantities to be estimated are stacked into a state-vector $\mathbf{x}$. Given the uncertainty of the vehicle actuation and measurements provided by its sensors, $\mathbf{x}$ is considered to be a random variable. Additionally, given the discrete nature of the digital processing of computers, $\mathbf{x}$ is considered to evolve in time in discrete time steps of period $\Delta t$. The random variable at each time step $k$, $\mathbf{x}(k)$, evolves into the next time step according to a motion model. This model can be written in its general form as:

$$\mathbf{x}(k+1) = f_k\left(\mathbf{x}(k), \mathbf{u}(k), \xi(k)\right),$$

where $f_k$ is the function describing the propagation of $\mathbf{x}(k)$ provided an input $\mathbf{u}(k)$, and $\xi(k)$ is the propagation noise, usually associated to uncertainties in $\mathbf{u}(k)$. At specific time steps, $\mathbf{x}(k)$ can be observed using sensor measurements $\mathbf{z}_o(k)$ . These measurements are related with $\mathbf{x}(k)$ according to a measurement model. This model can be written in its general form as:

$$\mathbf{z}_o(k) = h_k\left(\mathbf{x}(k), \eta(k)\right),$$

where $h_k$ is the function relating $\mathbf{z}_o(k)$ with $\mathbf{x}(k)$, and $\eta(k)$ is the measurement noise. Note that $f_k$ and $h_k$ can change at each time step.

A Kalman Filter estimates $\mathbf{x}(k)$ at each time step using the previous motion and observation models. In this filter, $\mathbf{x}(k)$ is assumed to be a Gaussian distribution $\mathbf{x}(k) \sim \mathcal{N}\left(\hat{\mathbf{x}}(k), \mathbf{P}(k)\right)$, with $\hat{\mathbf{x}}(k)$ and $\mathbf{P}(k)$ being respectively, the mean and covariance of the distribution. The filter takes $\hat{\mathbf{x}}(k)$ as the estimate for $\mathbf{x}(k)$. The filter also considers the propagation noise $\xi(k)$ and the measurement noise $\eta(k)$ to be Gaussian distributions, with zero mean and a covariance of $\mathbf{Q}_\xi(k)$ and $\mathbf{R}_\eta(k)$, respectively. When a Jacobian can be computed for functions $f_k$ and $h_k$, the Extended Kalman Filter can be used to compute $\hat{\mathbf{x}}(k)$, using the motion and measurement models. This filter computes $\hat{\mathbf{x}}(k)$ in a two-step process. The first step, named as prediction, computes the posterior distribution $\mathbf{x}(k+1)$, provided a function $f_k$, a prior distribution

$\mathbf{x}(k)$, an input $\mathbf{u}(k)$, and a noise distribution $\xi(k) \sim \mathscr{N}\left(\mathbf{0}, \mathbf{Q}_{\xi}(k)\right)$, as follows:

$$\hat{\mathbf{x}}(k+1) = f_k\left(\hat{\mathbf{x}}(k), \mathbf{u}(k), \mathbf{0}\right) \qquad \mathbf{P}(k+1) = \mathbf{A}_k \mathbf{P}(k) \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{\xi}(k) \mathbf{W}_k^T,$$

$$\mathbf{A}_k = \frac{\partial f_k}{\partial \mathbf{x}} \qquad \mathbf{W}_k = \frac{\partial f_k}{\partial \xi},$$

where $\frac{\partial f_k}{\partial \mathbf{x}}$ and $\frac{\partial f_k}{\partial \xi}$ are the Jacobian of $f_k$ computed for $\mathbf{x}$ and $\xi$, respectively. The second step, named as update, computes the posterior distribution $\mathbf{x}(k)$, provided a function $h_k$, a prior distribution $\mathbf{x}^{-}(k)$, a measurement $\mathbf{z}_o(k)$, and a noise distribution $\eta(k) \sim \mathscr{N}\left(\mathbf{0}, \mathbf{R}_{\eta}(k)\right)$, as follows:

$$\mathbf{K} = \mathbf{P}(k) \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}(k) \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_{\eta}(k) \mathbf{V}_k^T\right),$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^{-}(k) + \mathbf{K}\left(\mathbf{z}_o(k) - h_k(\hat{\mathbf{x}}^{-}(k)\mathbf{0})\right) \qquad \mathbf{P}(k) = \left(\mathbf{I} - \mathbf{K}\mathbf{H}_k\right) \mathbf{P}^{-}(k),$$

$$\mathbf{H}_k = \frac{\partial h_k}{\partial \mathbf{x}} \qquad \mathbf{V}_k = \frac{\partial h_k}{\partial \eta},$$

where $\frac{\partial h_k}{\partial \mathbf{x}}$ and $\frac{\partial h_k}{\partial \eta}$ are the Jacobian of $h_k$ computed for $\mathbf{x}$ and $\eta$, respectively. When the functions $f_k$ and $h_k$ are linear functions of $\mathbf{x}$, $\xi$ and $\eta$, this filter simplifies to the standard Kalman Filter, which is an optimal linear filter for uncorrelated (white) noise. When the functions $f_k$ and $h_k$ are non-linear (which is the case of some models of this thesis), this filter only provides approximate estimate of the distribution of $\mathbf{x}$. The previous filter is used to acquire the necessary estimations throughout this thesis.

# Appendix B

# Infrared-based Communication Algorithm

In the proposed infrared-based relative positioning system presented in Section 5.1.2, each IR beacon of a marker is able to emit pulses at specific time slots according to a TDMA algorithm. As described in Section 5.1.2.1, in its emission slot, the IR beacon can either emit or not. This allows the creation of digital pulse sequences for each IR beacon, on top of which a digital communication algorithm can be implemented. This algorithm enables the transmission of the ID of each individual IR beacon and an additional bitstream in order to allow the communication between sensors. The beacon ID is directly related to an unique digital pulse sequence that the IR beacon transmits in its assigned emission slot. The encoding of the beacon ID and the digital communication data to a digital pulse sequence to be emitted by each beacon is performed by the *Communication Stack* module illustrated in Fig. 5.11. The encoding algorithm will now be described in detail.

