7-1-2016

Exploiting Heterogeneity in Networks of Aerial and Ground Robotic Agents

Patricio J. Cruz Davalos

Patricio Javier Cruz Dávalos

Candidate

Electrical and Computer Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Rafael Fierro, Chairperson

Dr. Meeko Oishi

Dr. Lydia Tapia

Dr. Brian Sadler

Exploiting Heterogeneity in Networks of Aerial and Ground Robotic Agents

by

Patricio Javier Cruz Dávalos

Ingeniero en Electrónica y Control, Escuela Politécnica Nacional, Quito-Ecuador, 2005

M.S., Electrical Engineering, University of New Mexico, 2012

DISSERTATION

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy in
Engineering

The University of New Mexico

Albuquerque, New Mexico

July, 2016

Dedication

To Magaly, Eimmy, and Sebastián, mi pequeña y grandiosa familia.
Thanks for being with me at every step of this journey.

"Mantén en comprensión cerebro y mano,
con resulta actitud marcha adelante,
que jamás ha triunfado el vacilante
y el efuerzo con fe nunca es en vano."
– Gustavo Alfredo Jácome

Acknowledgments

I would like to begin by thanking Prof. Rafael Fierro for his trust, support, and time invested in my work. Thanks for your guidance, patience, and encouragement during all these years. Without your help, the rich journey that my family and I have taken would not have been possible. Mil gracias, Prof. Fierro!

I am very grateful to Prof. Meeko Oishi, Prof. Francesco Sorrentino, Prof. Chaouki Abdallah, and Prof. Lydia Tapia for all their advice and feedback during my studies and especially about this work. Also, I wish to thank Dr. Brian Sadler from ARL for his time, ideas, and expertise which have been invaluable to my research.

During my time at UNM, I have had the pleasure to meet with its extraordinary faculty and staff, specially at the ECE department. Thanks all for your enthusiasm and friendliness. Also, I would like to thank all my fellow lab-mates who have been part of our group, the MARHES Lab, during these years. Thanks for your help running experiments, preparing demos, proofreading papers, and making these years memorable.

Thanks to all the great friends, immigrants and natives, that my family and I have made during all these years. Our life has been definitely better because of you. A very special thanks to Janet, our angel in Albuquerque. You are an example of humility, kindness and wisdom. Thanks for making us feel part of your family.

My research was supported in part by the ARL MAST CTA #W911NF-08-2-0004. I am also very grateful to the SENESCYT Ecuadorian Scholarship Program for providing part of the financial support for my studies.

Last but not least, I would like to thank my family with all my heart. To my mother Lupe, my father Miguel, my brother Carlos, and my sister Nadya, thank you for your encouragement, unconditional love, and continuous help despite the distance. To my "parents-in-love" Hilda and Víctor, thank you for your understanding and for caring about us. To Magaly, Eimmy, and Sebas, thanks for your patience, tireless support, love, and joy during this long path. You are the best! I owe you everything I have achieved all these years.

Exploiting Heterogeneity in Networks of Aerial and Ground Robotic Agents

by

Patricio Javier Cruz Dávalos

Ingeniero en Electrónica y Control, Escuela Politécnica Nacional,
Quito-Ecuador, 2005
M.S., Electrical Engineering, University of New Mexico, 2012
Ph.D., Engineering, University of New Mexico, 2016

Abstract

By taking advantage of complementary communication technologies, distinct sensing functionalities and varied motion dynamics present in a heterogeneous multi-robotic network, it is possible to accomplish a main mission objective by assigning specialized sub-tasks to specific members of a robotic team. An adequate selection of the team members and an effective coordination are some of the challenges to fully exploit the unique capabilities that these types of systems can offer. Motivated by real world applications, we focus on a multi-robotic network consisting off aerial and ground agents which has the potential to provide critical support to humans in complex settings. For instance, aerial robotic relays are capable of transporting small ground mobile sensors to expand the communication range and the situational awareness of first responders in hazardous environments.

In the first part of this dissertation, we extend work on manipulation of cable-suspended loads using aerial robots by solving the problem of lifting the cable-

suspended load from the ground before proceeding to transport it. Since the suspended load-quadrotor system experiences switching conditions during this critical maneuver, we define a hybrid system and show that it is differentially-flat. This property facilitates the design of a nonlinear controller which tracks a waypoint-based trajectory associated with the discrete states of the hybrid system. In addition, we address the case of unknown payload mass by combining a least-squares estimation method with the designed controller.

Second, we focus on the coordination of a heterogeneous team formed by a group of ground mobile sensors and a flying communication router which is deployed to sense areas of interest in a cluttered environment. Using potential field methods, we propose a controller for the coordinated mobility of the team to guarantee inter-robot and obstacle collision avoidance as well as connectivity maintenance among the ground agents while the main goal of sensing is carried out. For the case of the aerial communications relays, we combine antenna diversity with reinforcement learning to dynamically re-locate these relays so that the received signal strength is maintained above a desired threshold.

Motivated by the recent interest of combining radio frequency and optical wireless communications, we envision the implementation of an optical link between micro-scale aerial and ground robots. This type of link requires maintaining a sufficient relative transmitter-receiver position for reliable communications. In the third part of this thesis, we tackle this problem. Based on the link model, we define a *connectivity cone* where a minimum transmission rate is guaranteed. For example, the aerial robot has to track the ground vehicle to stay inside this cone. The control must be robust to noisy measurements. Thus, we use particle filters to obtain a better estimation of the receiver position and we design a control algorithm for the flying robot to enhance the transmission rate. Also, we consider the problem of pairing a ground sensor with an aerial vehicle, both equipped with a hybrid radio-frequency/optical wireless communication system. A challenge is positioning the flying robot within

optical range when the sensor location is unknown. Thus, we take advantage of the hybrid communication scheme by developing a control strategy that uses the radio signal to guide the aerial platform to the ground sensor. Once the optical-based signal strength has achieved a certain threshold, the robot hovers within optical range.

Finally, we investigate the problem of building an alliance of agents with different skills in order to satisfy the requirements imposed by a given task. We find this alliance, known also as a coalition, by using a bipartite graph in which edges represent the relation between agent capabilities and required resources for task execution. Using this graph, we build a coalition whose total capability resources can satisfy the task resource requirements. Also, we study the heterogeneity of the formed coalition to analyze how it is affected for instance by the amount of capability resources present in the agents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

During the last decade, technology has pushed the limits providing fast-response sensors and actuators, low-weight high-efficiency batteries, and high-performance embedded processors. At the same time, novel methods and strategies to control and coordinate groups of robots have been developed and implemented. As a result, networked robotic teams are now more capable of providing essential support to human teams in a variety of civilian and military missions, such as urban search-and-rescue, environmental surveillance, and target localization. The accomplishment of these missions places special requirements on the robotic platforms that cannot be easily fulfilled by a team of homogeneous robots. The introduction of robots with different kinematics, sensing and communication capabilities increases the team robustness with respect to the homogeneous counterpart. This allows to operate in highly dynamic environments and to solve increasingly complex tasks. It is also more cost-effective to coordinate a group of robots with limited individual capabilities than to build and control a single powerful robot. In addition, it is impractical to develop large teams of homogeneous robots at the same site, at the same time. A successful

coordinated control and an effective selection of team members can facilitate further research by allowing to easily pool heterogeneous robots to create teams for short-notice task with minimal a priori information.

In multi-robotic systems, the term heterogeneity is used to describe a team of robots that consists of agents with variations in their hardware structure (*hardware-based heterogeneity*) and/or their mission objective (*objective-based heterogeneity*) [1]. Hardware heterogeneity includes, for example, different agent dynamics and distinct sensing capabilities and communication ranges. With respect to objective-based heterogeneity, the overall mission goal can be divided into multiple sub-objectives which are assigned to each agent. A clear example of heterogeneity based on the mission goal is the dynamic sub-task coordination of robotic team in the RoboCup soccer competitions [2]. To exploit the benefits of employing heterogeneous multi-robotic systems, two fundamental decisions should be made depending on the mission to be accomplished: (i) the type of agents which will make up the team; and (ii) the algorithms to be employed to drive the team.

About the latter, the difficulty in deriving suitable controllers for heterogeneous robotic networks lies in handling the hardware variations to combine behaviors in a way that the overall objective is achieved while properties like convergence or stability are maintained. A significant amount of effort has been made to design and implement control laws and coordination methods for a prescribed heterogeneous multi-agent system. Indeed, there are algorithms for teams of autonomous heterogeneous agents to carry out a variety of tasks like sensor coverage [1], area exploration [3] and connectivity maintenance [4], just to mention a few.

Selecting the team of robots which will execute a desired task is a critical step before proceeding with the control strategy implementation. The formation of such a multi-agent system depends mainly on the task requirements and the capabilities available on the heterogeneous agents. The challenge of building a multi-robotic heterogeneous group is referred in the literature as the multi-robot coalition formation

problem [5]. The general idea behind solving the coalition formation is choosing the best group of agents to perform a task. An increase understanding of how a coalition can be generated is beneficial since it enables, for instance, to dynamically add or substitute agents in highly uncertain environments.

1.2 Problem Statement

By using multiple robotic platforms with different dynamics and capabilities, it is possible to overcome the limitations imposed by a homogeneous team of agents trying to accomplish a main mission, such as searching an area of interest, detecting and tracking targets, or surveilling complex environments. A planned selection of the heterogeneous team members together with an effective coordinated control allows fully exploitation of the unique agent's functionalities which is critical to the success of the mission. We focus on a group of UAVs ,*e.g.*, quadrotors, interacting with a group of ground robots, *e.g.*, OctoRoACHes [6], such that they coordinate efforts to achieve a common goal. These platforms are shown in Figure 1.1(a). We envision a scenario where this heterogeneous multi-robotic team extends the situational awareness of task forces in hazardous terrains and confined spaces like caves or urban environments. This heterogeneous robotic network can provide operational capabilities to first human responders that would be otherwise costly, or dangerous to achieve.

Our main goal is to exploit the different dynamics, sensing and communication functionalities of ground and aerial robotic platforms. Being more specific, we consider that quadrotors can transport micro-sized ground robots to a target place as cable-suspended loads. In the last decade, diverse efforts have been made to solve the problem of cable-suspended load transportation using aerial robots. The lift of the payload from the ground is not generally considered even though this maneuver is critical before transporting the cargo. Therefore, it is important to model, analyze,

(a) (b)

Figure 1.1: (a) A group of OctoRoACH crawling robots together with a group of quadrotors. (b) Mission scenario: a quadrotor used as communication relay for a team of ground mobile sensors that are deployed to explore a disaster area. This sketch also shows an optical wireless communication link between the quadrotor and one of the mobile sensors.

and plan the lift of the cable-suspended load to guarantee a safe operation.

Once the ground robots are deployed, they can explore an area where predefined regions of interest has to be sensed. In such operations, wireless communications over the robotic network need to be reliable for coordination purposes. When the transmission is through the air medium, and since we are dealing with multiple vehicles, complications such as multi-path fading and shadow effects arise. These phenomena create a variety of constraints on the possible relative positions of the agents. Thus, we are interested in developing coordination strategies to enhance the connectivity of the network while a common goal is achieved.

Mutual communication among flying/hovering quadrotors and ground-based mobile sensors is fundamentally important, especially for cooperative autonomy. While radio frequency (RF) communications is the common method, optical wireless (OW) links have been proposed as an ideal complement to mitigate some of the weaknesses of RF systems. Figure 1.1(b) shows a possible mission scenario. In this figure, the red light beam between the quadrotor and one of the ground robots illustrates an optical

wireless link between these two platforms. Hybrid RF/OW communications can offer temporary large-scale data transfers within a mobile wireless network. Therefore, we are motivated to study designs for an optical wireless link between an aerial and a ground robotic platform. This type of link requires maintaining an adequate relative transmitter-receiver position for reliable communications.

A critical step before proceeding with the implementation of coordinated control algorithms is to select the best group of agents to perform a given mission. This best group is generally known as a coalition, *i.e.*, an alliance or union of different species of agents that satisfies the requirements imposed by a mission. A continuous understanding of how a coalition is formed is beneficial since it allows to dynamically add or substitute agents on the fly in highly uncertain environments. Thus, it can open the possibility of building teams for short-notice tasks. In addition, it is cost-effective to coordinate a group of resource-bounded agents than operate a single powerful agent. These are just few reasons why this topic remains an active area of research. We also envision to analyze how the heterogeneity of the formed coalition is affected by the resources required to perform a task and the ones available in the potential coalition members.

1.3 Related Work

This dissertation builds on several areas of research, so we have divided the literature review in the following subsections: cable-suspended load aerial transportation, multi-robot coordination under communication constraints, optical wireless communications for robotic agents, and multi-robot coalition formation.

1.3.1 Cable-suspended Load Aerial Transportation

In the last decade, the field of aerial robotics has experienced a fast-growth, especially in the case of multi-rotor UAVs. Possibly the most common multi-rotor aerial platform nowadays is the quadrotor, a simple machine which consists of four individual rotors attached to a rigid cross frame. Quadrotors have better 3-D mobility than fixed-wing UAVs. Their traits (vertical take off and landing, hovering while changing its heading, flying ahead or laterally with the possibility of varying its height, and carrying payloads) have opened a wide spectrum of applications ranging from persistent surveillance [7] to interaction with external objects [8]. Among such applications, aerial load transportation has attracted the attention of several research groups [9–18]. Two approaches have been mainly used for transporting the load. The first one consists on equipping the quadrotor with grippers [9–12], so the load is carried closer to its center of gravity slowing down the response of the vehicle. The second is connecting the payload to the quadrotor by a cable [13–18] which preserves its agility, but the cargo swing can affect the flying characteristics of the aerial robot.

Our work focuses on the problem of transporting cable-suspended loads. In [13], the movement of the system (quadrotor plus suspended load) is restricted to the XZ plane and a controller based on feedback linearization guides the quadrotor to follow a series of waypoints or a predefined trajectory. Trajectory generation under the assumption of minimal load swing at the end of a transport motion is addressed for example by [14]. Dynamic programming and a discrete linearized model of the quadrotor-load system are used to compute an optimal trajectory to be executed by the aerial robot. A similar approach is presented in [15], but a reinforcement learning technique is adopted to generate a swing-free optimal trajectory in a known obstacle-filled environment. The definition and analysis of the hybrid model for the quadrotor plus suspended-load system are introduced, for instance, by [17] to deal with the case when the tension on the cable goes to zero. In this work, a geometric controller is designed whereby local stability properties are achieved. A similar scenario is

considered by [16] where a hybrid model is adopted for the quadrotor carrying a cable-suspended load and the trajectory generation problem is formulated as a Mixed Integer Quadratic Program. The general assumption of a massless cable is relaxed in [18] where the cable connecting the load with the aerial robot is modeled as serially-connected links. Also, geometric control is used to stabilize the vehicle in order that the links are aligned in their vertical position below the quadrotor.

1.3.2 Multi-robot Coordination under Communication Constraints

Coordinated control refers to the ability of a team of robots to work together in order to accomplish a task [19]. Typical coordination tasks for a multi-robot network include sensor coverage [1], area exploration [20], and flocking control [21], to mention a few. In these applications a multi-robot team is generally deployed in an environment populated with obstacles. Thus, the robotic agents experience uncertainty in communication, navigation and sensing. For example, the robots have to move towards a region of interest while avoiding collisions with obstacles and other robots. At the same time, the objects in the environment can create phenomena like shadow effects and secondary reflections which degrade the performance of the inter-agent wireless communication. Consequently, the coordinated control of the multi-robot network should incorporate wireless communication constraints.

The objective of maintaining connectivity together with additional requirements like collision avoidance of a multi-agent system has been extensively studied in the literature. Fink et al. [19] demonstrate that a team of networked robots can maintain end-to-end connectivity in complex environments while they move to accomplish their pre-assigned task. Decentralized controllers based on navigation functions for a group of robots are developed in [22] to satisfy individual sensing goals while some neighborhood connectivity relationships are maintained. In [23], communication

range and line-of-sight are used as motion constraints for a swarm of point robots which goes from an initial to a final configuration in a cluttered environment. Rooker and Birk [24] introduce a multi-robot exploration algorithm that uses a utility function built taking into account the constraints of wireless networking. Monitoring the communication link quality or the construction of a signal strength map are strategies described in [25] for a good link quality maintenance in the deployment of a mobile robot network.

The use of heterogeneous robots to enhance the communications capabilities of the whole multi-robot network has attracted significant attention because it brings new research challenges compared with its homogeneous counterpart. Indeed, the introduction of robots with different kinematics, sensing and communication resources increases the team robustness to operate in highly dynamic environments. Cortez et al. [1] describe a mobile communication relay to a network of sensors and derive connectivity constraints among the network members. Moreover, these constraints are used to maximize the feasible motion sets of the sensing agents. In this work, the network is assumed to move in a free-obstacle environment and its agents are considered as point robots. In [4], a team of UGVs performs a collaborative task while a team of UAVs is positioned in a configuration so that they optimize the communication link quality to support the team of UGVs, but the authors assume that the UGVs are static to guarantee the connectivity of the UAV-UGV network. By use of a heterogeneous robotic system, a search/pursuit scenario is implemented in [3], where a control algorithm guarantees a certain level of Signal-to-Interference plus Noise Ratio (SINR) among the members of the system. Even though the field-of-view of the sensors in the network is considered, the geometry of the agents is neglected. Communication maintenance for a group of heterogeneous robots is enforced in [26] by a passivity-based decentralized strategy. This approach allows creation/deletion of communication links at any time as long as global connectivity is preserved, but the strategy is not tested in the case that the network has a main goal like sensing on top of maintaining connectivity.

1.3.3 Optical Wireless Communications for Robotic Agents

The standard communication technology for wireless operation of robotic platforms is radio-frequency (RF). However, this technology has some limitations including security issues, congested spectrum, and can generate unwanted interference in sensitive environments. At higher frequencies, such as microwave, small scale fading and penetration loss dominate the propagation and result in significant signal attenuation and distortion, especially indoors. Due to these limitations, optical wireless (OW) communications has been proposed as a complement technology to RF systems [27–29]. The underlying concept of OW technology is very simple: utilize optical beams to transmit data through the air from one point to other. This technology has unique advantages over RF such as a wide unlicensed spectrum, lighter and smaller components, relatively simple processing on data transmission/reception, and a high level of security against jamming and network sniffing because of the narrow and high directionality of the optical beam. Indeed, free-space optical (FSO) links have augmented the capacity and capability of RF networks [30]. These hybrid OW/RF systems can offer temporary point-to-point high throughput links within the network. In addition, indoor OW systems are currently being revisited as part of the visible light communication (VLC) framework that aims to combine lighting and communications employing commercially visible light emitting diodes [31, 32]. Furthermore, wireless ultraviolet (UV) technology offers the potential of overcome the line-of-sight (LOS), pointing, acquisition and tracking limitations of optical communication systems [33, 34]. Novel free-space optical (FSO) systems are also under development at the Connectivity Lab at Facebook [35]. This lab seeks to develop new technology for deploying Internet infrastructure to provide affordable connectivity access all over the world.