To define the digital pulse sequence, each IR beacon is associated to a binary code with $N_{code}$ bits. This code is set as the beacon ID and it is unique for each IR beacon. The digital pulse sequence has the same length as the binary code, and each position of the sequence is defined by the respective bit position in the code. If bit $l$ of the binary code is '1', the IR beacon will be scheduled to emit on its the emission slot every time the position $l$ of the digital pulse sequence is to be transmitted, as depicted in Fig. B.1a. The digital pulse sequence is transmitted from the lowest to the highest significant bit (right to left) as shown in the figure. Considering the previous binary code length, there are $2^{N_{code}}$ codes that can be used to define beacon IDs. However, not all codes can be used.

Firstly, binary codes that have a small number of '1's will correspond to digital pulse sequences with a small number of pulses. This can be a problem for the inter-beacon synchronization. Take for example the beacon ID 0, which translates to a binary code of '0000000000' for $N_{code} = 10$. The previous described encoding applied to this code will generate a digital pulse sequence with no pulses. In this case, it is not possible to synchronize the respective IR beacon with the others, since no information is given by the digital pulse sequence. All other codes will generate digital pulse sequences with at least one pulse. However, the larger the number of emitted pulses the better the achieved inter-beacon synchronization. In this work, the previous binary code is furthered encoded using the Manchester algorithm, presented in [36], in order to make all binary codes with the same number of '1's. The Manchester algorithm

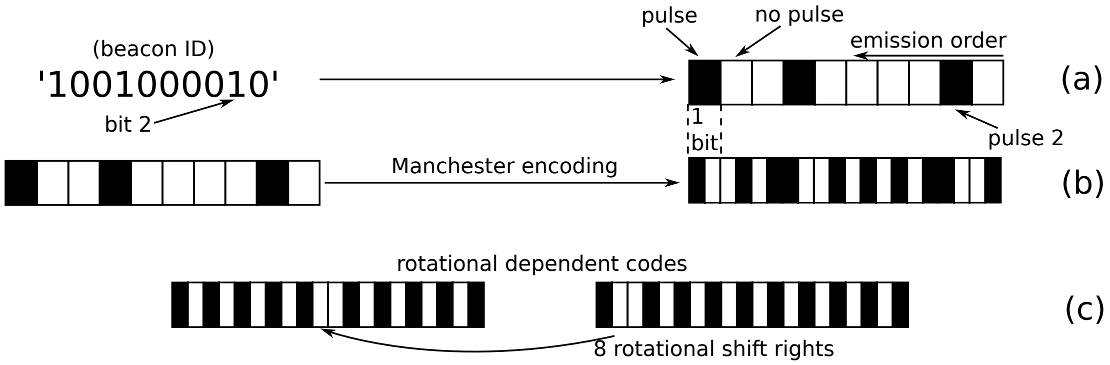


Figure B.1: IR-based positioning system beacon ID encoding algorithm. (a) How the beacon ID translates into a digital pulse sequence. (b) Manchester encoding of the beacon ID in order to ensure the same number of pulses for all the generated digital pulse sequences. (c) Two codes that are rotational dependent might be misplaced misplaced with each other at the receiver side. Here, a beacon ID code size of $N_{code} = 10$ is used.

extends the base binary code to a length of $N_{code_{Mch}} = 2N_{code}$. Each bit of the base binary code is encoded into two bits of the respective Manchester code. A digital '1' of the base binary code is encoded as a '10' pair, and a digital '0' is encoded as a '01' pair (or vice-versa), as shown in Fig. B.1b.

Secondly, codes that are rotational dependent after the previous Manchester encoding, such as the ones presented in Fig. B.1c, can not be used, since the time frame is differently measured for each IR sensor. For example, it might be that sensor $i$ measures time eight emission slots in advance from sensor $j$ (or $\tau^i = \tau^j + 8T$, where $T$ is the emission slot period discussed in Section 5.1.2.1). In this situation, the pulse from an IR beacon of marker $k$ occurring at a certain time instance $\tau^*$ will translate to bit $l$ for sensor $i$, and to bit $l-8$ for sensor $j$. This means that the digital pulse sequence measured by sensor $j$ is the same as the one measured by sensor $i$, but shifted eight times to the right. So, it is clear that if the two codes presented in Fig. B.1c are used as two different beacon IDs, they might be misplaced with each other given the difference in the time frame between sensors. Therefore, only binary codes that have at least one different bit at any possible rotation between each other are used.

To detect the previous codes at the receiver side, a list of digital pulse sequences, corresponding to all the Manchester encoding of the binary codes of size $N_{code}$ defining a beacon ID, is created at each sensor. When $N_{code}$ pulses are read from a respective beacon time slot, the resulting digital pulse sequence is matched with the correct element of the code list. The received digital pulse sequence is rotated $N_{code}$ times in order to compensate for the previous described code rotational misalignments. However, digital pulse sequence detection errors might occur due to channel noise. This noise can be generated from other devices (e.g. external MCS operating at the same IR frequencies), occlusion from the propellers, as discussed in Section 5.1.2.5, or simply by the sensor not being able to detect the pulses because it is too far from the IR beacon. This can make the wrongly detected digital pulse sequence not have a match in the previously formed code list, or to have a match with the wrong element of the code list. The latter problem causes a wrong detection of the beacon ID, ultimately leading to localization errors since beacon IDs are directly associated to the IR beacon position in the marker frame, as discussed in

Section 5.1.2.1.

Therefore, in order to have some robustness to possible digital pulse sequence detection errors at the receiver side, the concept of hamming distance is used to select the binary codes that can be used for beacon ID assignment. As described in [47], the humming distance between two binary codes is defined by the number of bits that are different between those two codes. For example, the previously referred codes, that have one different bit at any possible rotation between each other, have a humming distance of one. By considering all the binary codes in these conditions, the previous discussed ID detection problems can occur with just one error on any position of the digital pulse sequence. However, it is possible to consider just binary codes with a humming distance greater than equal to $d_{humming}$ ($d_{humming}$ different bits at any possible rotation). Note that every error on a position of the digital pulse sequence increases the hamming distance between the emitted and the detected digital pulse sequence by one. The hamming distance can be computed between the detected binary code and each element of the code list. If all elements in the code list have a humming distance greater than or equal to $d_{humming}$, humming distances between the detected binary code and an element of the code list $e < d_{humming}/2$ can mean that: a number of $e$ errors have occurred during the detection of the digital pulse sequence; a number of errors larger than $d_{humming}/2$ have occurred during the detection of the digital pulse sequence, which led the detected binary code closer to a beacon ID that does not correspond to the emitting IR beacon. This means that, with this approach, it is still possible to match the detected binary codes to the right beacon ID up to a maximum of $d_{humming}/2$ errors in the digital pulse sequence detection.