For robotic platforms, OW communications has been principally explored and applied in the case of underwater vehicles. Since RF is not effective underwater and acoustic-based communication has slow data rates with high latency, OW technol-

ogy has become an attractive alternative solution for wireless data transmission in aquatic settings. The design and control algorithms of an autonomous underwater vehicle equipped with an hybrid acoustic/optical communication system is detailed in [36, 37]. Furthermore, [38] presents a real-time video streaming solution for this underwater vehicle based on a VLC system. In [39], the architecture of a visible-light link for underwater robots is explained. This link can be employed for the dual-use of communication and localization of a mobile aquatic robot. The design and development of a small size LED-based communication system for autonomous underwater vehicles is discussed in [40].

In the case of land applications, OW communications has been proposed to provided coherent connectivity to large numbers of compact nodes which form a mobile ground sensor network [41]. Also, OW systems has been employed for the remote control of small ground robots as in the case of an iRobot PackBot® [42]. Thanks to the payload of this robot, a sturdy system for active pointing and tracking has been implemented and mounted on top of the robot. Similarly, a hybrid FSO/RF transmission system for the deployment of a network of Pionner P3-AT robots is proposed in [43]. Here, vision-based alignment and routing protocols are used to establish optical links among the network. To the best of our knowledge, OW communication systems have not been proposed or studied yet for the case of a heterogeneous robotic team of aerial and ground vehicles.

1.3.4 Building Multi-robot Coalitions

Picking the team of agents which will execute a desired task is a critical step before proceeding with the implementation of coordination algorithms. The problem of forming such a multi-agent system is known as the multi-agent coalition formation problem [5, 44]. Coalition is a term that states alliance or union among different species in a biological domain. The general idea behind the coalition formation of agents is to select the best agents to perform a task. Thus, the members of the

coalition can assemble their complementary capabilities to satisfy the requirements imposed by a given mission. A better understanding about how to form coalitions will facilitate picking up teams of robots having possibly very minimal a priori information of the environment or the whole mission goal. In addition, it will allow integrating new agents into existing heterogeneous robotic teams in order to perform efficiently under dynamic or uncertain conditions.

It is well known that the optimal solution for the multi-agent coalition problem is NP-hard [45]. However, closer problems such as Set Partitioning and Set Covering have been extensively studied and many heuristics for approximate solutions of the coalition formation problem have been proposed [5, 45]. However, as noted in [44], many of these algorithms cannot be applied directly for the case of multi-robot systems since end effector resources and sensing capabilities are not transferable between robots. In the last years, several of the approaches to multi-agent coalition formation have been modified to facilitate its application in the multi-robot domain [44,46,47]. For example, the authors in [44] expand the well-known Shehory algorithm for generating a multi-agent coalition to the case of robots by adapting, for instance, its task and communication format. Also, bio-inspired optimization algorithms [48] and game theory approaches [49] have been applied to solve this problem for the case of robots. In [48], ant colony optimization is combined with a simulated annealing technique to address the general combinatorial problem associated with multi-robot coalition formation. An evolutionary stable strategy from Game Theory is applied in [49] to built coalition of robots for the task of detection and capture of an intruder.

1.4 Contributions

A common assumption when solving the problem of transporting a cable-suspend load using a quadrotor is that the system is always in the air, so the lift of the load from the ground is not generally considered. However, aerial cargo lifting is a

fast and efficient way to move materials to locations beyond the practical reach of perimeter cranes. Furthermore, this maneuver is critical before transporting the payload. For example for cargo lifting using helicopters, the vehicle has to be over the load before the helicopter starts to lift according to safety regulations for these types of operations [50, 51]. To the best of our knowledge, the problem of autonomously lifting a cable-suspended load by a quadrotor UAV has not been discussed in the literature yet. The quadrotor plus cable-suspended load system experiences switching dynamics during the lifting. This switching behavior arises when, for instance, the cable goes instantaneously from being slack to being taut, *i.e.*, the cable tension jumps from zero to a non-zero value, while the quadrotor is climbing. In addition, the system experiences another state jump when the load is not in contact with the ground anymore. Indeed, this second switching condition has not been considered in the hybrid model introduced in the literature for the cable-suspended aerial transportation system [16, 17]. Due to these transitions, the lift maneuver can be broken down into a collection of simpler discrete states or modes with different dynamics for each one. Thus, we decompose the lift maneuver into modes that characterize the dynamics of the quadrotor-load system at particular operation regimes. Furthermore, we define a hybrid system based on these modes and show that this hybrid model is differentially-flat according to the definition given by [17]. This property facilitates the generation of trajectories since a smooth trajectory with reasonably bounded derivatives can be followed by a differentially-flat system. Therefore, we generate a minimum jerk trajectory using a series of waypoints associated with the modes of the lift maneuver. Then, we designed a nonlinear controller that enables the tracking of the generated trajectory. Also, we present the experimental evaluation of the proposed approach by using a commercially-available micro-scale quadrotor UAV to lift a cable-suspended load from the ground.

By combining the lift maneuver with the work in cable-suspended load transportation using aerial robotic, it is possible to deploy a group of small ground robots for carrying out sensing or search missions in complex environments. Once they are

deployed, we propose a control strategy to enhance connectivity of this network of robots while a number of regions of interest are sensed. We consider that the aerial robots are better equipped to relay information over longer distance. On the other hand, the ground mobile vehicles have better sensing capabilities. This allows that the ground mobile sensor network spreads out more effectively around the mission space. Our algorithm takes into account the geometry of the sensor field-of-view (FOV) and of the robotic platforms. Also, it guarantees inter-robot collision avoidance as well as collision avoidance with general shaped obstacles in the environment. We use potential field methods to coordinate the ground mobile sensors, so they move toward the regions of interest, avoiding obstacles and inter-agent collisions. The same method is applied to control one aerial communication relay to maintain connectivity among the ground sensors and a fixed base station. We extend our approach to the case of multiple aerial relays employing a cooperative reinforcement learning (RL) technique to dynamically relocate the relays in such a way that the whole multi-robot stays connected. In particular, our algorithm seeks to increase the received signal strength (RSS) among the relays and the sensors. RSS is a reasonable proxy for link quality, so by improving the RSS, it is possible to enhance connectivity in networked robotic teams [52, 53].

Mutual communication among flying and ground robots is required for coordinated task execution. While the standard communication technology for this goal is RF, OW technology is the perfect complement to augment the capacity of RF-based networks. Thus, we envision integrating OW systems into heterogeneous mobile robotic platforms; specifically, flying robots carrying optical transmitters and ground robots equipped with optical receivers. We consider an OW link which provides a secure communication channel while RF provides mutual, global or other perhaps less secure communications. Such hybrid communication system will expand the capabilities of such wireless heterogeneous robotic networks. To the best of our knowledge, OW communications has not been proposed for the case of a robotic team of aerial and ground vehicles. Possibly the major shortcoming of OW

technology that has been delayed its mobile application is the line-of-sight (LOS) pointing and tracking requirement. Indeed, this challenge has to be addressed to fully exploit the benefits of the optical link. We present our approach to address the problem of tracking a ground receiver by an aerial transmitter in order to establish a point-to-point optical communication link. Based on the optical link model, we establish a *connectivity cone* over the receiver where a minimum transmission rate is guaranteed. We consider that only noisy measurements of the receiver position are available for the transmitter. Using Bayesian methods, we compute an estimation of the location of the *connectivity cone*. Then, we develop a control to reduce the distance between the aerial transmitter and the cone. Once the transmitter is within the cone, the control acts to optimize the possible communication rate. Also, we consider that a ground sensor and an aerial vehicle are both equipped with a hybrid RF/optical communication system - RF for low bandwidth transmission and optical for high rate transfer. A challenging problem is positioning the flying robot within optical communication range, especially when the distance is large and the sensor location is unknown. Thus, we propose a solution to the problem of autonomously localizing the sensor node relative to the aerial vehicle. We take advantage of the hybrid communication scheme by developing a control strategy that uses the radio signal to guide the aerial platform to the sensor node. Once the optical-based signal strength is over a desired threshold the robot hovers within optical range.

One critical step to exploit the distinct resources available on a set of agents is to form an alliance of agents that satisfies the requirements imposed by a given mission. Therefore, we study the coalition formation problem and propose a solution using a weighted bipartite graph which expresses the relation between the resources required to execute a task and the capabilities available in each one of the possible agents. This bipartite graph enables to form a coalition whose total capability resources satisfy the resource requirements imposed by the given task. In addition, we measure the heterogeneity of the formed coalition and analyze how it is affected by the resources required by the task and the resources present in the agents.

1.5 Overview

In Chapter 2, we discuss our approach to perform autonomously the lift maneuver of cable-suspended load by an aerial robot. Indeed, the proposed control methodology is able to carry-out the lift maneuver even when the load mass is unknown. Chapter 3 details a coordinated control algorithm for a network formed by a group of ground mobile robots and a group of aerial communication relays which are deployed in a cluttered environment to sense areas of interest. The study of the design of an OW communication link between an aerial and a ground robot is explained in Chapter 4. Also, our control strategy to solve the line-of-sight pointing and tracking requirement for establishing a reliable OW link is part of this section. Our proposed approach to solve the multi-robot coalition problem based on weighted bipartite graphs is presented in Chapter 5. Moreover, we analyze in this section how the heterogeneity of the coalition is affected by the conditions required by the task and the amount of resources present in the agents. Finally, Chapter 6 provides our concluding remarks and potential future work directions.

1.6 Summary of Publications

The work of Chapter 2 has been published in the 2014 *IEEE Conference on Control Applications* and in the 2015 *American Control Conference*. The journal version of this chapter has been submitted for publication in *Autnomous Robots*.

[i] P. Cruz and R. Fierro, "Autonomous lift of a cable-suspended load by an unmanned aerial robot", *IEEE Conference on Control Applications* (CCA), part of *IEEE Multi-conference on Systems and Control* (MSC), Nice/Antibes, France, Oct 8-10, 2014, pp. 802-807.

[ii] P. J. Cruz, M. Oishi, and R. Fierro, "Lift of a cable-suspended load by a quadrotor: a hybrid system approach", *American Control Conference* (ACC), Chicago,

USA, July 1-3, 2015, pp 1887-1892.

[iii] P. J. Cruz and R. Fierro, "Cable-suspended lifting by a quadrotor UAV: hybrid model, trajectory generation and control", *Autonomous Robots*, submitted April 2016. (Under review)

The results of Chapter 3 are published in:

[iv] P. Cruz, R. Fierro, W. Lu, S. Ferrari, and T. Wettergren, "Maintaining robust connectivity in heterogeneous robotic networks", *SPIE Defense, Security and Sensing Conference, Unmanned Systems Technology XV*, April 29-May 3, 2013, pp. 87 410N87 410N15.

[v] P. J. Cruz, C. Abdallah, and R. Fierro, "Cooperative learning for robust connectivity in multi-robot heterogeneous networks", in *Control of Complex Systems: Theory and Applications*, K. Vamvoudakis and J. Sarangapani (eds.), Elseiver, August 2016. (To appear)

The work in Chapter 4 has been published in the 2015 *International Conference in Unmanned Aircraft Systems* and in the 2016 *American Control Conference*. The journal version of this chapter is under preparation and it will be submitted to the International Journal of Robotics Research.

[vi] P. J. Cruz, and R. Fierro, "Towards optical wireless communications between micro unmanned aerial and ground systems", *International Conference in Unmanned Aircraft Systems* (ICUAS), Denver, CO, USA, June 9-12, 2015, pp. 669-676.

[vii] P. J. Cruz, B. M. Sadler, and R. Fierro, "Sensor localization using hybrid RF/optical wireless communications for an aerial data mule", *American Control Conference* (ACC), Boston, USA, July 6-8, 2016, pp. 7085-7091.

[viii] P. J. Cruz, B. M. Sadler, C. Hintz, J. Dostal, and R. Fierro, "Enabling optical wireless communications for micro-sized aerial and ground robots", *International Journal of Robotics Research* (IJRR). (Under preparation)

The results of Chapter 5 are published in:

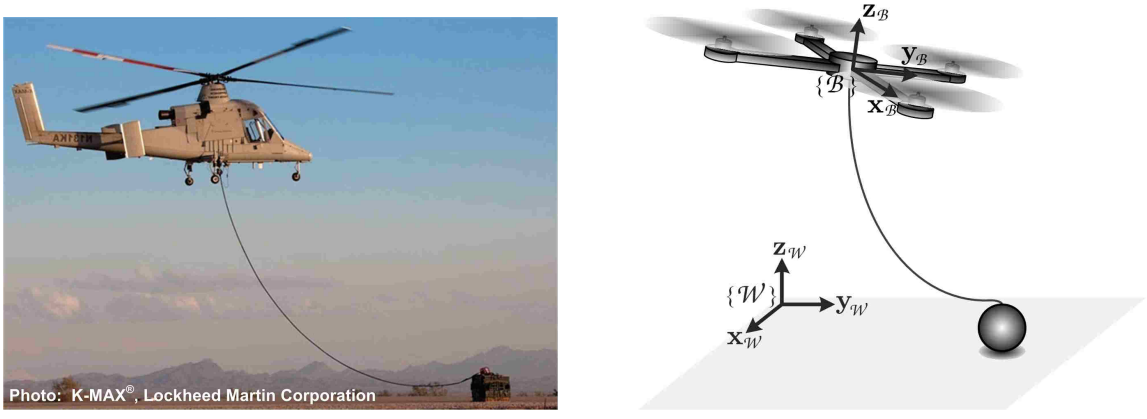
[ix] P. J. Cruz and R. Fierro, "Building coalitions of heterogeneous agents using weighted bipartite graphs", *IEEE Conference on Decision and Control* (CDC), Osaka, Japan, December 15-18, 2015, pp. 2822-2828.

Chapter 2

Autonomous Lift of a Cable-Suspended Load

Unmanned rotorcraft vehicles are better suited for a variety of applications than unmanned fixed-wing systems. For example, autonomous cargo delivery using robocopters has been successfully evaluated for delivering pallets of cargo to remote military bases [54], see Figure 2.1(a). A particular type of unmanned rotorcraft vehicle, the quadrotor, has been extensively used by academic and governmental research groups worldwide showing its capability for aerial manipulation [55], aerial construction [12], and cable-suspended load transportation [14]. In the last years, diverse efforts have been made to solve the problem of manipulating a cable-suspended load with quadrotors [14, 17, 56–61]. A common assumption when dealing with this problem is that the load is always in the air. To the best of our knowledge, how the lift maneuver can be performed autonomously has not been addressed in the literature yet. During this critical maneuver before transporting the load, the quadrotor-load system experiences switching conditions. For example, the cable goes instantaneously from being slack to being taut. This transition is known as *cable collision* [62]. Furthermore, the system has another state jump when the load is not in contact with the ground anymore. Because of these transitions, the maneuver can be broken down

into a collection of modes with different system dynamics for each one. Decomposing the maneuver simplifies the planning and control of the overall system. A similar approach has been used successfully in a variety of applications including backflip maneuvers for quadrotors [63] and aircraft trajectory planning [64].



(a) (b)

Figure 2.1: (a) The K-MAX® developed by Lockheed Martin Corporation. It has been successfully tested for remote controlled cargo delivery. (b) A quadrotor with an attached cable-suspended load which is lying over the ground. $\{\mathcal{W}\}$ and $\{\mathcal{B}\}$ are the inertial and body-fixed coordinate frames, respectively.

Therefore, we decomposed the lift maneuver into simpler hybrid modes: *Setup, Pull* and *Raise*, which represent the dynamics of the system at particular operation regimes. We define a hybrid system for the lift maneuver and we demonstrate that the hybrid model is indeed a differentially-flat hybrid system according to the definition given in [17]. The flatness property facilitates the generation of trajectories since a smooth trajectory with reasonably bounded derivatives can be followed by the system. Thus, we generate a minimum jerk trajectory based on a series of waypoints related to the decomposition of the lift maneuver. A nonlinear geometric controller enables the tracking of the generated trajectory which is validated by numerical simulations. Furthermore, we modify the control design to verify the proposed approach by carrying out experiments on an actual quadrotor with a cable-suspended load.

This chapter is based on our work published in [65, 66].

2.1 Preliminaries

We consider a scenario where a quadrotor has a point-mass load attached by a massless and unstreatchable cable. We assume the cable is attached to the center of mass (CoM) of the quadrotor and the load mass is less than the maximum payload of the quadrotor. Also, we consider that the air drag is negligible. Figure 2.1(b) shows the system together with the inertial coordinate frame $\{\mathcal{W}\}$, and the body-fixed frame $\{\mathcal{B}\}$. The origin of $\{\mathcal{B}\}$ coincides with the CoM of the quadrotor. Based on Figure 2.1(b), we introduce first the following definitions:

$\{\mathbf{x}_{\mathcal{W}}, \mathbf{y}_{\mathcal{W}}, \mathbf{z}_{\mathcal{W}}\}$	unit vectors along the axes of $\{\mathcal{W}\}$,
$\{\mathbf{x}_{\mathcal{B}}, \mathbf{y}_{\mathcal{B}}, \mathbf{z}_{\mathcal{B}}\}$	unit vectors along the axes of $\{\mathcal{B}\}$ with respect to $\{\mathcal{W}\}$,
$m_q \in \mathbb{R}_{>0}$	mass of the quadrotor,
$\mathbf{J} \in \mathbb{R}^{3\times 3}$	inertia matrix of the quadrotor with respect to $\{\mathcal{W}\}$,
$\mathbf{r}_q, \mathbf{v}_q \in \mathbb{R}^3$	position and velocity of the quadrotor with respect to $\{\mathcal{W}\}$, $\mathbf{r}_q = [x_q \;\; y_q \;\; z_q]^T$ and $\mathbf{v}_q = [\dot{x}_q \;\; \dot{y}_q \;\; \dot{z}_q]^T$,
$\mathbf{R} \in \mathrm{SO}(3)$	rotation matrix from $\{\mathcal{B}\}$ to $\{\mathcal{W}\}$,
$\mathbf{\Omega} \in \mathbb{R}^3$	angular velocity of the quadrotor in $\{\mathcal{B}\}$,
$F \in \mathbb{R}_{\geq 0}$	total thrust produced by the quadrotor,
$\mathbf{M} \in \mathbb{R}^3$	moment produced by the quadrotor,
$m_l \in \mathbb{R}_{>0}$	mass of the load,
$\mathbf{r}_l, \mathbf{v}_l \in \mathbb{R}^3$	position and velocity of the load with respect to $\{\mathcal{W}\}$, $\mathbf{r}_l = [x_l \;\; y_l \;\; z_l]^T$ and $\mathbf{v}_l = [\dot{x}_l \;\; \dot{y}_l \;\; \dot{z}_l]^T$,
$\ell \in \mathbb{R}_{>0}$	length of the cable, and
$T \in \mathbb{R}_{\geq 0}$	tension on the cable.