Although the previous methods gives robustness for small channel noise, larger channel noise, such as external devices or the UAV propellers, can still generate a mismatch on the chosen beacon ID. In order to minimize the occurrences of beacon ID detection mismatches, the concept of fuzzy logic is used. A fuzzy logic bit is defined as a real number between 0 and 1, instead as an integer '0' or '1'. This bit is able to describe the probability of a bit of a digital pulse sequence being a '1'. Every time the bit $l$ of the binary code is received, the respected bit $l$ of the fuzzy logic code is updated as follows:

$$ID^i_{fuzzy,jml} = ID^i_{fuzzy,jml} w_{fuzzy} + (1 - w_{fuzzy}) ID^i_{jml}, \tag{B.1}$$

where $ID^i_{fuzzy,jml}$ is the bit $l$ of the fuzzy logic code associated to marker $j$'s IR beacon $m$ detected at sensor $i$, and $ID^i_{jml}$ is the bit $l$ of the binary code associated to marker $j$'s IR beacon $m$ currently detected at sensor $i$. The value of $ID^i_{jml}$ can be either '1' or '0'. The parameter $w_{fuzzy}$ defines how much noise can the system tolerate. Each bit of the fuzzy logic code is detected as a '1', if the respective fuzzy logic bit is greater than a threshold. Each bit of the fuzzy logic code is detected as a '0', if the respective fuzzy logic bit is lower than a threshold. The fuzzy logic code is only considered to be a valid deterministic code when all the bits are detected as either a '1' or a '0'. The fuzzy logic code is able to discard errors in the detected binary code that are not persistent in the same binary code bit. This allows the correct detection of the beacon IDs, even when the channel noise produces a number of binary code errors greater than $d_{humming}/2$.

In this work, $N_{code} = 12$ and $d_{humming} = 5$. The number of beacon IDs that can be considered in the previous conditions is 32. The binary codes can be detected without problems up to 2 errors on the detected digital sequence. However, it is possible to change both parameters in order to have higher robustness to channel noise or larger number of beacon IDs. The beacon IDs are converted to binary codes encoded by the previously described Manchester algorithm. The resulting binary code is then sent as a digital pulse sequence by the respective IR beacon. At the receiver side, the parameter $w_{fuzzy}$ is set to 0.5 in this work, but larger values can be considered. Larger values will slow down the convergence of the fuzzy logic bits to a detected '1' or '0'. Therefore, by increasing $w_{fuzzy}$, the beacon ID detection becomes more robust to channel noise without having to increase $N_{code}$ and $d_{humming}$. However, too large $w_{fuzzy}$ values will create substantial delays in obtaining a valid deterministic code, since it might take a substantial amount of time for each bit to be detected as a '1' or '0', which can lead to delays in obtaining the inter-vehicle localization, as also briefly discussed in Section 5.1.1.

In the previous fuzzy algorithm, after a valid deterministic code is detected, every detected bit $l$ that does not match the respective bit of the valid deterministic code is considered to be information that is independent from the digital pulse sequence identifying the beacon ID. This information can be generated from the previously described channel noise, or it can be used to encode data in order to allow communication between sensors. Note that the most common channel noise that cannot be eliminated comes from propeller occlusions or weak pulse signals incoming from far away IR beacons. This channel noise transforms the '1's of the digital pulse sequence into '0's. The '0's of the digital pulse sequence remain the same. Therefore, in order to not to increase the noise affecting the '1's of the digital pulse sequence, the communication data is only transmitted in the '0' positions of the digital pulse sequence. Note that all beacon IDs are encoded into a digital pulse sequence with the same number of '0's. Therefore, the maximum communication rate with this approach is the same for all beacon IDs. The previous communication encoding into the digital pulse sequence is depicted in Fig. B.2a.

Each bit of a stream of data bits (or a bitstream) is encoded into the digital pulse sequence at sequential '0' positions of the respective beacon ID binary code. A '0' is encoded as a '0' meaning that the digital pulse sequence remains the same. A '1' is encoded as a '1' meaning the digital pulse sequence will change in the respective position. In order to add robustness to channel noise, a repetition-code algorithm is used [12]. This algorithm consists of sequentially transmitting $N_{rep}$ copies of the bit, each copy called a sub-bit, in order to add redundancy, as described in Fig. B.2a. The bit detected at the receiver side is selected as the integer '0' or '1' that had the most detected repetitions in a sequence of $N_{rep}$ transmitted sub-bits. However, if this communication scheme has to transmit continuous sequence of '1's, all '0's of the digital pulse sequence will be used. If the sequence of '1's is sufficiently long (period dependent on $w_{fuzzy}$), the receiver will start detecting '1's in positions where the binary code is supposed to be '0'. This generates problems for the beacon ID detection, as previously described.