Now, let $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ be the three coordinate axis unit vectors without a frame of reference, *i.e.*, $\mathbf{e}_1 = [1 \;\; 0 \;\; 0]^T$, $\mathbf{e}_2 = [0 \;\; 1 \;\; 0]^T$ and $\mathbf{e}_3 = [0 \;\; 0 \;\; 1]^T$, then algebraically in

$\{\mathcal{W}\}$

$$\mathbf{x}_{\mathcal{W}} = \mathbf{e}_1, \mathbf{y}_{\mathcal{W}} = \mathbf{e}_2, \text{ and } \mathbf{z}_{\mathcal{W}} = \mathbf{e}_3. \tag{2.1}$$

This implies by construction that

$$\mathbf{x}_{\mathcal{B}} = \mathbf{R}\mathbf{e}_1, \mathbf{y}_{\mathcal{B}} = \mathbf{R}\mathbf{e}_2, \text{ and } \mathbf{z}_{\mathcal{B}} = \mathbf{R}\mathbf{e}_3. \tag{2.2}$$

One of the transitions that the quadrotor-load system experiences during the lift maneuver is the jump of the cable tension T from zero to a nonzero value. This happens when the cable goes instantaneously from being slack to being taut. This transition is known as *cable collision* [62] and because of it, we need to consider the models of the quadrotor with and without a cable-suspended load.

2.1.1 Quadrotor Dynamics

The equations of motion of the quadrotor without carrying any load are the ones defined when just the quadrotor is under consideration and they can be written as [67, 68]

$$\begin{aligned} \dot{\mathbf{r}}_q &= \mathbf{v}_q, \\ m_q \dot{\mathbf{v}}_q &= -m_q g \mathbf{z}_{\mathcal{W}} + F \mathbf{z}_{\mathcal{B}}, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\Omega}}, \\ \mathbf{J}\dot{\boldsymbol{\Omega}} &= -\boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} + \mathbf{M}, \end{aligned} \tag{2.3}$$

where g is the constant gravitational acceleration, and the *hat map* $\hat{\cdot} : \mathbb{R}^3 \to \mathrm{SO}(3)$ denotes the skew-symmetric matrix defined by the condition that $\hat{\boldsymbol{\Omega}}\boldsymbol{b} = \boldsymbol{\Omega} \times \mathbf{b}$ for the vector cross product of $\boldsymbol{\Omega}$ and any vector $\mathbf{b} \in \mathbb{R}^3$ [67].

2.1.2 Quadrotor-Suspended-Load Dynamics

When the quadrotor is carrying a cable-suspended load, the cable tension is nonzero and the load is on the air. Then, the dynamics of the quadrotor and the load can be

written down using the tension in the cable [17]

$$\begin{aligned}
\dot{\mathbf{r}}_q &= \mathbf{v}_q, \\
m_q\dot{\mathbf{v}}_q &= -m_q g\mathbf{z}_{\mathcal{W}} + F\mathbf{z}_{\mathcal{B}} - T\boldsymbol{\mu}, \\
\dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\Omega}}, \\
\mathbf{J}\dot{\boldsymbol{\Omega}} &= -\boldsymbol{\Omega}\times\mathbf{J}\boldsymbol{\Omega} + \mathbf{M}, \\
\dot{\mathbf{r}}_l &= \mathbf{v}_l, \\
m_l\dot{\mathbf{v}}_l &= -m_l g\mathbf{z}_{\mathcal{W}} + T\boldsymbol{\mu}.
\end{aligned} \tag{2.4}$$

Here, $\boldsymbol{\mu}$ is the unit vector from the load to the quadrotor. For this system, the quadrotor and load positions are related by

$$\mathbf{r}_q = \mathbf{r}_l + \ell\boldsymbol{\mu}. \tag{2.5}$$

2.2 Lift Maneuver

In this section, we first formulate the problem of lifting a cable-suspended load by a quadrotor UAV and then we decompose the lift maneuver into three modes: *Setup, Pull* and *Raise* which represent the dynamics of the whole system in specific regimes during the maneuver.

2.2.1 Problem Statement

Starting with the quadrotor hovering at a given altitude not necessarily right on top of the load, see Figure 2.1(b), the goal is to lift the load until it reaches a predefined height denoted as h. Since the quadrotor has attached the cable-suspended load since the beginning, the relative quadrotor-load distance cannot be more than a cable-length apart. Also, we consider that the mass of the load m_l is unknown. Under these conditions, we formulate the lifting problem.

Problem 2.1. *Having a cable-suspended load lying on the ground at the initial position* $\mathbf{r}_{l_0}$ *which is attached to a quadrotor UAV hovering at the position* $\mathbf{r}_{q_0}$ *such that*

$$\|\mathbf{r}_{q_0} - \mathbf{r}_{l_0}\| < \ell, \tag{2.6}$$

the quadrotor has to lift the load until it reaches the final position

$$\mathbf{r}_{l_f} = \mathbf{r}_{l_0} + h\mathbf{z}_{\mathcal{W}} \tag{2.7}$$

assuming that the load mass is unknown.

Remark 2.1. *According to* (2.6)*, the quadrotor does not start right on top of the cable-suspended load with the cable tensioned. However, the aerial robot has to reach this position before proceeding to lift the load according to safety guidelines for aerial transportation of external payloads* [50, 51]*. In fact, the quadrotor can exert the highest lift force when it is right over the load with the cable fully extended. We denote this position as* $\mathbf{r}_{\text{pull}}$ *and it is given by*

$$\mathbf{r}_{\text{pull}} = \mathbf{r}_{l_0} + \ell\mathbf{z}_{\mathcal{W}}. \tag{2.8}$$

In Section 2.2.2, we design the hybrid system that models the lift maneuver. Subsequently, in Section 2.4, we present the methodology to perform this maneuver under the assumption of no knowledge of the load mass. We use a least-squares method to estimate the mass of the system and from there the load mass since the mass of the quadrotor is known. The simulation and experimental results are detailed in Sections 2.5 and 2.6, respectively.

2.2.2 Lift Maneuver Modes

The modes of the lift maneuver are sketched in Figure 2.2. These modes are *Setup, Pull* and *Raise*. We break down the lift maneuver into these simpler modes to characterize the dynamics of the system in specific regimes during the maneuver. This

(a) (b) (c)

Figure 2.2: The lift maneuver: (a) Setup, (b) Pull, and (c) Raise. The initial state of the system is illustrated in (a).

decomposition is due to the jump from zero to nonzero cable tension and because of the load transition from being in contact with the ground to be in the air. Furthermore, the aerial robot has to be over the load before starting to lift it according to the safety regulations for the aerial transportation of external payloads [51]. Therefore, the *Setup* and *Pull* are modes where the quadrotor gets ready to lift the load, while the *Raise* mode is where the payload is finally lifted to the final position $\mathbf{r}_{l_f}$.

Setup

From condition (2.6), the quadrotor starts at an initial position where the cable is not fully extended. Therefore, the cable tension is equal to zero, see Figure 2.2(a). Due to this condition, the quadrotor and the attached payload can be considered as separate systems. Thus, the dynamics of the aerial vehicle are the ones given in (2.3), while the load is at rest. Then, the equations of motion for this mode can be

written as

$$\begin{bmatrix} \dot{\mathbf{r}}_q \\ \dot{\mathbf{v}}_q \\ \dot{\mathbf{R}} \\ \dot{\boldsymbol{\Omega}} \\ \dot{\mathbf{r}}_l \\ \dot{\mathbf{v}}_l \end{bmatrix} = \begin{bmatrix} \mathbf{v}_q \\ -g\mathbf{e}_3 + \frac{F}{m_q}\mathbf{R}\mathbf{e}_3 \\ \mathbf{R}\hat{\boldsymbol{\Omega}} \\ \mathbf{J}^{-1}\left(-\boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} + \mathbf{M}\right) \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{2.9}$$

The system jumps to the next mode, *Pull*, when the quadrotor and the load are exactly a cable-length apart, *i.e.*, when the cable is fully extended. We can express this condition as

$$\|\mathbf{r}_q - \mathbf{r}_l\| \geq \ell. \tag{2.10}$$

When condition (2.10) holds, the cable jumps from being slack to be taut. This jump, known as *cable collision*, causes that the positions of the aerial vehicle and the load remain the same, but their change in velocity can be modeled as a perfectly inelastic collision [62]. For the next derivations, we follow closely the work made by [62]. Any collision, elastic or inelastic, can be modeled using the conservation of momentum. Thus, the relation between translational velocity before and after the impact can be described by

$${}^{+}\mathbf{v}_q = {}^{-}\mathbf{v}_q + \frac{\delta}{m_q}\boldsymbol{\mu}, \text{ and} \tag{2.11}$$

$${}^{+}\mathbf{v}_l = {}^{-}\mathbf{v}_l - \frac{\delta}{m_l}\boldsymbol{\mu}, \tag{2.12}$$

where δ is the impulse of the collision. Here, the pre-superscript $-$ $(+)$ denotes the situation just before (after) the collision. The relative velocity of the two attachment points on the cable (one in the quadrotor and the other in the load) characterizes an impact by

$$-k_e\left({}^{-}\mathbf{v}_q - {}^{-}\mathbf{v}_l\right) \cdot \boldsymbol{\mu} = \left({}^{+}\mathbf{v}_q - {}^{+}\mathbf{v}_l\right) \cdot \boldsymbol{\mu}, \tag{2.13}$$

where $k_e \in [0, 1]$ is the elasticity constant such that $k_e = 0$ describes a perfect inelastic collision and $k_e = 1$ describes a perfect elastic collision. The cable collision is modeled as a perfect inelastic collision in order to ensure that ${}^{+}\mathbf{v}_q - {}^{+}\mathbf{v}_l = \mathbf{0}$. Replacing (2.11) and (2.12) into (2.13), the impulse δ can be isolated. Since the load is at rest before the perfect inelastic collision, we get

$$\delta = -m_q m_l \frac{{}^{-}\mathbf{v}_q \cdot \boldsymbol{\mu}}{m_q + m_l}. \tag{2.14}$$

By using (2.14), one can determine the impulse from the cable collision and then applying (2.11) and (2.12), it is possible to compute the states after the transition.

Pull

The quadrotor is over the load with the cable fully extended at this mode, so $T \neq 0$. Even though the cable tension T is not any more zero, it could be not enough to lift the payload. Thus, the load is still on the ground. Furthermore, the forces acting on the quadrotor-load system are balanced and the whole system is motionless. Since the system is at equilibrium, we get that

$$-m_q g \mathbf{e}_3 + F\mathbf{R}\mathbf{e}_3 - T\boldsymbol{\mu} = \mathbf{0},$$

and solving for the tension, we obtain

$$T = \|F\mathbf{R}\mathbf{e}_3 - m_q g \mathbf{e}_3\|. \tag{2.15}$$

Thus, the total thrust F has to be increased in order to increment the tension on the cable and then lift the load. Once T is slightly over the load weight $m_l g$, there is enough tension that the cable-suspended load starts being lifted. Therefore, we define the condition

$$T > m_l g \tag{2.16}$$

as an indication to jump to the *Raise* mode. On the other hand, the transition from *Pull* back to *Setup* occurs when the cable tension becomes zero, *i.e.*, when the cable

returns to be slack. This condition can be expressed as

$$T \leq \epsilon, \tag{2.17}$$

where ϵ is small and positive. Notice that the quadrotor and load positions are related in this mode by (2.5) since the cable is fully extended.

Raise

At this stage, the load is in the air with the quadrotor over it with the cable completely taut (Figure 2.2(c)). Thus, the equations of motion are given by the quadrotor-suspended-load dynamics given in (2.4) from where we get that

$$\begin{bmatrix} \dot{\mathbf{r}}_q \\ \dot{\mathbf{v}}_q \\ \dot{\mathbf{R}} \\ \dot{\boldsymbol{\Omega}} \\ \dot{\mathbf{r}}_l \\ \dot{\mathbf{v}}_l \end{bmatrix} = \begin{bmatrix} \mathbf{v}_q \\ -g\mathbf{e}_3 + \frac{F}{m_q}\mathbf{R}\mathbf{e}_3 - \frac{T}{m_q}\boldsymbol{\mu} \\ \mathbf{R}\hat{\boldsymbol{\Omega}} \\ \mathbf{J}^{-1}\left(-\boldsymbol{\Omega} \times \mathbf{J}\boldsymbol{\Omega} + \mathbf{M}\right) \\ \mathbf{v}_l \\ -g\mathbf{e}_3 + \frac{T}{m_l}\boldsymbol{\mu} \end{bmatrix}. \tag{2.18}$$

From the last component of (2.18), it is possible to find the cable tension during this mode obtaining

$$T = \|m_l\dot{\mathbf{v}}_l + m_l g\mathbf{e}_3\|. \tag{2.19}$$

The cable is also fully extended in this mode, so the quadrotor and load positions are related by (2.5). In addition, the system goes back to the *Pull* mode when the load is again over the ground. We can capture this condition by the following relation

$$|z_l - z_{l_0}| \leq \epsilon. \tag{2.20}$$

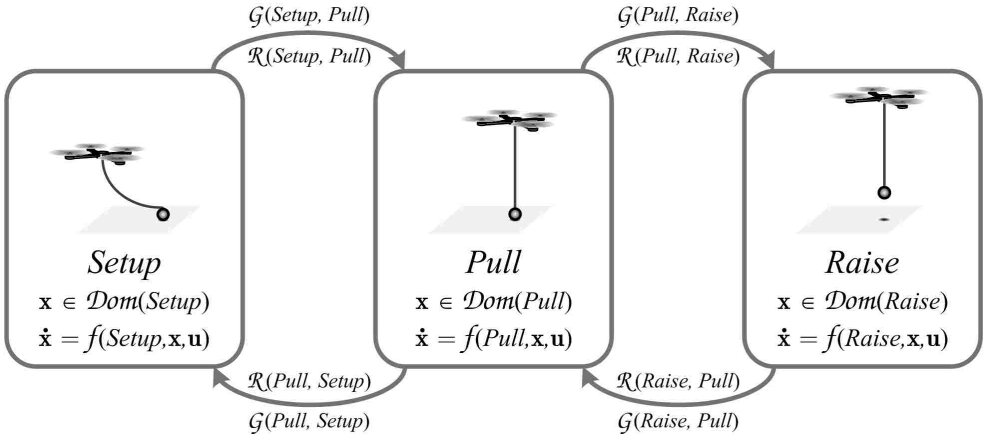


Figure 2.3: The lift maneuver represented as a hybrid system. The definitions of its discrete and continuous states, vector fields, domains, edges, guards, and reset maps are given in Section 2.3.

2.3 Differentially-Flat Hybrid System

Based on the decomposition of the lift maneuver presented in Section 2.2, we define a hybrid system (see Figure 2.3). Following the hybrid automaton representation [69], we define the hybrid model as the tuple

$$\mathcal{H} = (\mathcal{Q}, \mathcal{X}, \mathcal{U}, f, \mathcal{D}om, \mathcal{E}, \mathcal{G}, \mathcal{R}, \mathit{Init}),$$

where

- $\mathcal{Q} = \{\mathit{Setup},\ \mathit{Pull},\ \mathit{Raise}\}$ is the set of discrete states,
- $\mathcal{X} = \mathrm{SO}(3) \times \mathbb{R}^{15}$ is the set of continuous states with the state $\mathbf{x} \in \mathcal{X}$ defined as $\mathbf{x} = \{\mathbf{r}_q, \mathbf{v}_q, \mathbf{R}, \mathbf{\Omega}, \mathbf{r}_l, \mathbf{v}_l\}$,
- $\mathcal{U} = \mathbb{R}^4$ is the set of input variables and we define $\mathbf{u} = \begin{bmatrix} F & \mathbf{M}^T \end{bmatrix}^T \in \mathcal{U}$ as the control input of the system,
- $f(\mathit{Setup}, \mathbf{x}, \mathbf{u})$ given by (2.9), $f(\mathit{Pull}, \mathbf{x}, \mathbf{u}) = \mathbf{0}$, and $f(\mathit{Raise}, \mathbf{x}, \mathbf{u})$ given by (2.18) are the vector fields,

- $\mathcal{Dom}(\mathit{Setup}) = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{v}_l = \mathbf{0} \text{ and } \|\mathbf{r}_q - \mathbf{r}_l\| < \ell\} \times \mathcal{U}$,
  $\mathcal{Dom}(\mathit{Pull}) = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{v}_q = \mathbf{v}_l = \mathbf{0} \text{ and } \mathbf{r}_q = \mathbf{r}_l + \ell\boldsymbol{\mu}\} \times \mathcal{U}$, and
  $\mathcal{Dom}(\mathit{Raise}) = \{\mathbf{x} \in \mathcal{X} \mid \mathbf{r}_q = \mathbf{r}_l + \ell\boldsymbol{\mu}\} \times \mathcal{U}$ are the domains,

- $\mathcal{E} = \{(\mathit{Setup},\ \mathit{Pull}), (\mathit{Pull},\ \mathit{Setup}), (\mathit{Pull},\ \mathit{Raise}), (\mathit{Raise}, \mathit{Pull})\}$ is the set of edges,

- $\mathcal{G}(\mathit{Setup},\ \mathit{Pull}) = \{\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U} \mid \|\mathbf{r}_q - \mathbf{r}_l\| \geq \ell\}$,
  $\mathcal{G}(\mathit{Pull},\ \mathit{Setup}) = \{\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U} \mid T \leq \epsilon\}$,
  $\mathcal{G}(\mathit{Pull},\ \mathit{Raise}) = \{\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U} \mid T > m_l g\}$, and
  $\mathcal{G}(\mathit{Raise},\ \mathit{Pull}) = \{\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U} \mid |z_l - z_{l_0}| \leq \epsilon\}$ are the guard conditions,

- the reset map $\mathcal{R}(\mathit{Setup},\ \mathit{Pull})$ is given by (2.11) and (2.12) where δ can be found using (2.14), while we assume that $\mathcal{R}(\mathit{Pull},\ \mathit{Setup})$, $\mathcal{R}(\mathit{Pull},\ \mathit{Raise})$, and $\mathcal{R}(\mathit{Raise},\ \mathit{Pull})$ are the identity map, and

- $\mathit{Init} = \{\mathit{Setup}\} \times \{\mathbf{x} \in \mathcal{X} \mid \mathbf{v}_l = \mathbf{0} \text{ and } \|\mathbf{r}_{q_0} - \mathbf{r}_{l_0}\| < \ell\}$ is the set of initial states.

Notice that the hybrid system $\mathcal{H}$ has a non-identity reset map only for the transition from *Setup* to *Pull*. Also, $\mathcal{G}(\mathit{Pull},\ \mathit{Setup})$ and $\mathcal{G}(\mathit{Pull},\ \mathit{Raise})$ are not state-based guard conditions. They depend on the tension value which can be found by (2.15) for the *Pull* mode. This hybrid model is a more complete version than the one that we introduced in [66] for the planar case of the lifting problem. For example, both edges (*Pull, Setup*) and (*Raise, Pull*) as well as their guards and reset maps have been added in this paper. As compared with hybrid models for a quadrotor carrying a cable-suspended-load found in the literature [16, 17], our hybrid automaton $\mathcal{H}$ considers also the transition from having the load on the ground to having it on the air and not only the jump from zero tension to nonzero tension.