However, this is solved by dividing the communication into pairs of digital pulse sequence periods ($N_{code_{Mch}}$ emission slots), as shown in Fig. B.2a. In this pair, the first period (or communication period) uses all the '0's in the digital pulse sequence for communication. In the second period (or recover period), only the '0's from the digital pulse sequence are transmitted (no communication).If a '1' is sent

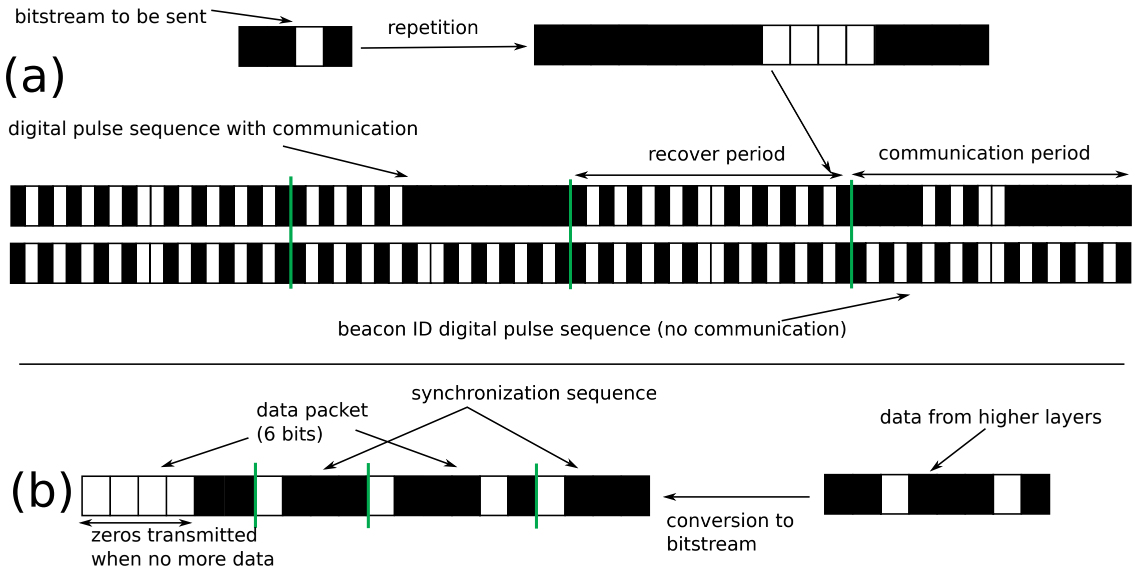


Figure B.2: IR-based positioning system beacon communication data transmission. (a) The bitstream is encoded to the digital pulse sequence with a repetition-code algorithm. The length of the communication and recover periods is the same as the digital pulse sequence length ($N_{code_{Mch}}$ emission slots). In this example, a beacon ID code size of $N_{code} = 10$ is used. (b) bitstream structure of a communication package (note the synchronization sequence). Data that does not fit in a single packet is relayed to another packet, after a synchronization sequence. When no more data is available, the last remaining positions of the current packet is filled with '0's. In this example, the packet size $N_{packet}$ is set to 6.

in a certain position $l$ of the digital pulse sequence during the communication period, the recover period will send a '0' in the same position. According to Eq. (B.1), the respective fuzzy logic bit value at the receiver side will remain unaltered after the two previous digital pulse sequence periods.

Finally, the data is provided by the higher program layers, as depicted in Fig. 5.11. The communication algorithm divides the data into packets of $N_{packet}$ bits. If no data is provided, '0's will be transmitted. At the beginning of each packet, a synchronization sequence of '1110' is transmitted into the channel. This helps the receiver side to synchronize with the beginning of the packet. The resulting bitstream is transmitted through the IR channel using the previous transmission algorithm. Note that each data bit (including the synchronization sequence) is transmitted with $N_{rep}$ sub-bits. The structure of the communication packet is depicted in presented in Fig. B.2b. Once the packet is received at the receiver side, the data is relayed to the programs at the higher layers, as shown in Fig. 5.11. These programs are responsible to interpret the data. The system does not deal with data losses or errors other than the ones that were covered in the previous discussion.

In this work $N_{rep} = 4$ which enables the communication system to correct up to one sub-bit detection error. Additionally $N_{packet} = 6$. This value, together with the previous transmission algorithm, allows to predict the maximum communication rate of the system.

The previous communication algorithm assumes no inter-beacon interference. In Appendix C, the emission synchronization algorithm that deals with this problem is presented.

# Appendix C

# Infrared-based Emission Synchronization Algorithm

In order to avoid inter-beacon interference on the developed infrared-based relative positioning system, a synchronization algorithm based on a TDMA algorithm was implemented (similarly to [91, 92]), as briefly described in Section 5.1.2.1. In this algorithm, each IR beacon $b_{jm}$ emits in a specific time slot $\tau_{jm}$. A TDMA period is defined between two consecutive emission slots of the same IR beacon. For each emission slot, a second downtime slot is added in order to avoid emission overlap of sequential IR beacons. All IR beacons from a marker have sequential emission slots, and each IR beacon of the marker is given a position in this sequence. This allows the definition of super-slots for each marker, as shown in Fig. 5.10. Marker $j$'s super-slot starts with the emission slot of its first IR beacon $\tau_j = \tau_{j1}$. The time period of super-slot $j$ can be computed from the number of IR beacons of marker $j$. The synchronization algorithm guarantees that markers $i$ and $j$ super-slots are always spaced by $\tau_{ij,d}$. This value is selected a priori for all markers. The synchronization algorithm is executed in the *Emission synchronization* module, illustrated in Fig. 5.11. Note that this algorithm is entirely implemented in the IR emission channel without using additional transmitting devices.

The algorithm starts at each time step $k$, where the *Beacon Detection* module (also illustrated in Fig. 5.11) collects beacon ID and emission slot information ($\tau^i_{jm}(k)$) of each IR beacon $b_{jm}$ within range of the sensor. From the measured $\tau^i_{jm}(k)$, the current initial emission slot of marker $j$'s super-slot ($\tau^i_j(k)$) can be computed. Recall that $\tau^i_j(k)$ corresponds to the emission slot of marker $j$'s first IR beacon ($\tau^i_{j1}(k)$). Also, Recall that all IR beacons occupy two emission slots and IR beacons from the same marker emit in consecutive slots. With that knowledge one can compute $\tau^i_j(k)$ as follows:

$$\tau^i_j(k) = \tau^i_{jm}(k) - 2(m-1).$$

Note that the knowledge of $m$, which relates to the beacon ID, is needed for the previous expression. Recall that the beacon ID was already detected in the *Beacon Detection* module. Finally, note that only one IR beacon of each marker is needed to compute $\tau^i_j(k)$. In case $\tau^i_{jm}(k)$ is collected for more than one

IR beacon of marker $j$, $\tau_j^i(k)$ is computed using the previous expression independently for each collected $\tau_{jm}^i(k)$. These values are then averaged together in order to reduce noise.