Next, we introduce the definition of differential flatness for the case of a hybrid system. Then, we demonstrate that indeed $\mathcal{H}$ is a differentially-flat hybrid system.

Definition 2.1 (Differentially-flat hybrid system [17])**.** *In general, a system is differentially-flat if its state and inputs can be written as functions of the selected outputs and their derivatives. In the case of a hybrid system, each discrete mode has to be differentially-flat with the guards being functions of the flat outputs and their derivatives, and the flat outputs of one mode arise as smooth functions of the flat outputs of the previous mode through the transition or reset map between both modes.*

Lemma 2.1. *The hybrid system $\mathcal{H}$ is a differentially-flat hybrid system.*

Proof. First, we show that each discrete mode, *Setup, Pull*, and *Raise*, are differentially flat. For the *Setup* mode, we select $\mathcal{Y}_{\text{setup}} = \{\mathbf{r}_q, \psi\}$ as the set of flat outputs where ψ is the yaw angle of the quadrotor. Notice that the state of the load is always equal to zero in this mode since the load is at rest, so $\mathbf{v}_l = \mathbf{0}$. Thus, it suffices to show that $\mathcal{Y}_{\text{setup}}$ is a set of flat outputs for the quadrotor. Indeed, this has been already proved by [70]. Therefore, the *Setup* mode is differentially flat. For the *Pull* mode, we choose $\mathcal{Y}_{\text{pull}} = \{\mathbf{r}_q, \psi\}$ as the set of flat outputs. The load is also motionless at this mode. Thus, based on the same reason as for the previous mode, the *Setup* mode is also differentially flat. For the *Raise* mode, we choose $\mathcal{Y}_{\text{raise}} = \{\mathbf{r}_l, \psi\}$. The position and velocity of the load can be obtained from $\mathcal{Y}_{\text{raise}}$ and $\dot{\mathcal{Y}}_{\text{raise}}$. For the quadrotor, first we need to express $\boldsymbol{\mu}$ as function of the flat input and its derivatives. From the hybrid system model, the vector field of this mode is given by (2.18). Using the last component of this vector field, we find that

$$\boldsymbol{\mu} = \frac{\ddot{\mathbf{r}}_l + g\mathbf{e}_3}{\|\ddot{\mathbf{r}}_l + g\mathbf{e}_3\|}. \tag{2.21}$$

Thus, $\boldsymbol{\mu}$ is a function of the second derivative of the flat output. For this mode, the quadrotor is a cable-length apart from the load, see Section 2.2.2. Indeed, their positions are related by (2.5). Replacing (2.21) into (2.5), we get the position of the quadrotor

$$\mathbf{r}_q = \mathbf{r}_l + \ell \frac{\ddot{\mathbf{r}}_l + g\mathbf{e}_3}{\|\ddot{\mathbf{r}}_l + g\mathbf{e}_3\|}. \tag{2.22}$$

Therefore, $\mathbf{r}_q$ is a function of the flat output and its second derivative. All remaining quantities, $\mathbf{v}_q$, $\mathbf{R}$, $\mathbf{\Omega}$, F, and $\mathbf{M}$, can be determined from the knowledge of $\mathbf{r}_q$ and ψ since these are flat outputs for the quadrotor.

We have demonstrated so far that the discrete modes of $\mathcal{H}$ are differentially-flat. Now, we check the guard conditions. Since $\mathcal{G}(\textit{Setup, Pull})$ and $\mathcal{G}(\textit{Raise, Pull})$ are state-based, both guards are clearly functions of their respectively flat outputs, $\mathcal{Y}_{\text{setup}}$ and $\mathcal{Y}_{\text{raise}}$, and their corresponding derivatives. The other two guard conditions, $\mathcal{G}(\textit{Pull, Setup})$ and $\mathcal{G}(\textit{Pull, Raise})$, are for the *Pull* mode and they depend on the cable tension. Indeed, the tension on the cable for this mode can be find applying (2.15) which depends on $\mathbf{R}$ and F. As we already shown, these two quantities can be written as functions of the set of flat outputs $\mathcal{Y}_{\text{pull}}$ and their derivatives. Hence, the cable tension can be fully determined by knowing $\mathcal{Y}_{\text{pull}}$. As a result, the guards $\mathcal{G}(\textit{Pull, Setup})$ and $\mathcal{G}(\textit{Pull, Raise})$ are functions of the set of flat outputs $\mathcal{Y}_{\text{pull}}$ and their high-order derivatives.

The map $\mathcal{Y}_{\text{pull}}$ to $\mathcal{Y}_{\text{raise}}$ and vice versa are the identity since $\mathcal{R}(\textit{Pull, Raise})$, and $\mathcal{R}(\textit{Raise, Pull})$ are the identity reset map. Similarly, we have for $\mathcal{Y}_{\text{pull}}$ to $\mathcal{Y}_{\text{setup}}$. For *Setup* to *Pull*, the reset map given by (2.11) and (2.12) where δ can be found using (2.14) help to make the transition from $\mathcal{Y}_{\text{setup}}$ to $\mathcal{Y}_{\text{pull}}$.

Consequently, we know for $\mathcal{H}$ that its discrete modes are differentially-flat, its guards are functions of the selected flat outputs for each mode and their corresponding derivatives, and the selected flat output for every mode arises from the flat output of the previous mode according to the reset maps. Hence by Definition 2.1, $\mathcal{H}$ is a differentially-flat hybrid system. □

2.4 Control Methodology

The overall structure of the control methodology presented in Section 2.4 is illustrated in Figure 2.4(a). In this section, first we detail the generation of a reference trajectory

$\mathbf{r}_q^{\text{ref}}$ to drive the system into the modes of the lift maneuver. Second, we design a geometric controller similar to the one presented in [68, 70] which can follow the prescribed trajectory $\mathbf{r}_q^{\text{ref}}$. Finally, we present an estimator of the load mass which can be used in the cases where it is unknown.

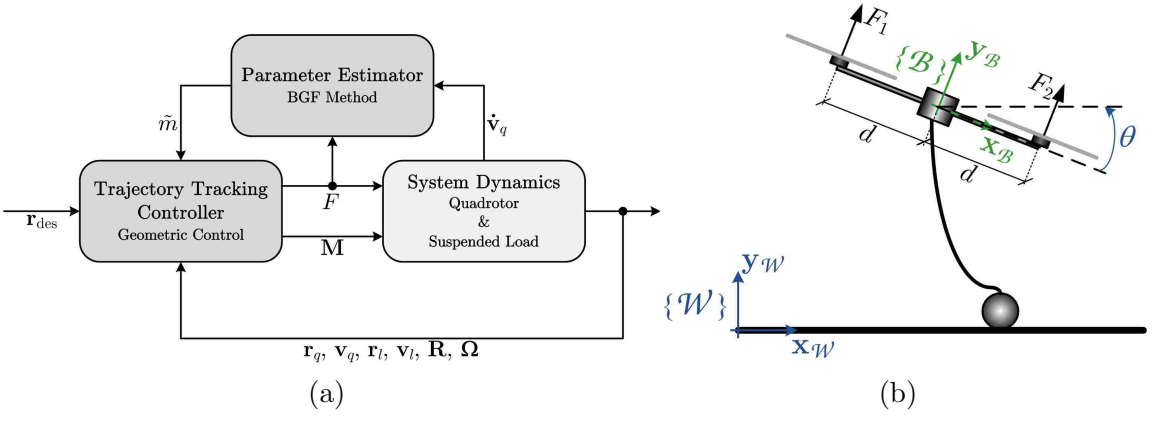


Figure 2.4: (a) Block diagram of the control methodology. (b) A planar quadrotor with an attached cable-suspended load. F_1 and $F_2 \in \mathbb{R}$ are the force produced by each motor. The total thrust produced by the quadrotor is given by $F = F_1 + F_2$ and $M = d(F_2 - F_1)$ is the torque produced by the quadrotor. Here d is the distance between the quadrotor CoM and each rotor axis.

2.4.1 Trajectory Generation

Building on the results of Section 2.3, we consider a trajectory in the space of flat outputs such as

$$\mathcal{Y}(t) : [0, t_f] \rightarrow \mathbb{R}^3 \times \text{SO}(2),$$

where

$$\mathcal{Y}(t) = \begin{bmatrix} x_q(t) & y_q(t) & z_q(t) & \psi(t) \end{bmatrix}^T.$$

Since a change in the yaw angle does not have any effect on the lift maneuver, we assume that $\psi(t) = 0°$ all the time. Thus, we need to create a trajectory to

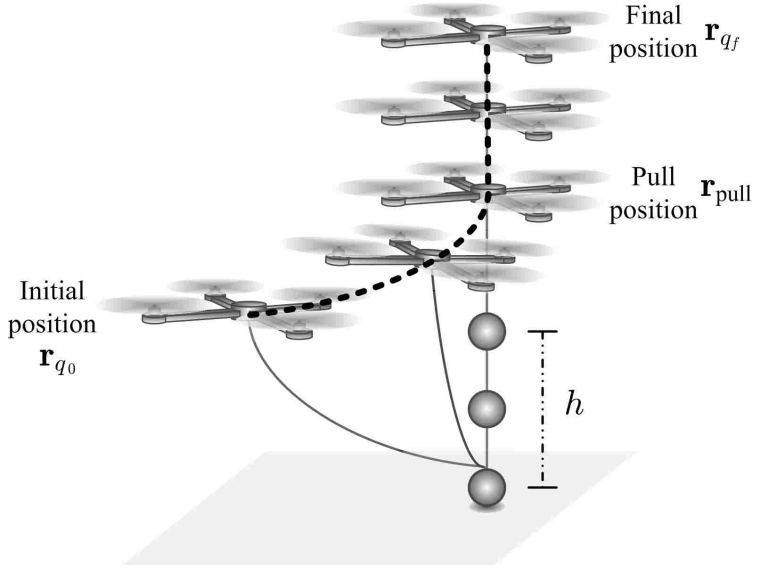


Figure 2.5: Overview of the trajectory required to execute the lift maneuver.

perform the lift maneuver just for the quadrotor position $\mathbf{r}_q(t) = [x_q(t)\ \ y_q(t)\ \ z_q(t)]^T$. Furthermore, minimizing the fourth derivative of the position (the derivative of the acceleration known as jerk) ensures a smooth trajectory for the quadrotor [71]. Next, we present our method to generate an optimal minimum jerk trajectory.

Related to each discrete state of the hybrid system, there are reference positions for the quadrotor that can be used to generate a trajectory to execute the lift maneuver. These waypoints are:

1. associated with the *Setup* mode, the initial position $\mathbf{r}_{q_0}$ which satisfies condition (2.6),

2. with the *Pull* mode, the position $\mathbf{r}_{\text{pull}}$ where the quadrotor can exert the highest lift force and it is given by (2.8), and

3. with the *Raise* mode, the final position $\mathbf{r}_{q_f}$ which relates the desired final position of the load with the final position of the quadrotor and it is

$$\mathbf{r}_{q_f} = \mathbf{r}_{l_f} + \ell \mathbf{e}_3. \tag{2.23}$$

We consider the second waypoint $\mathbf{r}_{\text{pull}}$ as a viapoint between $\mathbf{r}_{q_0}$ and $\mathbf{r}_{q_f}$, *i.e.*, a point prior to reach the final quadrotor position. Thus, we generate a trajectory that starts at $\mathbf{r}_{q_0}$, passes through $\mathbf{r}_{\text{pull}}$, and ends at $\mathbf{r}_{q_f}$ (see Figure 2.5). We assume that the aerial vehicle stops at the viapoint $\mathbf{r}_{\text{pull}}$ and ends at rest at the final goal point $\mathbf{r}_{q_f}$. Therefore, we have two segments in our trajectory: from $\mathbf{r}_{q_0}$ to $\mathbf{r}_{\text{pull}}$ and from $\mathbf{r}_{\text{pull}}$ to $\mathbf{r}_{q_f}$, where the quadrotor starts and ends at rest for both cases. Since the generation of the trajectory is identical for both segments and for all the three coordinates, we take as an example the case for x_q, the x-axis component of $\mathbf{r}_q$. Let $\mathbf{s}$ be the state defined as

$$\mathbf{s} = \begin{bmatrix} s_1 & s_2 & s_3 \end{bmatrix}^T = \begin{bmatrix} x_q & \dot{x}_q & \ddot{x}_q \end{bmatrix}^T,$$

then we define the dynamics of $\mathbf{s}$ as

$$\dot{\mathbf{s}} = \begin{bmatrix} s_2 & s_3 & u \end{bmatrix}, \tag{2.24}$$

with the initial condition $\mathbf{s}(0) = [x_0 \quad 0 \quad 0]^T$. Notice that $\dddot{x}_q = \dot{s}_3 = u$ is the jerk. We want to minimize with respect to u the following cost function

$$J = \int_0^{t_f} u^2 dt, \tag{2.25}$$

having as final state constraint $\mathbf{s}(t_f) = \begin{bmatrix} x_f & 0 & 0 \end{bmatrix}^T$. Here, t_f and x_f are the desired final time and position, respectively. This constraint has to be satisfied without error, so it is a hard constraint [72, 73].

We follow the methodology explained in [73] in order to find the solution of our continuous-time optimization problem. The Hamiltonian $\mathscr{H}$ for our case is

$$\mathscr{H} = u^2 + \lambda_1 s_2 + \lambda_2 s_3 + \lambda_3 u, \tag{2.26}$$

where $\lambda_1, \lambda_2, \lambda_3$, are the adjoint variables which form the vector $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \ \lambda_3]^T$ which is known as the adjoint vector. The optimal value of u (denoted as u^*) can be

found by solving

$$\begin{aligned}\dot{\mathbf{s}} &= \frac{\partial \mathscr{H}}{\partial \boldsymbol{\lambda}} = \begin{bmatrix} s_2 & s_3 & u \end{bmatrix}^T, \\ \dot{\boldsymbol{\lambda}} &= -\frac{\partial \mathscr{H}}{\partial \mathbf{s}} = \begin{bmatrix} 0 & -\lambda_1 & -\lambda_2 \end{bmatrix}^T, \\ 0 &= \frac{\partial \mathscr{H}}{\partial u} = 2u + \lambda_3. \end{aligned} \tag{2.27}$$

The last equation in (2.27) indicates that $u^* = -\frac{\lambda_3}{2}$, so we need to find λ_3 to determine its optimal value. Replacing u^* into the first equation of (2.27) yields the state and the adjoint equations

$$\dot{\mathbf{s}} = \begin{bmatrix} s_2 & s_3 & -\frac{\lambda_3}{2} \end{bmatrix}^T, \text{ and} \tag{2.28}$$

$$\dot{\boldsymbol{\lambda}} = \begin{bmatrix} 0 & -\lambda_1 & -\lambda_2 \end{bmatrix}^T, \tag{2.29}$$

respectively, which have the initial condition $\mathbf{s}(0) = [x_0 \quad 0 \quad 0]^T$ and the final constraint $\mathbf{s}(t_f) = \begin{bmatrix} x_f & 0 & 0 \end{bmatrix}^T$. We can solve (2.29) assuming that we knew the final condition for the adjoint vector $\boldsymbol{\lambda}(t_f) = [\lambda_{1_f} \ \lambda_{2_f} \ \lambda_{3_f}]^T$. Using this solution and replacing it in (2.28), we can find the state trajectories which yields

$$\begin{aligned} s_1 = x_q &= -\frac{\lambda_{1_f}}{240}t^5 + \frac{\lambda_{1_f}t_f + \lambda_{2_f}}{48}t^4 - \frac{k}{24}t^3 + x_0, \\ s_2 = \dot{x}_q &= -\frac{\lambda_{1_f}}{48}t^4 + \frac{\lambda_{1_f}t_f + \lambda_{2_f}}{12}t^3 - \frac{k}{8}t^2, \\ s_3 = \ddot{x}_q &= -\frac{\lambda_{1_f}}{12}t^3 + \frac{\lambda_{1_f}t_f + \lambda_{2_f}}{4}t^2 - \frac{k}{4}t, \end{aligned} \tag{2.30}$$

where $k = \lambda_{1_f}t_f^2 + 2\lambda_{2_f}t_f + 2\lambda_{3_f}$ and the final conditions for the adjoint variables $\lambda_{1_f}, \lambda_{2_f}$ and λ_{3_f} are given by

$$\begin{bmatrix} \lambda_{1_f} \\ \lambda_{2_f} \\ \lambda_{3_f} \end{bmatrix} = \begin{bmatrix} -\frac{t_f^2}{40} & -\frac{t_f}{16} & -\frac{1}{12} \\ -\frac{t_f^2}{16} & -\frac{t_f}{6} & -\frac{1}{4} \\ -\frac{t_f^2}{12} & -\frac{t_f}{4} & -\frac{1}{2} \end{bmatrix}^{-1} \begin{bmatrix} \frac{x_f - x_0}{t_f^3} \\ 0 \\ 0 \end{bmatrix}. \tag{2.31}$$

By applying (2.30) for each coordinate and for each path segment, we generate the minimum jerk reference trajectory $\mathbf{r}_q^{\text{ref}}$ whose first and second derivatives are denoted as $\mathbf{v}_q^{\text{ref}}$ and $\mathbf{a}_q^{\text{ref}}$, respectively. Tracking this trajectory, we can execute the lift maneuver. In the next section, we design the controller to accomplish this goal.

2.4.2 Geometric Control

First, we define the position and velocity errors by

$$\mathbf{e}_p = \mathbf{r}_q - \mathbf{r}_{\text{des}}, \quad \mathbf{e}_v = \mathbf{v}_q - \dot{\mathbf{r}}_{\text{des}}. \tag{2.32}$$

Next, the desired force vector for the controller is computed

$$\mathbf{F}_{\text{des}} = -\mathbf{K}_p\mathbf{e}_p - \mathbf{K}_v\mathbf{e}_v + \tilde{m}\left(g\mathbf{z}_{\mathcal{W}} + \ddot{\mathbf{r}}_{\text{des}}\right), \tag{2.33}$$

where $\mathbf{K}_p$ and $\mathbf{K}_v$ are positive definite gain matrices and $\tilde{m}$ is the mass of the system given by

$$\tilde{m} = \begin{cases} m_q & \text{if } \mathcal{q} = \{Setup\}, \\ m_q + m_l & \text{otherwise,} \end{cases} \tag{2.34}$$

where $\mathcal{q} \in \mathcal{Q}$. From (2.33), notice that $\|\mathbf{F}_{\text{des}}\| \neq 0$ at any time. The translational dynamics of the aerial vehicle is controlled by the projection of the desired force vector onto the third body-fixed axis giving us the total required thrust

$$F = \mathbf{F}_{\text{des}} \cdot \mathbf{z}_{\mathcal{B}}, \tag{2.35}$$

where $\mathbf{z}_{\mathcal{B}}$ was defined in (2.2). On the other hand, we need to consider the attitude error and the angular velocity error to find the desired torque $\mathbf{M}$. For the attitude error, the desired direction of the third body-fixed axis to stabilize the translational dynamics can be chosen by $\mathbf{z}_{\mathcal{B}_{\text{des}}} = \frac{\mathbf{F}_{\text{des}}}{\|\mathbf{F}_{\text{des}}\|}$. Since the yaw angle should be maintained equal to zero, the heading direction of the aerial robot in the plane normal to $\mathbf{z}_{\mathcal{B}_{\text{des}}}$ can be determined by the desired direction of the first body-fixed axis $\mathbf{x}_{\mathcal{B}_{\text{des}}} = [\cos 0^\circ \;\; \sin 0^\circ \;\; 0]^T = [1 \;\; 0 \;\; 0]^T$. To obtain the desired attitude $\mathbf{R}_{\text{des}}$, we project $\mathbf{x}_{\mathcal{B}_{\text{des}}}$ onto the plane normal to $\mathbf{z}_{\mathcal{B}_{\text{des}}}$ obtaining

$$\mathbf{R}_{\text{des}} = \begin{bmatrix} \mathbf{y}_{\mathcal{B}_{\text{des}}} \times \mathbf{z}_{\mathcal{B}_{\text{des}}} & \mathbf{y}_{\mathcal{B}_{\text{des}}} & \mathbf{z}_{\mathcal{B}_{\text{des}}} \end{bmatrix} \in \text{SO}(3), \tag{2.36}$$

where $\mathbf{y}_{\mathcal{B}_{\text{des}}} = \frac{\mathbf{z}_{\mathcal{B}_{\text{des}}} \times \mathbf{x}_{\mathcal{B}_{\text{des}}}}{\|\mathbf{z}_{\mathcal{B}_{\text{des}}} \times \mathbf{x}_{\mathcal{B}_{\text{des}}}\|}$. Then, we define the attitude error

$$\mathbf{e}_R = \frac{1}{2}\left(\mathbf{R}_{\text{des}}^T\mathbf{R} - \mathbf{R}^T\mathbf{R}_{\text{des}}\right)^\vee, \tag{2.37}$$

where $^\vee$ represents the *vee map*: $\mathrm{SO}(3) \to \mathbb{R}^3$ [68]. Under the assumption of keeping $\psi = 0$, the desired angular velocity of the quadrotor can be written as [70]

$$\boldsymbol{\Omega}_{\mathrm{des}} = p_{\mathrm{des}} \mathbf{x}_{\mathcal{B}_{\mathrm{des}}} + q_{\mathrm{des}} \mathbf{y}_{\mathcal{B}_{\mathrm{des}}}. \tag{2.38}$$

Here $p_{\mathrm{des}} = -\mathbf{h}_\omega \cdot \mathbf{y}_{\mathcal{B}_{\mathrm{des}}}$ and $q_{\mathrm{des}} = \mathbf{h}_\omega \cdot \mathbf{x}_{\mathcal{B}_{\mathrm{des}}}$ with

$$\mathbf{h}_\omega = \frac{\tilde{m}}{F} \left(\dddot{\mathbf{r}}_{\mathrm{des}} - \left(\mathbf{z}_{\mathcal{B}_{\mathrm{des}}} \cdot \dddot{\mathbf{r}}_{\mathrm{des}} \right) \mathbf{z}_{\mathcal{B}_{\mathrm{des}}} \right).$$

Thus, we can specify the angular velocity error by

$$\mathbf{e}_\Omega = \boldsymbol{\Omega} - \boldsymbol{\Omega}_{\mathrm{des}}. \tag{2.39}$$

Using (2.37) and (2.39), the desired moment to be produced by the quadrotor can be computed as follows

$$\mathbf{M} = -\mathbf{K}_{\mathrm{R}} \mathbf{e}_{\mathrm{R}} - \mathbf{K}_\Omega \mathbf{e}_\Omega + \left(\boldsymbol{\Omega} \times \mathbf{J} \boldsymbol{\Omega} \right), \tag{2.40}$$

where $\mathbf{K}_{\mathrm{R}}$ and $\mathbf{K}_\Omega$ are diagonal gain matrices. From (2.35) and (2.40), we can compute the control inputs to the quadrotor cable-suspended load system.

2.4.3 System Mass Estimation

So far we have assumed that the total mass of the system $m_l + m_q$ is known. However, this is not generally the case specially for the mass of the load m_l which is a critical parameter during the *Pull* and *Raise* modes. Thus, it is required to infer this unknown parameter from on-line measurements of the input and output signals of the system. In our case, the next three Newton's equations give information about the mass of the system [55]

$$\begin{aligned} \tilde{m} \dot{v}_{q_x} &= F \mathbf{z}_{\mathcal{B}} \cdot \mathbf{x}_{\mathcal{W}} + f_x, \\ \tilde{m} \dot{v}_{q_y} &= F \mathbf{z}_{\mathcal{B}} \cdot \mathbf{y}_{\mathcal{W}} + f_y, \\ \tilde{m} \dot{v}_{q_z} &= F \mathbf{z}_{\mathcal{B}} \cdot \mathbf{z}_{\mathcal{W}} + f_z, \end{aligned} \tag{2.41}$$

where $\dot{v}_{q_x}$, $\dot{v}_{q_y}$, $\dot{v}_{q_z}$ are the components of the translational acceleration vector of the quadrotor. We denote as $\mathbf{a}_q$ to this acceleration vector, *i.e.*, $\mathbf{a}_q = [\dot{v}_{q_x}\ \dot{v}_{q_y}\ \dot{v}_{q_z}]^T$, and f_x, f_y, f_z are the forces acting on the quadrotor. Indeed, f_x and f_y are considered as lateral aerodynamic disturbance forces, while f_z is formed by the weight and the cable tension. The system of equations in (2.41) can be expressed as

$$\mathbf{a}_q = \mathbf{W}\boldsymbol{\theta}, \tag{2.42}$$

with

$$\mathbf{W} = \begin{bmatrix} F\mathbf{z}_{\mathcal{B}} \cdot \mathbf{x}_{\mathcal{W}} & 1 & 0 & 0 \\ F\mathbf{z}_{\mathcal{B}} \cdot \mathbf{y}_{\mathcal{W}} & 0 & 1 & 0 \\ F\mathbf{z}_{\mathcal{B}} \cdot \mathbf{z}_{\mathcal{W}} & 0 & 0 & 1 \end{bmatrix}, \text{ and } \boldsymbol{\theta} = \begin{bmatrix} \frac{1}{\tilde{m}} & \frac{f_x}{\tilde{m}} & \frac{f_y}{\tilde{m}} & \frac{f_z}{\tilde{m}} \end{bmatrix}^T.$$

Notice that $\mathbf{a}_q$ is the acceleration of the quadrotor, so it can be measured using an appropriate sensor. The matrix $\mathbf{W}$ can be computed since F is given by (2.35) and $\mathbf{z}_{\mathcal{B}}$ by (2.2). Thus, the only unknowns in (2.42) are the parameters in $\boldsymbol{\theta}$. Let $\tilde{\boldsymbol{\theta}}$ be the estimation of $\boldsymbol{\theta}$, so the prediction error can be determined by $\mathbf{e}_{\text{pred}} = \mathbf{a}_q - \mathbf{W}\tilde{\boldsymbol{\theta}}$. A useful technique to estimate the unknown parameter vector is the least-squares method with forgetting factor [74]. In this technique, one minimizes

$$J = \int_0^t \exp\left[-\int_s^t \gamma(r)dr\right] \left\|\mathbf{a}_q(s) - \mathbf{W}(s)\tilde{\boldsymbol{\theta}}\right\|^2 ds,$$

with respect to $\tilde{\boldsymbol{\theta}}$. Here γ is the time-varying data forgetting factor. The solution to this optimization problem [74, 75] gives the parameter update law of the form

$$\dot{\tilde{\boldsymbol{\theta}}} = -\mathbf{P}\mathbf{W}^T\mathbf{e}_{\text{pred}}, \tag{2.43}$$

where the estimator gain matrix $\mathbf{P}$ is updated according to

$$\dot{\mathbf{P}} = -\gamma\mathbf{P} - \mathbf{P}\mathbf{W}^T\mathbf{W}\mathbf{P}. \tag{2.44}$$

In order to use (2.43) and (2.44) for on-line estimation, we need to provide our best guess to initialize $\tilde{\boldsymbol{\theta}}$ and the initial gain $\mathbf{P}_0$ is diagonal for simplicity. The benefit of

data forgetting is the ability of tracking time-varying parameters. However, it suffers from possible gain unboundedness. One technique to overcome this problem is to choose γ as

$$\gamma = \gamma_0 \left(1 - \frac{\|\mathbf{P}\|}{k_0}\right), \tag{2.45}$$

so the data forgetting is active when $\mathbf{W}$ is persistently exciting and it is suspended when $\mathbf{W}$ is not [74]. Intuitively, the persistent excitation condition means that the dynamics are excited sufficiently to identify the unknown parameters. The positive constants γ_0 and k_0 represent the maximum forgetting rate and the specified bound for the magnitude of the gain matrix, respectively. The least-squares method with γ given by (2.45) is known as the bounded-gain-forgetting (BGF) estimator [74]. This estimator has been successfully used for controlling robot manipulators [74] and also vehicles [76]. We use this method to estimate the mass of the system.

2.5 Simulation Results

In order to verify the proposed approach, we run numerical simulations first for the 2-D version of the lift problem and then for the 3-D case. We use the load mass estimator just for the 3-D case.

2.5.1 2-D Case

We run first a set of simulations for the planar case of the lifting problem. Similar as in [17,63], we assume that the out of plane dynamics can be stabilized. Thus, we specialize the hybrid system, the control methodology and the trajectory generation to this case. A complete treatment for the planar case can be found in our published work [66]. Figure 2.4(b), shows the 2-D version of the system illustrated in Figure 2.1(b). For this case, we assume that we have complete knowledge of the system mass at all times. Therefore, we do not consider the estimator detailed in Section

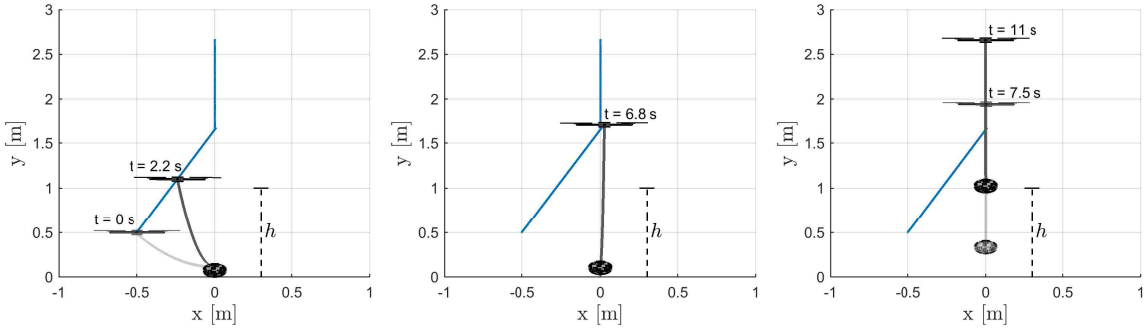
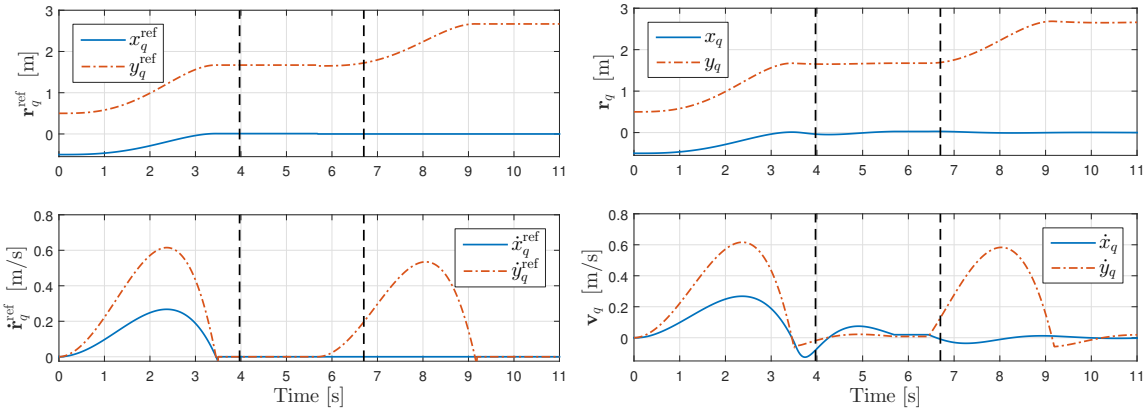


Figure 2.6: Snapshots of the quadrotor-load system during the simulation of the lift maneuver: (a) *Setup*, (b) *Pull*, and (d) *Raise*. The blue line is the trajectory generated to execute the lift maneuver. h is the desired height.

2.4.3 for the simulation. At the beginning of the simulation, we generate the trajectory segments using (2.30) and (2.31). Employing the generated segments and the geometric controller specialized for the planar case, we run the simulation such that the quadrotor performs the lift maneuver. Figure 2.6 shows a series of snapshots of the maneuver. The discrete mode jumps from *Setup* to *Pull* around of 4 seconds after starting the maneuver and from *Pull* to *Raise* after around 6.8 seconds. The total time to perform the lift maneuver is about 11 seconds. The impulse δ during the cable collision computed by (2.14) is equal to 0.052.

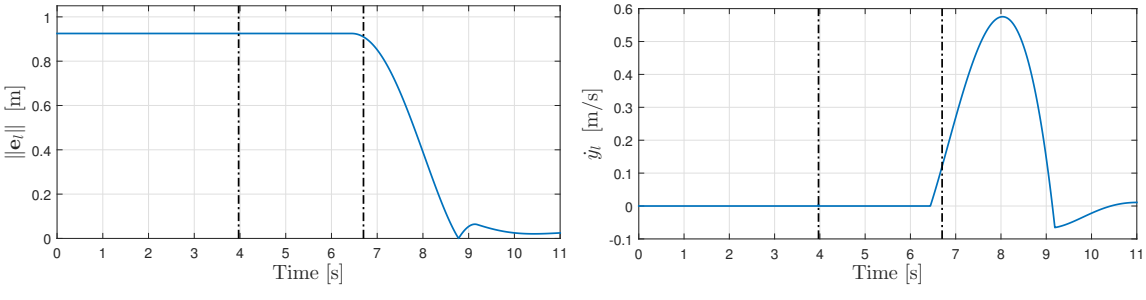
The desired trajectory, position and velocity, and the actual trajectory are illustrated in Figures 2.7(a) and 2.7(b), respectively. The dashed lines in all the figures indicate the instants at which the maneuver jumps from one mode to the next mode. At the *Pull* mode, we keep the last point of the first sub-trajectory, so the quadrotor can reach a hovering position over the load while its thrust is increased. From the plots in Figure 2.7(b), it is clear that the quadrotor is not directly above the load once the first sub-trajectory is over. Once the load is not touching the ground, the geometric controller continues doing a good job tracking the prescribed trajectories. The norm of the error $\mathbf{e}_l$ defined as the error between the load position and its desired

final position is shown in Figure 2.8(a). This error is calculated by $\mathbf{e}_l = \mathbf{r}_l - \mathbf{r}_{l_f}$, where $\mathbf{r}_{l_f}$ is defined in (2.7). Clearly, the goal of lifting the load to the desired altitude h is successfully accomplished. The velocity of the load is illustrated in Figure 2.8(b).



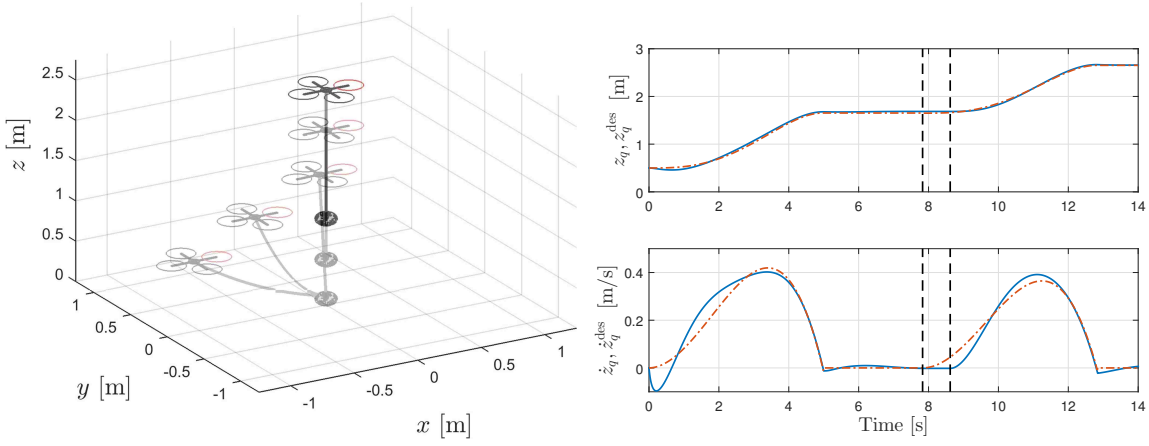
(a) (b)

Figure 2.7: (a) Reference trajectory, position (top) and velocity (bottom). (b) Actual trajectory described by the quadrotor, position (top) and velocity (bottom). The dashed vertical lines highlight the time instants at when the system transitions from *Setup* to *Pull* and from *Pull* to *Raise*.



(a) (b)

Figure 2.8: (a) Norm of the load position error with respect to the desired final position of the load. (b) Velocity of the load.



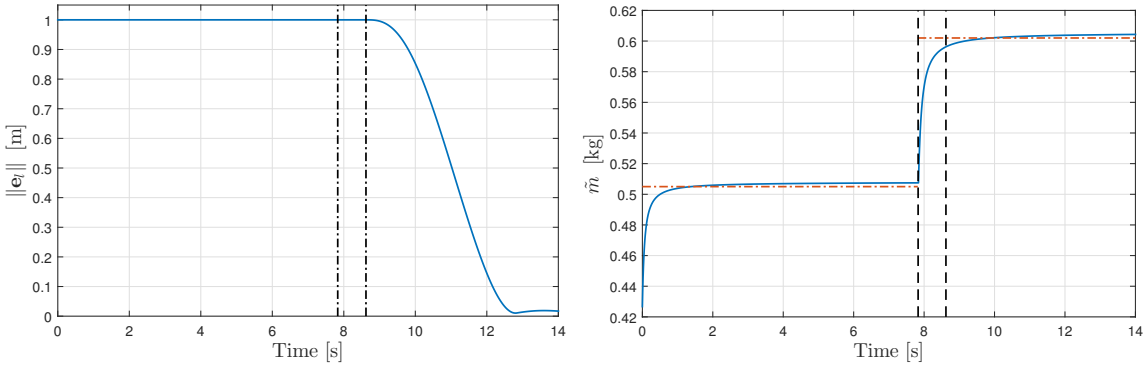
(a) (b)

Figure 2.9: (a) A composite image of the execution of the lift maneuver. (b) z-axis position (Top) and velocity (Bottom) for the quadrotor. The reference trajectory is in red while the actual trajectory is in blue. The dashed vertical lines highlight the time instants at when the system transitions from *Setup* to *Pull* and from *Pull* to *Raise*.