The algorithm then proceeds by computing the slot difference between $\tau_j^i(k)$ with the current emission slot of marker $i$'s super-slot ($\tau_i^i(k)$), which is controlled by sensor $i$. The slot difference is therefore computed as $\tau_{ij}(k) = \tau_j^i(k) - \tau_i^i(k)$. With that information, the algorithm can adjust $\tau_i^i(k)$ in order to respect the desired emission slot differences for all detected markers, $\tau_{ij,d}$. This is done using a consensus equation as follows:

$$\tau_i^i(k+1) = \tau_i^i(k) + \frac{w}{N}\sum_{j=1}^{N}(\tau_{ij}(k) - \tau_{ij,d}),$$

where $N$ is the number of detected markers and $w$ is an importance weight. In this work $w$ was set to 1, but this weight can be a function of how close the detected marker is (related to the measured RSS) for example. At time step $k+1$, IR beacon $b_{i1}$ will emit in the emission slot $\tau_i^i(k+1)$ ($b_{i2}$ in slot $\tau_i^i(k+1)+2$ and so forth). Finally, the entire procedure of readjusting $\tau_i^i(k+1)$ restarts again with the *Beacon Detection* module collecting new beacon ID and emission slot information. Note that the previous consensus equation is applied on a single dimension and to linear functions. Therefore, all $\tau_{ij}(k)$ values will converge to $\tau_{ij,d}$ if all sensors execute this algorithm (assuming that all sensors are detecting at least one neighboring marker).

One important note is that, although each IR beacon emits in a discrete time slot $\tau_i^i$ with respect to sensor $i$, it might be detected by other sensors $j$ in a time slot that is misaligned with the discrete slots of those sensors, as shown in Fig. C.1a. This can happen since the internal clock of the sensors is not completely synchronized. The slot misalignment problem can cause the IR beacon emissions to be detected in two slots, as shown in the figure. The first (lead) slot is considered to be the detected emission slot. The second (back) slot is considered to be the downtime slot. The slot misalignment problem can also interfere with the measured RSSs, since while the emission becomes misaligned with the detected emission slot its energy gets distributed to the neighboring slots, as shown in the figure. These behaviors were observed by conducting an experiment with an emitting marker and a sensor with their internal clocks slightly unsynchronized. The emission slot information and the measured RSS for one IR beacon detected by an IR receiver are shown in Fig. C.1b. The results show that the detected emission slot of the IR beacon was changing over time with a small constant velocity (defined by the small internal clock asynchrony between the two devices). Additionally, the RSS measured in the lead and back slots of the IR beacon were fluctuating over time, suggesting the previously described energy distribution problem. The detected emission slot can be realigned with the IR beacon emission and the RSS measurement fluctuation seen in Fig. C.1b can be compensated by implementing an algorithm that analyses the lead and back slots together.

Regarding the realignment between the detected emission slot and the IR beacon emission, note that the relationship between the lead and back slot RSS measurements for a misaligned emission can be modeled roughly from the charging and discharging dynamics of the RC circuitry deployed on the IR receivers' detection hardware, presented in Fig. 5.19. The relationship between the lead and back slot

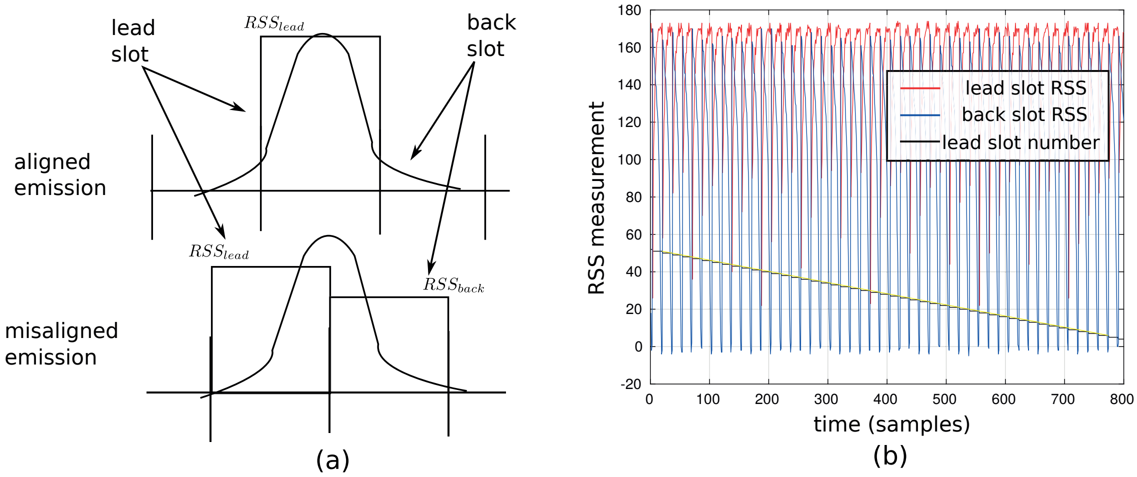


Figure C.1: Illustration of the slot misalignment problem. (a) Depicting an aligned and a misaligned emission with respect to the detected emission slot. Note the energy distribution over the neighboring slots for the misaligned emission situation. (b) Emission slot information and the measured RSS for one IR beacon detected by an IR receiver, for two unsynchronized devices. Note the detected emission slot changing in tame, as well as the RSS fluctuations measured on the lead and back slots.

RSS measurements is computed in this work by the following function:

$$f_p = \frac{RSS_{lead}}{RSS_{lead} + RSS_{back}},$$

where $RSS_{lead}$ and $RSS_{back}$ are respectively the RSS measurements acquired on the lead and back slots (also illustrated in Fig C.1. The value of $f_p$ is between 0 and 1.