2.5.2 3-D Case

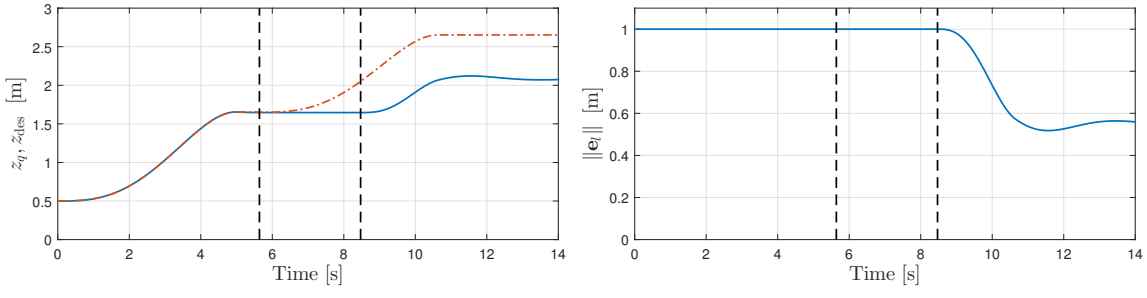
As in the previous case, we generate first the trajectory segments using (2.30)-(2.31) at the beginning of the simulation. We implement the complete control scheme depicted in Figure 2.4(a), so we include for this case the estimation block of the load mass. When we run the simulation, we use the first trajectory segment storing its last point until condition (2.16) holds. In this way, we allow for the quadrotor to reach the hovering position over the load. Then, we use the second segment for the rest of the simulation. The total time to accomplish the lift maneuver is around of 14 s.

Figure 2.9(a) shows a series of snapshots of the quadrotor-load system during the lift maneuver. In Figure 2.9(b), we show the results for the position and velocity for the quadrotor in the z-axis. Similar results are obtained for the other axes. The black dot lines indicate the time at which the maneuver jumps from one mode to the



(a) (b)

Figure 2.10: (a) Norm of the load position error $\mathbf{e}_l = \mathbf{r}_l - \mathbf{r}_{l_f}$. (b) Estimated mass (blue) and true mass (red) during the lift maneuver. The dashed vertical lines highlight the time instants at when the system transitions from *Setup* to *Pull* and from *Pull* to *Raise*.

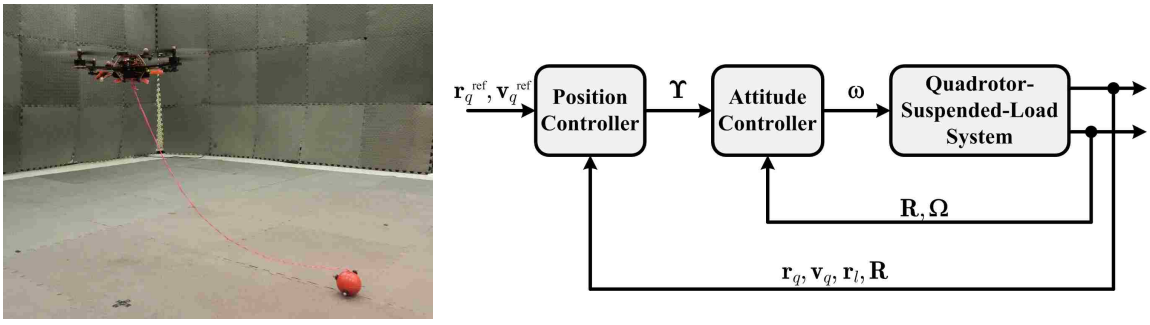


(a) (b)

Figure 2.11: Results without using the mass estimator. (a) z-axis position for the quadrotor. The desired trajectory is in red and the actual trajectory is in blue. (b) Norm of the load position error.

next mode. The transition between *Setup* to *Pull* occurs at 7.83 s and from *Pull* to *Raise* at 8.62 s. The norm of the error of the load position with respect to the goal $\|\mathbf{e}_l\|$ is shown in Figure 2.10(a). Since the mass of the system changes during the lift maneuver, we use the BGF method to estimate it during the whole maneuver. The estimated mass during the simulation is depicted in Figure 2.10(b). The red dashed

lines represent the true value. The simulation results show a good response of the estimator in tracking the change of the system mass. This helps to the geometric controller to track the prescribed trajectory, so it is possible to accomplish the goal of lifting the cable-suspended load until it is at the desired altitude h. To validate the mass estimator, we used our control architecture but without using the BGF estimator, *i.e.*, assuming that the mass of the system is always equal to the mass of the quadrotor. In this case, the aerial vehicle fails to follow the desired trajectory and then the goal of lifting the load to the desired h is not achieved at least in the time taken when the estimator is employed, see Figure 2.11.



(a) (b)

Figure 2.12: (a) The quadrotor with the cable-suspended payload employed for experimental verification. (b) Block diagram of the cascade control loops: inner attitude control and outer position control.

2.6 Experimental Verification

The quadrotor UAV that we use for experimental validation is the "AscTec Hummingbird" [77], see Figure 2.12(a). This aerial vehicle is equipped with linear acceleration sensors, gyroscopes measuring the angular velocities, a triple-axial compass module, motor drivers, and a flight control unit (FCU), the AscTec Autopilot [78]. This FCU reads sensor data, computes angular velocities and angles in all axes (roll,

pitch and yaw), and runs an attitude controller at a rate of 1 kHz sending the desired speed for each motor to the respective driver. Furthermore, the FCU is designed to receive attitude (roll and pitch angles), yaw-rate, and thrust commands through a wireless serial link which enables the autonomous control of the quadrotor. In fact, this attitude controller has been extensively tested in a variety of applications [78–80].

2.6.1 Cascade Controller

We use a cascade control structure that is shown in Figure 2.12(b). As inner loop, the attitude controller provided in the FCU is employed whereas that the outer loop is the position controller. The input commands for the inner loop are desired roll ϕ and pitch θ angles, desired yaw rate $\dot{\psi}$, and desired thrust F. We denote this control input as $\boldsymbol{\Upsilon} = [\phi \ \theta \ \dot{\psi} \ F]^T$. The output of the attitude controller are the commanded rotational velocities of the four rotors denoted as $\boldsymbol{\omega} = [\omega_1 \ \omega_2 \ \omega_3 \ \omega_4]^T$ in Figure 2.12(b). This attitude control loop delivered with the FCU is a black box for the user, so it is not focus of this paper. Please refer to [78, 79] for a complete discussion about this controller.

The position control loop is implemented by applying nonlinear dynamic inversion [81, 82]. Having an adequate knowledge of the plant dynamics, this method transforms the nonlinear system into a linear system without any simplification through suitable control inputs. As a result, standard linear controllers can then be applied. This also aligns well with the differential flatness property shown in Lemma 2.1. We can perform linear control strategies, like a PD controller, by choosing pseudo control commands in the space of the flat outputs and their derivatives, and finally turning those into desired input commands [80].

Based on the vector fields defined for each state of the hybrid system $\mathscr{H}$ (Section 2.3), the translation dynamics of the quadrotor can be expressed as

$$\mathbf{a}_q = \frac{1}{m_q} F \mathbf{R} \mathbf{e}_3 - g \mathbf{e}_3 - \frac{1}{m_q} T \boldsymbol{\mu}, \tag{2.46}$$

where $\mathbf{a}_q$ is the acceleration of the quadrotor, *i.e.*, $\mathbf{a}_q = \dot{\mathbf{v}}_q = \ddot{\mathbf{r}}_q$, and

$$T = \begin{cases} 0 & \text{for the } \textit{Setup} \text{ mode,} \\ \|F\mathbf{R}\mathbf{e}_3 - m_q g\mathbf{e}_3\| & \text{for the } \textit{Pull} \text{ mode,} \\ \|m_l\dot{\mathbf{v}}_l + m_l g\mathbf{e}_3\| & \text{for the } \textit{Raise} \text{ mode.} \end{cases} \tag{2.47}$$

In (2.46), the attitude angles ϕ, θ, and ψ are encoded in $\mathbf{R}$ which is the rotation matrix from the body-fixed frame $\{\mathcal{B}\}$ and the inertial frame $\{\mathcal{W}\}$. The rotation sequence $z-x-y$ is generally used to model this rotation [67, 70], so the rotation matrix is given by

$$\mathbf{R} = \begin{pmatrix} c_\psi c_\theta - s_\phi s_\psi s_\theta & -c_\phi s_\psi & c_\psi s_\theta + c_\theta s_\phi s_\psi \\ c_\theta s_\psi + c_\psi s_\phi s_\theta & c_\phi c_\psi & s_\psi s_\theta - c_\psi c_\theta s_\phi \\ -c_\phi s_\theta & s_\phi & c_\phi c_\theta \end{pmatrix}, \tag{2.48}$$

where c_α and s_α are shorthand forms for $\cos(\alpha)$ and $\sin(\alpha)$, respectively. As we indicated in Section 2.4.1, a change in the yaw angle ψ does not have any effect on lifting the load since it is attached at the CoG of the quadrotor. Therefore, we assume that this angle is kept all the time equal to zero, *i.e.*, $\psi = 0^\circ$ and $\dot{\psi} = 0^\circ/s$. Replacing (2.48) in (2.46) and making $\psi = 0^\circ$, $\mathbf{a}_q = [\ddot{x}_q \;\; \ddot{y}_q \;\; \ddot{z}_q]^T$, and $T\boldsymbol{\mu} = [\tau_z \;\; \tau_y \;\; \tau_z]^T$ yields that

$$m_q \begin{bmatrix} \ddot{x}_q \\ \ddot{y}_q \\ \ddot{z}_q \end{bmatrix} = \begin{bmatrix} Fs\theta - \tau_x \\ -Fc\theta s\phi - \tau_y \\ Fc\theta c\phi - m_q g - \tau_z \end{bmatrix}. \tag{2.49}$$

Solving (2.49) for the controls of the system θ, ϕ and F, we get

$$\begin{aligned} \theta &= \arcsin \frac{m_q\ddot{x}_q + \tau_x}{F}, \\ \phi &= -\arctan \frac{m_q\ddot{y}_q + \tau_y}{m_q\ddot{z}_q + m_q g + \tau_z}, \\ F &= \sqrt{f_x^2 + f_y^2 + f_z^2}. \end{aligned} \tag{2.50}$$

Here, $f_x = m_q\ddot{x}_q + \tau_x$, $f_y = m_q\ddot{y}_q + \tau_y$, and $f_z = m_q\ddot{z}_q + m_q g + \tau_z$. From (2.50), we can find the control input $\boldsymbol{\Upsilon} = [\phi \;\; \theta \;\; 0 \;\; F]^T$ for the inner attitude loop based on the

desired acceleration for the quadrotor $\mathbf{a}_q = [\ddot{x}_q \ \ddot{y}_q \ \ddot{z}_q]^T$. Consequently, we take $\mathbf{a}_q$ as our pseudo control input. Since we can only command the second derivative of the quadrotor's position, a reference trajectory to track has to be sufficiently smooth. We achieve this in Section 2.4.1 by minimizing the jerk, so we guarantee that the third derivative of the quadrotor's reference position exists. We compute the pseudo control input by the following linear error controller

$$\mathbf{a}_q = \mathbf{a}_q^{\text{ref}} + \mathbf{K}_{\text{v}}(\mathbf{v}_q^{\text{ref}} - \mathbf{v}_q) + \mathbf{K}_{\text{p}}(\mathbf{r}_q^{\text{ref}} - \mathbf{r}_q), \tag{2.51}$$

with $\mathbf{K}_{\text{v}}$ and $\mathbf{K}_{\text{p}}$ being diagonal gain matrices. According to (2.51), position and speed control are performed by the outer control loop.

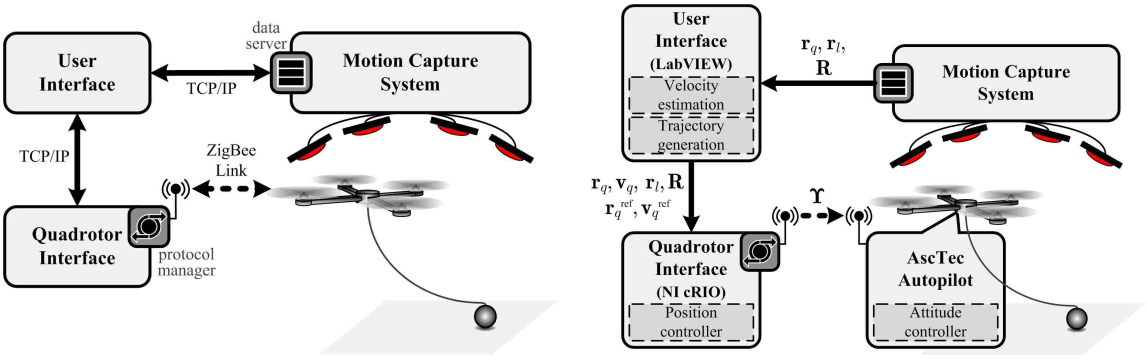


Figure 2.13: (a) The system architecture and its communication links at the MARHES Lab. (b) General structure of the system architecture components as used for controlling the aerial vehicle.

2.6.2 System Setup

To validate the proposed method for lifting a cable-suspended load by a quadrotor, we conducted a series of experiments using an AscTec Hummingbird quadrotor [77] that is part of the robotic testbed of the MARHES Lab[1] at the University of New Mexico

[1] http://marhes.unm.edu

(UNM). The quadrotor with the attached cable-suspended load is shown in Figure 2.12(a). The Hummingbird quadrotor is 0.54 m in diameter, weighs approximately 500 g including its battery, and has a maximum payload of 200 g. The load is a ball with 0.076 m in diameter and weighs 178 g. This load is suspended from a 1-meter-long cable.

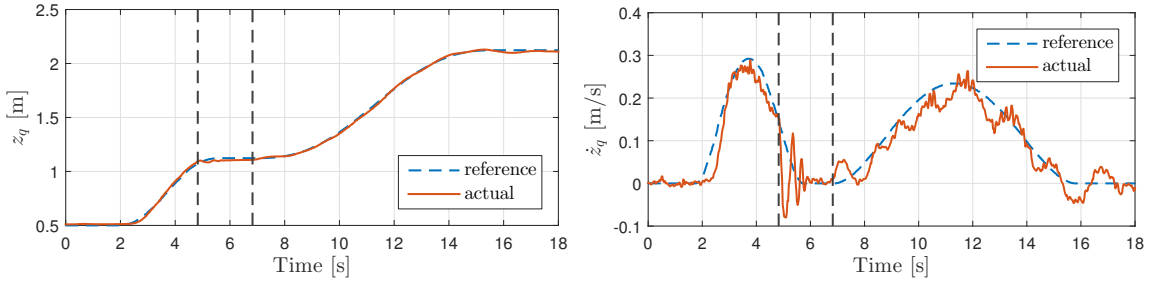
The system architecture implemented at the MARHES Lab to perform the experimental tests is illustrated in Figure 2.13(a). This figure also shows the communication links between the system components. The attitude and position of the aerial vehicle and the load are provided by a motion capture system with millimeter accuracy running at 100Hz. The entire control application is implemented in LabVIEW where we created two programs: User Interface and Quadrotor Interface. The first one runs on a Windows-based computer while the second is deployed in a National Instrument (NI) CompactRIO (cRIO) real-time controller [83]. The general structure of these two interfaces is shown in Figure 2.13(b). The arrows in this figure illustrate the flow of information to implement the cascade control scheme depicted in Figure 2.12(b).

The User Interface program acquires the pose data of the quadrotor and the position data of the suspended load, applies the numeric differentiation algorithm detailed in [84] for velocity estimation, and generates the lift trajectory using the methodology explained in Section 2.4.1. The Quadrotor Interface program reads the actual position, velocity and attitude data from the User Interface program as well as the generated reference trajectory (position and velocity). It filters high-frequency noise from the actual values by using a low-pass fifth-order finite impulse response (FIR) filter. Subsequently, it computes the input commands for the onboard attitude controller according to the control design detailed in Section 2.6.1, and then these commands are transmitted to the quadrotor. The User Interface and the Quadrotor Interface are executed at 1 kHz since we employed the LabVIEW Real-Time (RT) Module to implement them. This module compiles and optimizes the LabVIEW graphical code for executing RT control applications [85]. More details

about the robotic testbed and the control architecture at the MARHES Lab are presented in [14, 86].

2.6.3 Results

We report three sets of experiments to demonstrate the validity of the proposed method. A video of these experiments can be found in [87].



(a) (b)

Figure 2.14: Z-axis position **a** and velocity **b** for the quadrotor during the first experiment. In this experiment, the aerial vehicle is commanded to lift the cable-suspended load to a desired height $h = 1$ m. Approximately at 4.8 s, the system jumps from *Setup* to *Pull* and approximately at 6.8 s from *Pull* to *Raise*. These time instants are highlighted by the dashed vertical lines.

In the first experiment, we run the lift maneuver of the load to the desired height of 1 m. Based on the initial positions of the quadrotor and the load, the reference lift trajectory is generated and then we command the aerial vehicle to execute it. The initial positions for the CoM of the load and the aerial robot are $\mathbf{r}_{l_0} = [0\ 0\ 0.038]^T$ m and $\mathbf{r}_{q_0} = [-0.5\ 0.5\ 0.5]^T$ m, respectively. The time allowed for the first trajectory segment (from $\mathbf{r}_{q_0}$ to $\mathbf{r}_{\text{pull}}$) is 4 s and for the second segment (from $\mathbf{r}_{\text{pull}}$ to $\mathbf{r}_{q_f}$) is 8 s, with a rest time of 1 s between segments. Figure 2.14 shows the Z-axis trajectory tracking data for the quadrotor, position z_q and velocity $\dot{z}_q$. Approximately at 4.8 s, the guard condition (2.10) is satisfied making the system jump from *Setup* to *Pull*.

We draw a dashed vertical line at this time instant in Figure 2.14. Subsequently, the system switches between *Pull* and *Setup* having as result a series of spikes in $\dot{z}_q$ before the system stays at the *Pull* mode, see Figure 2.14(b). The system then jumps from *Pull* to *Raise* since the guard condition $T > m_q g$ holds. This occurs when the elapsed time is approximately 6.8 s. Similarly as before, this time instant is also pointed up by a dashed vertical line in Figure 2.14.