This relationship can be clearly observed in the results of Fig. C.1b. From these results, the instant of time where the lead slot changes is recorded. A slot change speed can be computed from the time period between different lead slot changes. Using the previously gathered information, an estimate of the position of the emission on the lead slot can be acquired. For example, if the lead slot was 49 and changed in the current instant of time to 48, it is most likely that the emission position in the lead slot will not be less than 48.9. This position will decrease through time at the rate given by the estimated slot change velocity. The decimal part of the estimated position of the emission on the lead slot indicates how much is the emission misaligned with the detection slots. This misaligned factor is therefore defined between 0 and 1, where 0.5 means maximum misalignment between the emission and the detected emission slot. Note that the value for $f_p$ and the misalignment factor can be computed at each instant of time. With the previous information acquired at all instants of time it is possible to map $f_p$ in function of the misalignment factor. Fig. C.2 presents the results of such mapping for one particular experiment. It is possible to see that $f_p$ is a clear injective function of the misalignment factor. In this work, this function was modeled using a piecewise linear polynomial as described in the figure. This function is considered to be the same to all IR receivers, since they have the same hardware circuitry that is defining this behavior.

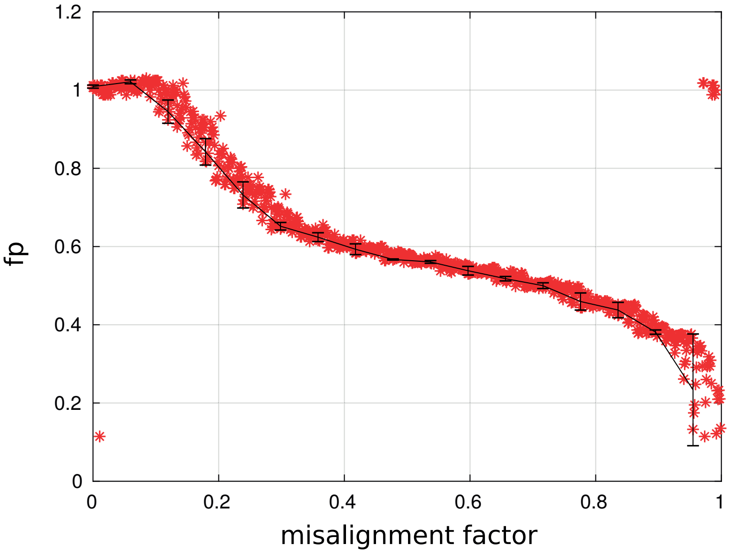


Figure C.2: Mapping between the lead and back slot ratio and the emission misalignment factor. The black continuous line represents the piecewise linear polynomial that was used to model $fp$ in function of the misalignment factor. The uncertainty bars indicate how much the gathered data deviates from the model.

At each instant of time, the measured $RSS_{lead}$ and $RSS_{back}$ are used to estimate the misalignment factor using the modeled $f_p$ function. This misalignment factor is used to control the misalignment between the emission and the detected emission slot. This is done by inflating or contracting the period of the detection slots on the sensor side. This inflating-contracting behavior slightly shifts the emission between the detection slots until the emission exhibits an alignment behavior (high lead RSS and low back RSS measurements). The slot period control is implemented with a proportional and integral controller using the computed misalignment factor. The integral component retains the value of the internal clock asynchrony between the devices. An integral component is considered for each detected device.

Regarding the RSS measurement fluctuations, they can be compensated by mapping the measured $RSS_{lead}$ and $RSS_{back}$ together. Fig. C.3 shows the mapping results conducted at different ranges between the emitting IR beacon and the IR receiver. Note that, for the data gathered at each individual range, the $RSS_{lead}$ near the $x$ axis of the plot have the same value than the $RSS_{back}$ near the $y$ axis. Additionally, these values are the largest compared with all the other values measured in the same range between the IR beacon and IR receiver. In fact, the $x$ and $y$ axes of this plot represent situations where the emission is aligned with the detected emission slots. The captured energy in these situations is at its maximum value, as illustrated in Fig. C.1. This value can be considered to be the true emission energy. When a misalignment occurs, the $(RSS_{lead}, RSS_{back})$ pair moves towards the center of the plot, as illustrated in Fig. C.3. In this situation, their values do not capture anymore the true emission energy as the energy becomes distributed between the lead and back slots (recall Fig. C.1a). However, as it can be observed from the plot, contour lines can be computed to represent the true emission energy during misalignment situations. Using these contour lines, a $(RSS_{lead}, RSS_{back})$ pair can be traced back to the RSS measure-

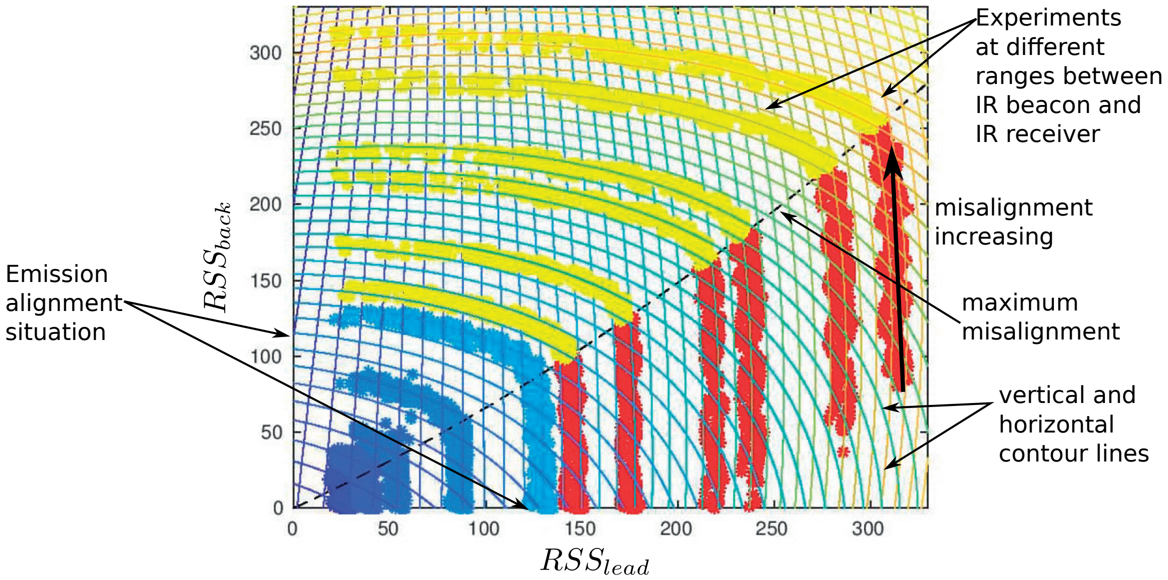


Figure C.3: Mapping between the lead and back slot RSS measurements at different ranges between two unsynchronized devices. Note the situations when the emission is aligned with the detected emission slot, and what happens when misalignment occurs. The vertical and horizontal continuous lines are contour lines that represent the same true emission energy (regardless of an alignment or a misalignment situation).

ment corresponding to the true emission energy. This trace back is implemented in this work through the following polynomial:

$$\begin{aligned} RSS = \quad & a_0 + a_1 RSS_{lead} + a_2 RSS_{back} + a_3 RSS_{lead} RSS_{back} + a_4 RSS_{lead}^2 + \\ & + a_5 RSS_{back}^2 + a_6 RSS_{lead}^2 RSS_{back} + a_7 RSS_{lead} RSS_{back}^2 \end{aligned}$$

where $RSS$ is the traced back RSS measurement. This value will remain roughly constant regardless of misalignments between the emission and the detected emission slots. The parameters of the previous polynomial are calibrated with $(RSS_{lead}, RSS_{back})$ pairs measured during the experiments. Note that different polynomial parameters are used in the two areas presented in Fig. C.3 (separated by the dashed line). This is done given the observed behavior of the $(RSS_{lead}, RSS_{back})$ pairs, which seem to abruptly change near the maximum misalignment situation. The two polynomials define the horizontal and vertical contour lines illustrated in the figure.

# Appendix D

# Ad-hoc Motion Capture System

For the experiments conducted in Section 7.3.3, despite the absence of a MCS in the used flying arena, a method for acquiring the ground truth of the UAV 3D positions was used. This method consisted of synchronizing and fusing height measurements ($z_{m_i}$) provided by the *Self State Estimator* on board the UAVs (see Section 4.2 for details about this estimator) with bearing measurements ($r_{\mathscr{C}_i}$) acquired by an offboard static camera looking downwards to the UAVs, as shown in Figs. D.2a and b. The localization procedure is depicted in Fig. D.1. The height measurements identify the horizontal plane of the absolute frame where the UAV is located. The bearing measurements are intersected with this plane in order to acquire the actual UAV position. The images and the height measurements are gathered during the experiment. However, the ground truth is computed offline.

To compute the bearing measurements ($r_{\mathscr{C}_i}$), the UAV positions on each camera image are first detected. The detection is carried out using a background subtraction method on the grayed-scale images acquired by the camera (this is possible since the camera is static in the environment). The regions in the image that were marked as different from the original background are merged into image blobs. The center of each blob is assumed to be an UAV position in the image. These blobs are then tracked across multiple images. In this work, the *Motion-Based Multiple Object Tracking* toolbox of MATLAB[12] was used to track the blobs. After analyzing all the images, the tracked blobs are filtered in order to remove false positives. Firstly, thresholds on the detected blobs maximum and minimum pixel size are applied in order to remove most of the undesirable noise. Finally, the remaining noise is removed by a software tool that was developed to let the user choose which of the remaining tracked blobs is associated to which UAV. The result is a set of UAV image positions $(u_i, v_i)$ for each acquired image, as illustrated in Fig. D.2a. It can be observed that the blob centers are very close to the actual center of the UAV bodies. The error (in pixels) between these positions and the positions manually chosen on the same images was used as an estimate for the blob position accuracy. The results showed an error of around 1% of the image width.

According to a camera pin-hole model, an image position $(u_i, v_i)$ is related to its respective 3D posi-

[12] https://ch.mathworks.com/

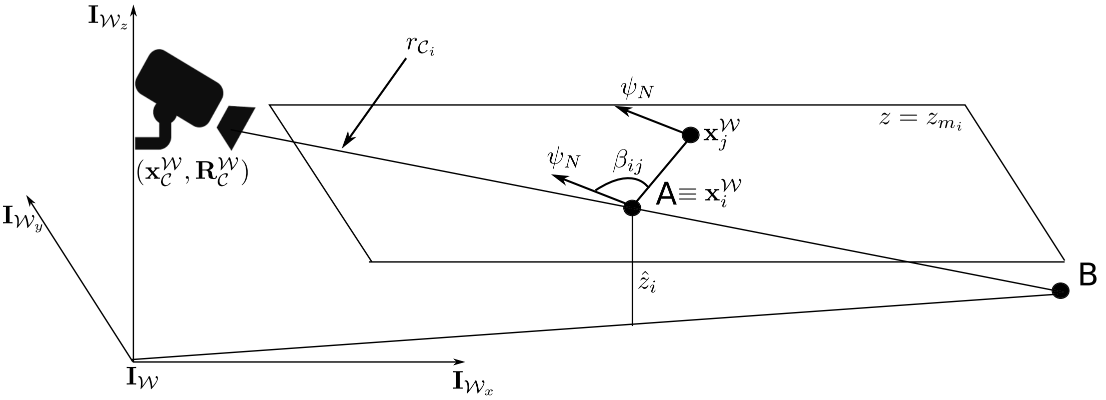

$\mathbf{I}_{\mathcal{W}_z}$
$r_{\mathcal{C}_i}$
$\psi_N$
$\mathbf{x}_j^{\mathcal{W}}$
$z = z_{m_i}$
$(\mathbf{x}_{\mathcal{C}}^{\mathcal{W}}, \mathbf{R}_{\mathcal{C}}^{\mathcal{W}})$
$\psi_N$
$\beta_{ij}$
A$\equiv \mathbf{x}_i^{\mathcal{W}}$
$\mathbf{I}_{\mathcal{W}_y}$
$\hat{z}_i$
B
$\mathbf{I}_{\mathcal{W}}$
$\mathbf{I}_{\mathcal{W}_x}$

Figure D.1: Acquiring the UAV 3D position ground truth by fusing height measurements with bearing measurements provided by an offboard static camera. The height measurements identify the horizontal plane $z = z_{m_i}$ where UAV $i$ is located. The line representing the bearing measurements provided by the camera ($r_{\mathscr{C}_i}$) intersect that plane in point A, which is where UAV $i$ is located in the horizontal plane. Point B is the intersection of $r_{\mathscr{C}_i}$ with the ground plane $z = 0$.