In order to verify the performance of executing the lift maneuver, we compute the following errors:

- the quadrotor position error $\mathbf{e}_{\mathbf{r}_q} = \mathbf{r}_q^{\text{ref}} - \mathbf{r}_q$,
- the quadrotor velocity error $\mathbf{e}_{\mathbf{v}_q} = \mathbf{v}_q^{\text{ref}} - \mathbf{v}_q$, and
- the load position error $\mathbf{e}_{\mathbf{r}_l} = \mathbf{r}_{l_f} - \mathbf{r}_l$.

The first two errors show the trajectory tracking control performance while the last one is the error of the load position with respect to its desired final height. These errors for the first experiment are shown in Figure 2.15. The time instants at when the system jumps from *Setup* to *Pull* and from *Pull* to *Raise* indicated in Figure 2.14 are also underlined in each plot of this figure. The X-axis position error for the quadrotor is less than 0.05 m as well as for the Y-axis, but this error is less than 0.03 m for the Z-axis, see Figure 2.15(a). For the velocity errors, Figure 2.15(b), they are less than 0.09 m/s for the X and Y axes. This is also the case almost all the time for the Z-axis component except at the instant that the system switches between *Setup* and *Pull*. Because the Z-axis velocity experiences a series of spikes during these jumps, see Figure 2.14(b), the error increases having as maximum 0.18 m/s. The load position errors e_{x_l} for the X-axis and e_{y_l} for the Y-axis are both less than 0.1 m, see Figure 2.15(c). Meanwhile, e_{z_l}, the load position error for Z-axis, decreases after the system reaches the *Raise* mode and it is less than 0.05 m at the end of the lift maneuver.

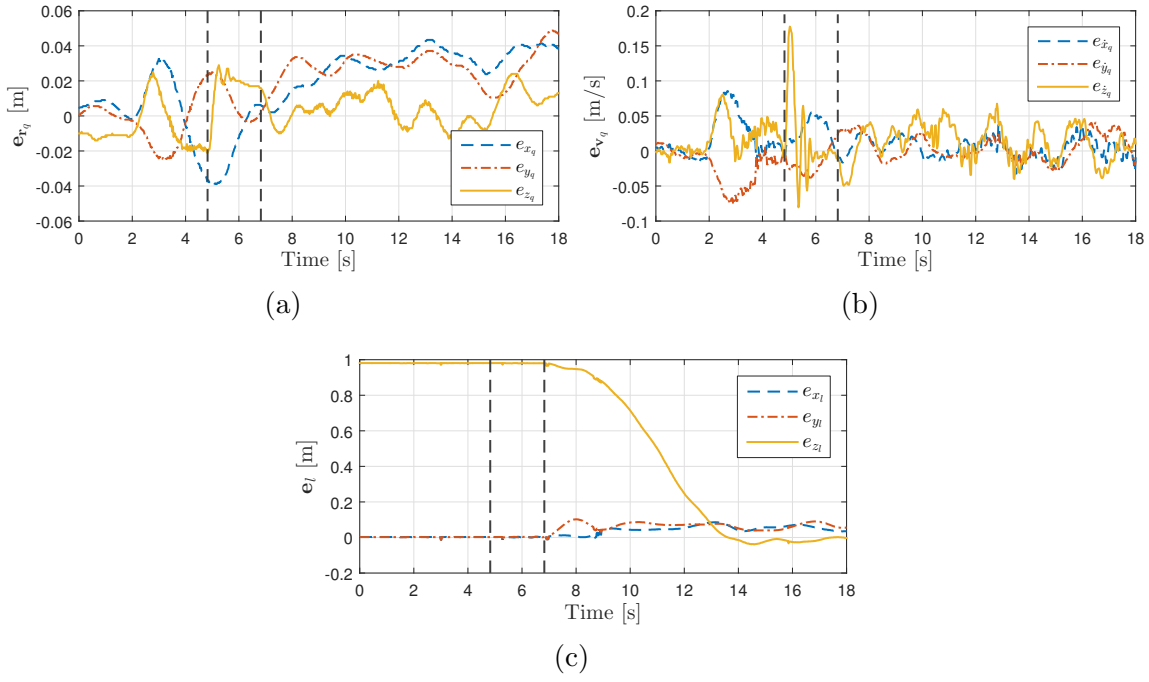


Figure 2.15: Performance data results for the first experiment. Trajectory tracking errors for the quadrotor, (a) position and (b) velocity errors. (c) Position error of the cable-suspended load with respect to its desired final position $\mathbf{r}_{l_f} = [0\ 0\ 1]^T$ m. The dashed vertical lines highlight the time instants at when the system transitions from *Setup* to *Pull* and from *Pull* to *Raise*.

In the second experiment, we study the effect of not considering the hybrid nature of the system for control purposes. Notice that the controller designed in Section 2.6.1 has a switching behavior since the cable tension T used to find the pseudo control input $\mathbf{a}_q$ has a different value depending on the mode of the lift maneuver. Indeed, T is given by (2.47) for the proposed controller. We re-ran the experiment of lifting the load to 1 m, but we set $T = m_l g$ for computing $\mathbf{a}_q$ during the entire maneuver. Thus, we do not use (2.47) for this second experiment. Figures 2.16(a), 2.16(b), and 2.17(a) show the results for z_q, $\dot{z}_q$, and $\mathbf{e}_{\mathbf{r}_l}$, respectively. Similarly as for the first experiment, the system jumps from *Setup* to *Pull* approximately at 4.8 s. However, the transition from *Pull* to *Raise* occurs approximately at 10 s.

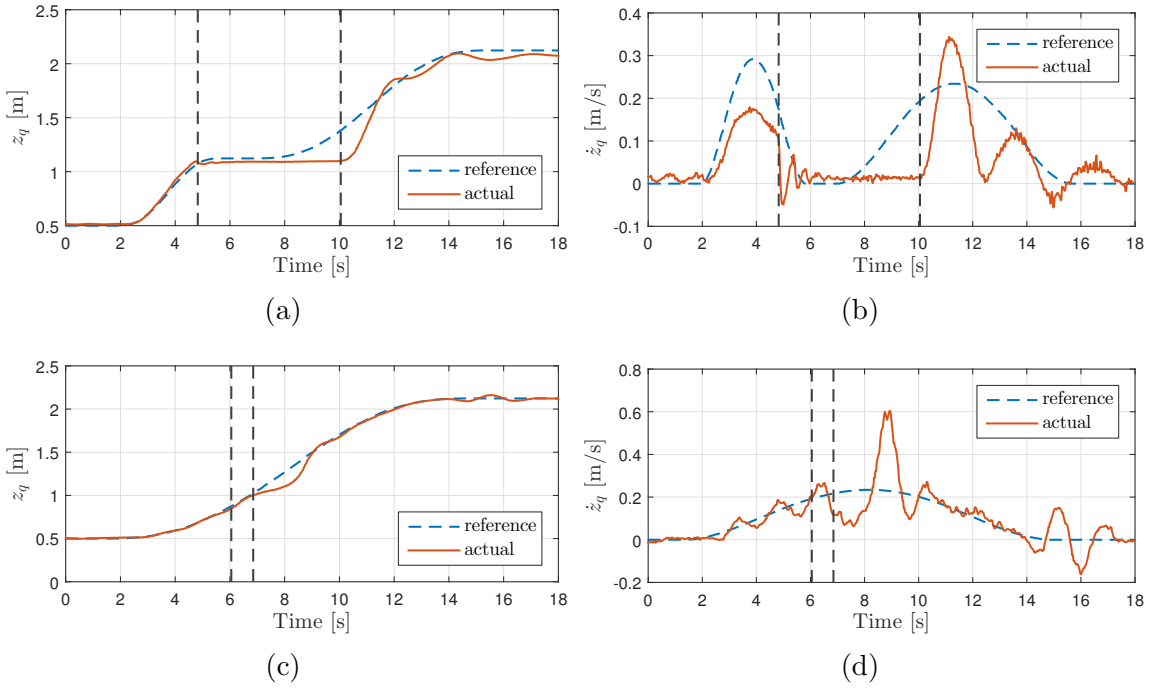
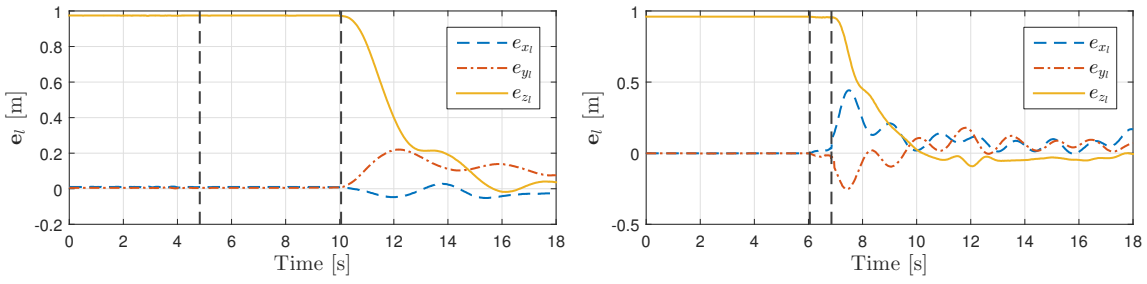


Figure 2.16: Experimental results for a no switching controller. (a) Desired and actual Z-axis position for the quadrotor. (b) Desired and actual Z-axis velocity for the quadrotor. Experimental results for a trajectory generated without considering the via point $\mathbf{r}_{\text{pull}}$. (c) Desired and actual Z-axis position for the quadrotor. (d) Desired and actual Z-axis velocity for the quadrotor. The dashed vertical lines highlight the time instants at when the system transition from *Setup* to *Pull* and from *Pull* to *Raise*.

Indeed, there is a considerable delay on continuing tracking the trajectory (position and velocity) as can be seen in Figures 2.16(a) and 2.16(b). This delay causes a transient, especially for $\dot{z}_q$. For the load position error illustrated in Figure 2.17(a), it is less than 0.1 m and 0.2 m for the X and Y-component, respectively. Meanwhile, it is less than 0.1 m at the end of the maneuver for the Z-axis. This experiment shows that even though the maneuver can be executed, the performance diminishes when the switching behavior of the quadrotor-suspended-load system is not under consideration for controlling the aerial vehicle.

For the third experiment, we generate the lift trajectory without using the via



(a) (b)

Figure 2.17: Position error for the load. (a) Experimental results for a no switching controller. (b) Experimental results for a trajectory generated without considering the via point $\mathbf{r}_{\text{pull}}$. The dashed vertical lines highlight the time instants at when the system transition from *Setup* to *Pull* and from *Pull* to *Raise*.

point $\mathbf{r}_{\text{pull}}$. Therefore, the trajectory just has one segment which goes from $\mathbf{r}_{q_0}$ to $\mathbf{r}_{q_f}$ with a total duration time of 12 s. We use the proposed controller without any modification as it was the case for the second experiment. The results for this case are shown in Figures 2.16(c), 2.16(d), and 2.17(b) for z_q, $\dot{z}_q$, and $\mathbf{e}_{\mathbf{r}_l}$, respectively. The system goes from *Setup* to *Pull* approximately at 6 s, while it goes from *Pull* to *Raise* approximately at 6.8 s. There is a small delay on tracking the Z reference position after the load is starting to be lifted, see Figure 2.16(c). This delay creates an oscillation in the Z-axis velocity component reaching a maximum of 0.6 m/s, see Figure 2.16(d). For the position error of the load, Figure 2.17(b), there is a short swing in the X and Y components right after the quadrotor starts to lift the load. As a result, the error reaches a maximum of 0.5 m and 0.25 m for e_{x_l} and e_{y_l}, respectively. At the end of the maneuver, e_{z_l} is less than 0.1 m. Although the load lift starts at a similar time instant that the one for the first experiment, the errors for the load position are higher when the via point $\mathbf{r}_{\text{pull}}$ is not considered for generating the trajectory to lift the cable-suspended load. In the last part of the video available in [77], we show side by side the three experiments for comparison purposes.

2.7 Conclusions

Aerial transportation of cable-suspended loads is an important application of flying robots. Before transporting the cargo, a critical maneuver is to lift the payload from the ground. In this chapter, we introduced a novel methodology to perform this maneuver using a quadrotor. We designed a hybrid system that captures specific operating regimes of the quadrotor-suspended-load system during the lift maneuver. In particular, we proved that this system is a differentially-flat hybrid system. Taking advantage of this property, we generated a dynamically feasible trajectory based on the discrete states of the hybrid system. By combining geometric control with least-squares parameter estimation, our trajectory tracking controller was able to perform the lift maneuver even without knowing the payload mass. Numerical simulations helped to validate the proposed approach. Furthermore, we presented experimental results illustrating the effectiveness of our method for the case when the payload mass is known. Significant improvement in tracking performance and reducing the load position error with respect to the final desired height were achieved when the hybrid modes were considered for generating the trajectory and controlling the aerial vehicle.

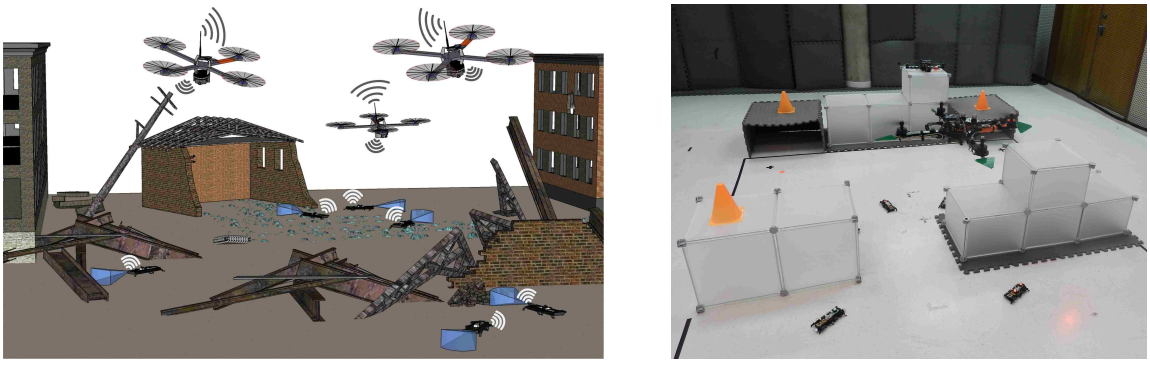
Important topics for future work include the practical implementation of the system mass estimator in order to execute the maneuver even without knowing the payload mass, and the extension of the hybrid model to address the maneuver of placing the cable-suspended load over the ground. This step is also critical in aerial cargo transportation. Another path to take is to study and implement cooperative lifting using multiple quadrotors. Non-uniform and heavier cable-suspended loads can be manipulated since each aerial vehicle can apply a tension force to different attachment points in the load.

Chapter 3

Coordinated Control Under Connectivity Constraints

Coordinated control refers to the ability of a team of robots to work together in order to accomplish a task. The mission scenario that we envision is shown in Figure 3.1(a) where a group of robots senses a target area in a disaster area with collapsed structures. In these situations, the use of heterogeneous systems made up of aerial and ground robotic vehicles would maximize the probability to efficiently and successfully accomplish the mission. A reliable wireless connectivity is an important factor to be considered when dealing with multi-robot systems specially for control coordination. Due to several limitations in the communication channel, especially when the transmission is through the air medium, complications such as shadow effects and secondary reflections arise. These phenomena create a variety of constraints on the possible relative positions of the agents. Thus, we are interested in developing strategies to enhance connectivity of the robotic network and a fixed base station while a given number of targets are sensed. In Section 3.1, we describe a target sensing algorithm for a group of ground robots while the network connectivity is guaranteed by an aerial relay agent which is better equipped to communicate over longer distances. Based on the results for the case of assuming one aerial relay, we

extend our approach in Section 3.2 to the case of multiple aerial relays.



(a) (b)

Figure 3.1: (a) A group of aerial relays cooperating with a group of mobile ground sensors to explore a disaster area with collapsed structures. (b) A heterogeneous robotic network made by a quadrotor equipped with four directional antennas, one in each arm, and three OctoRoACHes moving around a cluttered environment.

3.1 Connectivity Maintenance - One Aerial Communication Relay

We focus on a multi-robotic network made by aerial relays and ground sensors that is deployed to sense areas of interest in an environment populated with obstacles. We use potential field methods to coordinate the ground mobile sensors and to control the relative location of the flying relay. The work presented in this section is based on our paper published in [20] in collaboration with Dr. Silvia Ferrari's group at Duke University.

Potential field methods are robot motion planning algorithms that control the robot movement based on the gradient field of a potential function [88]. These methods were originally developed as an on-line collision avoidance approach, applicable when a robot does not have a prior model of the obstacles, but senses them during

motion execution [89]. The main idea of most proposed potential functions [88, 89] is that a robot should be attracted toward its target configuration, while being repulsed by possible obstacles. Therefore, the obstacle and target configuration are considered as sources to construct a potential function U. In general, U consists of two components: an attractive potential U_{att} generated, for example, by the target configuration and a repulsive potential U_{rep} generated, for example, by the obstacles. Thus, the total potential is given by

$$U(\mathbf{x}) = U_{att}(\mathbf{x}) + U_{rep}(\mathbf{x}),$$

where $\mathbf{x} = [x_1 \ x_2 \ ... \ x_n]^T \in \mathbb{R}^n$ is the configuration state of the robot. The force applied on the robot is proportional to the negative gradient of U

$$\nabla U(\mathbf{x}) = \left[\frac{\partial U((x))}{\partial x_1} \ \frac{\partial U((x))}{\partial x_2} \ ... \ \frac{\partial U((x))}{\partial x_n} \right]^T,$$

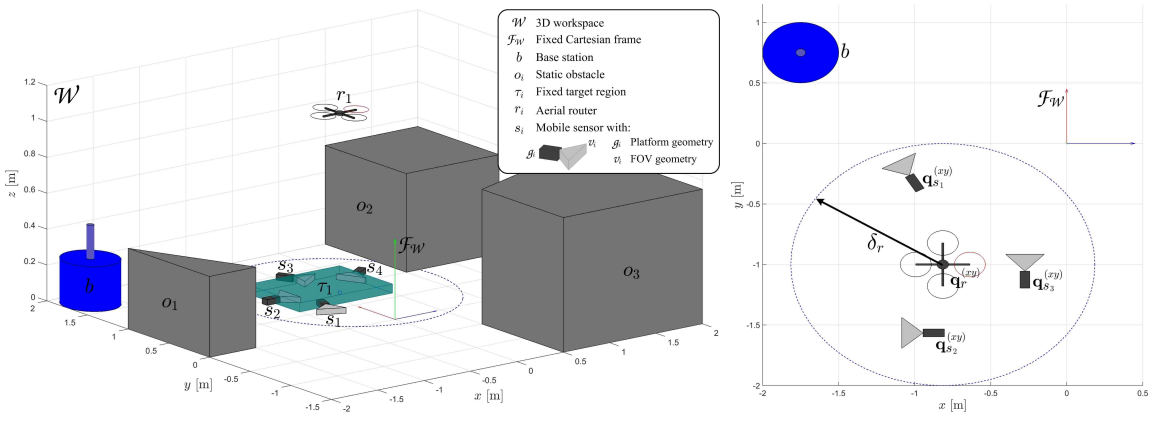
and it is used to design a controller for the robot movement. As in [88, 89], an attractive potential can be represented as

$$U_{\text{att}}(\mathbf{x}) = \frac{1}{2}\eta_{\text{att}}\varrho_t^2(\mathbf{x}), \tag{3.1}$$

where η_{att} is a scaling factor and $\varrho_t(\mathbf{x})$ is the Euclidean distance between the robot and the target configuration. Meanwhile, a repulsive potential can be given by

$$U_{\text{rep}}(\mathbf{x}) = \begin{cases} \frac{1}{2}\eta_{\text{rep}}\left(\frac{1}{\varrho_o(\mathbf{x})} - \frac{1}{d_0}\right)^2 & \text{if } \varrho_o(\mathbf{x}) \leq d_0, \\ 0 & \text{if } \varrho_o(\mathbf{x}) > d_0, \end{cases} \tag{3.2}$$

where η_{rep} is a scaling factor, $\varrho_o(\mathbf{x})$ is the Euclidean distance between the robot and the nearest obstacle, and d_o is the distance of influence of the obstacles. In particular, an on-line potential field method essentially acts as a descent optimization procedure, so it may get stuck at a local minimum other than the goal configuration. However, the combination of potential field methods with graph searching techniques has demonstrated to be a valid approach such that a robot escapes the local minimum [88, 90].