tion in the environment $\mathbf{x}_i^{\mathscr{W}}$ as follows:

$$\lambda(u_i, v_i, 1)^T = \mathbf{M}(\mathbf{R}_{\mathscr{C}}^{\mathscr{W}})^{-1}(\mathbf{x}_i^{\mathscr{W}} - \mathbf{x}_{\mathscr{C}}^{\mathscr{W}}), \tag{D.1}$$

where $(\mathbf{x}_{\mathscr{C}}^{\mathscr{W}}, \mathbf{R}_{\mathscr{C}}^{\mathscr{W}})$ are the camera extrinsic parameters, and $\mathbf{M}$ is the camera intrinsic parameter matrix. This matrix is defined as:

$$\mathbf{M} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix},$$

where $(f_x, f_y)$ is the camera focal length (in image pixels) in each image dimension, and $(u_0, v_0)$ is the camera principal point. The UAV 3D position is described by $\mathbf{x}_i^{\mathscr{W}}$. Note that the previous expression can be modified as follows:

$$\mathbf{x}_i^{\mathscr{W}} = \mathbf{x}_{\mathscr{C}}^{\mathscr{W}} + \lambda \mathbf{M}^{-1}\mathbf{R}_{\mathscr{C}}^{\mathscr{W}}(u_i, v_i, 1)^T. \tag{D.2}$$

By adding the height measurements, one can say that $z_i^{\mathscr{W}} = z_{m_i}$. From this information, and by knowing $(\mathbf{x}_{\mathscr{C}}^{\mathscr{W}}, \mathbf{R}_{\mathscr{C}}^{\mathscr{W}}, \mathbf{M})$ and the respective UAV position in the image $(u_i, v_i)$, it is possible to compute $\lambda$ using Eq. (D.2). Once $\lambda$ is computed, Eq. (D.2) can be reused to compute the remaining UAV horizontal position information $(x_i^{\mathscr{W}}, y_i^{\mathscr{W}})$.

The camera intrinsic parameters $\mathbf{M}$ were computed using a standard camera calibration procedure with a checkerboard. The camera extrinsic parameters $(\mathbf{x}_{\mathscr{C}}^{\mathscr{W}}, \mathbf{R}_{\mathscr{C}}^{\mathscr{W}})$ were computed by choosing the parameters that minimized the error between the predicted and actual projections of several points from the flying arena. An example of the predicted and actual projections of some selected points is illustrated in Fig. D.2b.

During each experiment, the height measurements were gathered on board each UAV at 40 $Hz$,

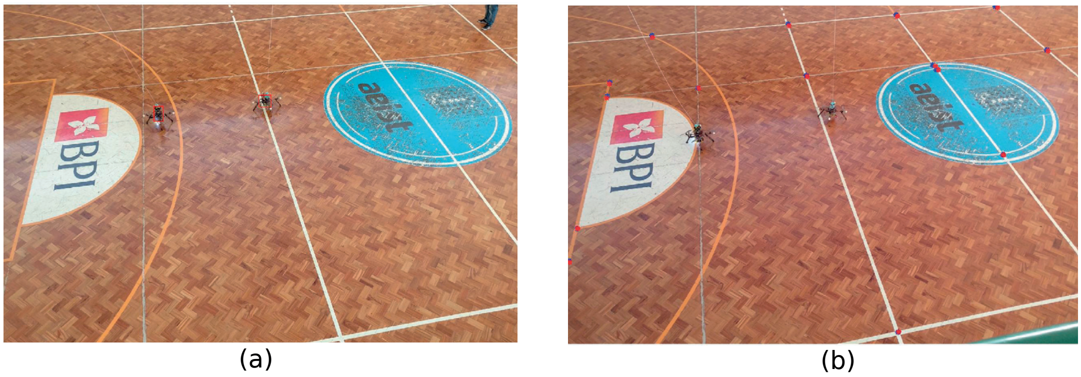


Figure D.2: Necessary image processing in order to acquire the bearing measurements used in the extraction of the UAV 3D position ground truth. (a) Tracking UAV positions in the image. The red squares represent the size of the current detect blob. (b) Calibration of the camera extrinsic parameters by using a set of positions carefully chosen from the flying arena. The red points correspond to the projection of those positions manually selected in the image, and the blue points correspond of the predicted projections of those positions, using the already calibrated camera extrinsic parameters.

while the camera images were forwarded to a computer at 30 $Hz$. After the experiments, all measurements are gathered in the same computer in order to apply the previously described UAV localization method. The height measurements are already associated to the respective UAV. Additionally, as previously mentioned, an user choses the association between the bearing measurements from the camera images and the respective UAV. To synchronize the measurements in time, the takeoff and landing events are used. After the measurements have been synchronized, a measurement set is built for each time step $k$, $(u_i(k), v_i(k), z_{m_i}(k))$. This set can then be used in Eq. (D.2) to compute UAV $i$'s 3D position at time step $k$, $\mathbf{x}_i^{\mathcal{W}}(k)$, as previously described.

After acquiring the position of each UAV $i$, the inter-vehicle localization between UAVs can be acquired by subtracting the position of each UAV. In this way it is possible to measure the relative range and elevation between the UAVs. However, relative bearing measurements can not be measured since the attitude of the UAV is not measurable with the previously described method. However, a rough estimate of the relative bearing between the UAVs can be computed by assuming that the UAVs are always pointing to a specific direction in the environment. In this work the magnetic north direction $\psi_N$ was chosen as this specific direction, as shown in Fig. D.1. Note in the figure how the relative bearing $(\beta_{ij})$ can be extracted using this position between the two UAVs and the assumption that the UAVs are always facing the north direction. The UAVs can control their attitudes towards this north direction using their onboard magnetometers from their IMU sensors.

The maximum relative range and elevation errors computed at different positions of the two UAVs in the environment was of 15 $cm$ and $2.7^o$. The relative bearing measurements were only considered as rough estimates, and their accuracy was not assessed.

Ce document a été imprimé au Centre d’impression EPFL,
imprimerie climatiquement neutre, certifiée myClimate.