(a) (b)

Figure 3.2: (a) 3-D simulation environment with a legend describing its components. (b) 2-D representation of the communication constraints specified by Definitions 3.1, 3.2 and 3.3.

3.1.1 Problem Formulation

Let $\mathcal{A}$ be a set of robots made of two types of agents: L mobile ground sensors and one aerial relay denoted by r. The mobile sensors form the set $\mathcal{S} = \{s_1, ..., s_N\}$ and then $r \notin \mathcal{S}$ and $\mathcal{A} = \mathcal{S} \cup \{r\}$. $\mathcal{A}$ operates in $\mathcal{W} \subset \mathbb{R}^3$ which is a compact subset of a three-dimensional Euclidean space where there is a fixed base station b. In $\mathcal{W}$, there are L convex obstacles grouped in the set $\mathcal{O} = \{o_1, ..., o_L\}$ and there are K static rigid targets that forms the set $\mathcal{T} = \{\tau_1, ..., \tau_K\}$ such that $\mathcal{O} \cap \mathcal{T} = \emptyset$. We denote $I_\mathcal{O}$ and $I_\mathcal{T}$ as the index sets of $\mathcal{O}$ and $\mathcal{T}$, respectively. For all $j \in I_\mathcal{T}$, we call ϕ_{τ_j} to the center of τ_j. Meanwhile embedded in $\mathcal{W}$, there is a fixed Cartesian frame $\mathcal{F}_\mathcal{W}$ with origin $\phi_\mathcal{W}$. $\mathcal{F}_\mathcal{W}$ allows us to describe the position and orientation of the agents, objects and targets in $\mathcal{W}$. For instance, $\forall j \in I_\mathcal{O}$, every point of o_j has a fixed position with respect to $\mathcal{F}_\mathcal{W}$ because o_j is considered rigid and fixed in $\mathcal{W}$. Figure 3.2(a) shows the 3D environment developed for simulation purposes. Also, we include a legend on the right top corner of this figure to facilitate the reference of the different assumptions.

Motion Dynamics

Let $I_{\mathcal{S}}$ be the index set of $\mathcal{S}$, then $I_{\mathcal{S}} = \{1, ..., N\}$. We consider $\forall i \in I_{\mathcal{S}}$ that s_i has a platform geometry $\mathcal{g}_i \subset \mathbb{R}^3$ and a field-of-view (FOV) geometry $\mathcal{v}_i \subset \mathbb{R}^3$ from which the robot can obtain sensor measurements [88,91], see Figure 3.2(a). We also assume $\mathcal{g}_i = \mathcal{g}_j$ and $\mathcal{v}_i = \mathcal{v}_j$ $\forall i, j \in I_{\mathcal{S}}$. Furthermore, $\mathcal{g}_i$ and $\mathcal{v}_i$ are both rigid and $\mathcal{v}_i$ has a fixed position and orientation with respect to $\mathcal{g}_i$. We say that $\forall i \in I_{\mathcal{S}}$ and $\forall j \in I_{\mathcal{T}}$ the sensor s_i gets measurements of the target τ_j when $\mathcal{v}_i \cap \tau_j \neq \emptyset$. In fact, we ensure this last condition when $s_i \in \mathcal{B}\left(\emptyset_{\tau_j}, d_{o_{sens}}\right)$ where $d_{o_{sens}}$ is the minimum distance to the target τ_j for getting measurements.[1] In addition, we suppose that $\mathcal{F}_{\mathcal{g}_i}$ is a Cartesian frame embedded in $\mathcal{g}_i$ with origin $\emptyset_{\mathcal{g}_i}$. Now let (x_{s_i}, y_{s_i}) be the position of $\mathcal{F}_{\mathcal{g}_i}$ respect to $\mathcal{F}_{\mathcal{W}}$ and let θ_i be the orientation of $\mathcal{F}_{\mathcal{g}_i}$ respect to $\mathcal{F}_{\mathcal{W}}$. Also, let v_{s_i} be the linear velocity of the i^{th} mobile sensor. Then, we define $\forall i \in I_{\mathcal{S}}$ the state vector of the sensor s_i as $\mathbf{q}_{s_i} = [x_{s_i} \quad y_{s_i} \quad v_{s_i} \quad \theta_i]^T$. In other words, we consider that every s_i is moving just on the xy plane with θ_i as its heading angle. Notice that $\mathbf{q}_{s_i}$ can be used to determine the position and orientation of $\mathcal{g}_i$ and $\mathcal{v}_i$ respect to $\mathcal{F}_{\mathcal{W}}$. The state vector of each mobile sensor, $\mathbf{q}_{s_i}$, must also satisfy the sensor dynamics that are given by the unicycle model,

$$\begin{cases} \dot{x}_{s_i} = v_{s_i} \cos\theta_i, \\ \dot{y}_{s_i} = v_{s_i} \sin\theta_i, \\ \dot{v}_{s_i} = a_i, \\ \dot{\theta}_i = \omega_i, \end{cases} \tag{3.3}$$

where a_i and ω_i are the i^{th} mobile sensor's linear acceleration and angular velocity, respectively. Thus, the control vector for the s_i sensor is $\mathbf{u}_{s_i} = [a_i \quad \omega_i]^T \in \mathbb{R}^2$.

On the other hand, for the mobile relay r, the range of communication coverage is denoted as δ_r. We also assume that r moves at a safe fixed height over the mobile sensors and over all the obstacles in $\mathcal{W}$. Furthermore, its motion dynamics are given

[1] We use $\mathcal{B}(\mathbf{q}, \delta)$ to denote the open ball of radius δ centered at $\mathbf{q}$.

by

$$\ddot{\mathbf{q}}_r = \mathbf{u}_r, \tag{3.4}$$

where $\mathbf{q}_r = [x_r \quad y_r \quad z_r]^T$ specifies the 3-D position of the flying relay with respect to $\mathcal{F}_{\mathcal{W}}$, while $\mathbf{u}_r \in \mathbb{R}^3$ is its acceleration control input.

Communication Links

For the next definitions, $\mathbf{q}^{(xy)}$ is the vector formed by having just the two first components of the vector $\mathbf{q}$, *e.g.*, $\mathbf{q}_{s_i}^{(xy)} = \begin{bmatrix} x_{s_i} & y_{s_i} \end{bmatrix}^T$. In our scenario, we assume there is a *point-to-point link* between the mobile relay r and the base station b at any time. Also, we suppose that r can manage at any time communication packets between any pair of sensors in $\mathcal{S}$ or between a sensor in $\mathcal{S}$ and b. We introduce the next two definitions.

Definition 3.1. *For all $i, j \in I_{\mathcal{S}}, i \neq j$, s_i has bidirectional communication with s_j if $\mathbf{q}_{s_i}^{(xy)}, \mathbf{q}_{s_j}^{(xy)} \in \mathcal{B}\left(\mathbf{q}_r^{(xy)}, \delta_r\right)$.*

Definition 3.2. *For every $i \in I_{\mathcal{S}}$, s_i has bidirectional communication with b if $\mathbf{q}_{s_i}^{(xy)} \in \mathcal{B}\left(\mathbf{q}_r^{(xy)}, \delta_r\right)$.*

Therefore, a mobile sensors can talk with one of its pairs just if both are within the ball of radius δ_r and centered at $\mathbf{q}_r^{(xy)}$. Similarly, a mobile sensor can receive/send information from/to the base station just if it is within the ball of radius δ_r and centered at $\mathbf{q}_r^{(xy)}$. It is clear if Definition 3.2 holds for every $i \in I_S$ then Definition 3.1 also holds. Consequently, we can combine both definitions in the following definition.

Definition 3.3. *If $\forall i \in I_{\mathcal{S}}, \mathbf{q}_{s_i}^{(xy)} \in \mathcal{B}\left(\mathbf{q}_r^{(xy)}, \delta_r\right)$ then s_i has bidirectional communication with any s_j where $j \in I_{\mathcal{S}}, i \neq j$, and s_i also has bidirectional communication with b.*

Notice that the set of robots $\mathcal{A}$ is an *heterogeneous robotic network* since its agents have different dynamics, communication ranges, and objectives. The ground sensors mission is mainly target sensing, while the aerial relay has to maintain the connectivity among the mobile sensors and the base station.

Under the previous assumptions, we are concerned with the following problem:

Problem 3.1. *A set of heterogeneous robots $\mathcal{A}$ formed by one mobile relay $\{r\}$ and N mobile sensors $\{s_1, ..., s_N\}$ must obtain measurements of K targets located in an obstacle populated environment such that $\mathcal{A}$ maintains inter-agent connectivity and also connectivity with a base station b.*

Since the set $\mathcal{A}$ works in a cluttered scenario, we have to add inter-robot collision prevention and obstacle avoidance to the objectives stated in the problem. Consequently, the problem considered in this paper aims to design a controller for the agents of the heterogeneous robotic network $\mathcal{A}$ such that they: (i) sense M static targets, (ii) keep inter-agent connectivity among the mobile sensors, (iii) keep mobile sensor and base station connectivity, (iv) avoid inter-agent collisions, and (v) avoid obstacle collisions. From our assumptions, the objectives (i) to (v) need to be considered for the design of the controller for the mobile sensors. On the contrary, the objectives (i), (iv) and (v) are not part of the controller design for the mobile relay r because we assume that it has only communication and not sensing capabilities, and it always flies at a safe height over the mobile sensors and obstacles. In the next section, the proposed controllers for the mobile sensors and for the mobile relay are described.

3.1.2 Coordinated Control

From the problem statement, we need to design two local controllers: one for the mobile sensors and one for the mobile relay. In this section, we develop these controllers. For the next definitions, $\|\cdot\|$ denotes the Euclidean norm.

Ground Sensor Controller

In this case, the objectives (i) to (v) have to be accomplished by the controller of each $s_i \in \mathcal{S}$. For the sensing objective (i), we consider that the set of mobile sensors $\mathcal{S}$ takes measurements of the M targets in $\mathcal{T} = \{\tau_1, ..., \tau_M\}$ in sequential order. This means the sensors first take measurements of τ_1, then of τ_2, and so on until $\mathcal{S}$ takes measurements of τ_M. For each s_i, we design its controller based on potential field methods. In particular, we take as base the potential functions given in (3.1) for an attractive potential and in (3.2) for a repulsive potential. We define $U_i(\mathbf{q}_{s_i})$, the potential function of the i^{th} mobile sensor with $i \in I_{\mathcal{S}}$ as

$$U_i(\mathbf{q}_{s_i}) = U_t(\mathbf{q}_{s_i}) + U_c(\mathbf{q}_{s_i}) + U_o(\mathbf{q}_{s_i}), \tag{3.5}$$

where $U_t(\mathbf{q}_{s_i})$ is the attractive potential of the j^{th} target with $j \in I_{\mathcal{T}}$, $U_c(\mathbf{q}_{s_i})$ is a combination of an attractive potential for keeping inter-sensor connectivity and a repulsive potential for avoiding inter-sensor collision, and $U_o(\mathbf{q}_{s_i})$ is a repulsive potential for obstacle avoidance.

The attractive potential $U_t(\mathbf{q}_{s_i})$ is given by

$$U_t(\mathbf{q}_{s_i}) = \begin{cases} \frac{1}{2}\eta_t \varrho_t^2(\mathbf{q}_{s_i}, \tau_j) & \text{if } \varrho_t(\mathbf{q}_{s_i}, \tau_j) \geq d_{o_{sens}}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where $d_{o_{sens}}$ is the required distance from s_i to τ_j to get sensor measurements, see Section 3.1.1, and $\varrho_t(\mathbf{q}_{s_i}, \tau_j)$ is the distance between the mobile sensor s_i and the center of target τ_j, *i.e.*, $\varrho_t(\mathbf{q}_{s_i}, \tau_j) = \varrho_t(\mathbf{q}_{s_i}, \emptyset_{\tau_j})$.

The potential $U_c(\mathbf{q}_{s_i})$ is defined as

$$U_c(\mathbf{q}_{s_i}) = \sum_{j=1, j\neq i}^{N} U(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) \tag{3.7}$$

with

$$U(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) = \begin{cases} \frac{1}{2}\eta_c \varrho_s^2(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) & \text{if } \varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) > \frac{\delta_r}{2}, \\ \frac{1}{2}\eta_c \left(\frac{1}{\varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j})} - \frac{1}{d_{o_{col}}}\right)^2 & \text{if } \varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) \leq d_{o_{col}}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.8}$$

where $\delta_r > d_{o_{col}}$, δ_r is the range of communication coverage, see Section 3.1.1, $d_{o_{col}}$ is the clearance inter-sensor distance, and $\varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) = \|\mathbf{q}_{s_i}^{(xy)} - \mathbf{q}_{s_j}^{(xy)}\|$.

The repulsive potential for obstacle avoidance $U_o(\mathbf{q}_{s_i})$ is given by

$$U_o(\mathbf{q}_{s_i}) = \sum_{j=1}^{L} U(\mathbf{q}_{s_i}, \boldsymbol{o}_j), \tag{3.9}$$

with

$$U(\mathbf{q}_{s_i}, \boldsymbol{o}_j) = \begin{cases} \frac{1}{2}\eta_{rep}\left(\frac{1}{\varrho_o(\mathbf{q}_{s_i}, \boldsymbol{o}_j)} - \frac{1}{d_{o_{obj}}}\right)^2 & \text{if } \varrho_o(\mathbf{q}_{s_i}, \boldsymbol{o}_j) \leq d_{o_{obj}}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.10}$$

where $\varrho_o(\mathbf{q}_{s_i}, \boldsymbol{o}_j)$ is the distance between s_i and the j^{th} obstacle, and $d_{o_{obj}}$ is the influence distance of the obstacles.

With respect to the scaling factors η_t, η_c, and η_o, we assign their values according to Algorithm 3.1. Using this algorithm, we adapt the relative importance of the potential functions $U_t(\mathbf{q}_{s_i})$, $U(\mathbf{q}_{s_i}, \mathbf{q}_{s_j})$, and $U(\mathbf{q}_{s_i}, \boldsymbol{o}_j)$ between each other at every step time. For example, if a mobile sensor s_i is far from a target, but there is more possibility of an inter-sensor collision than an obstacle collision, *i.e.*, $e_c > e_o > e_t$, then we give more priority to the potential function $U(\mathbf{q}_{s_i}, \mathbf{q}_{s_j})$ with Algorithm 3.1 in order to avoid the inter-sensor collision.

The gradient $U_i(\mathbf{q}_{s_i})$ for every $i \in I_{\mathcal{S}}$ is given by

$$\nabla U_i(\mathbf{q}_{s_i}) = \left[\frac{\partial U_i(\mathbf{q}_{s_i})}{\partial x_{s_i}} \quad \frac{\partial U_i(\mathbf{q}_{s_i})}{\partial y_{s_i}} \quad \frac{\partial U_i(\mathbf{q}_{s_i})}{\partial \theta_i}\right],$$

so the artificial force induced by the potential function is $\mathbf{F}_i(\mathbf{q}_{s_i}) = -\nabla U_i(\mathbf{q}_{s_i})$. From Section 3.1.1, the control vector for the s_i sensor is $\mathbf{u}_{s_i} = [a_i \quad \omega_i]$, so the potential-based control law is given by

$$\begin{aligned} a_i &= -[\cos\theta_i \quad \sin\theta_i \quad 0]\nabla U_i(\mathbf{q}_{s_i})^T - k_0 v_{s_i} \text{ and} \\ \omega_i &= k_1\left[\text{atan2}\left(\frac{\partial U_i(\mathbf{q}_{s_i})}{\partial x_{s_i}}, \frac{\partial U_i(\mathbf{q}_{s_i})}{\partial y_{s_i}}\right) - \theta_i\right], \end{aligned} \tag{3.11}$$

where k_0 and k_1 are positive constants.

Algorithm 3.1 Assignment of the scaling factors η_t, η_c and η_o

Require: : β_1, β_2, and β_3 positive scaling factors such that $\beta_1 > \beta_2 > \beta_3$

$$e_t = \begin{cases} |\varrho_t(\mathbf{q}_{s_i}, \tau_j) - d_{o_{sens}}| & \text{if } \varrho_t(\mathbf{q}_{s_i}, \tau_j) \geq d_{o_{sens}} \\ 0 & \text{otherwise} \end{cases}$$

$$e_c = \begin{cases} |\varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) - \frac{\delta_r}{2}| & \text{if } \varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) > \frac{\delta_r}{2} \\ |\varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) - d_{o_{col}}| & \text{if } \varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) \leq d_{o_{col}} \\ 0 & \text{otherwise} \end{cases}$$

$$e_o = \begin{cases} |\varrho_o(\mathbf{q}_{s_i}) - d_{o_{obj}}| & \text{if } \varrho_o(\mathbf{q}_{s_i}) \leq d_{o_{obj}} \\ 0 & \text{otherwise} \end{cases}$$

if $e_t \geq e_c$ and $e_t \geq e_o$ **then**
    $\eta_t = \beta_1$
    if $e_c \geq e_o$ **then**
        $\eta_c = \beta_2$, $\eta_o = \beta_3$
    else
        $\eta_c = \beta_3$, $\eta_o = \beta_2$
else if $e_t < e_c$ and $e_t \geq e_o$ **then**
    $\eta_t = \beta_2$, $\eta_c = \beta_1$, $\eta_o = \beta_3$
else if $e_t \geq e_c$ and $e_t < e_o$ **then**
    $\eta_t = \beta_2$, $\eta_c = \beta_3$, $\eta_o = \beta_1$
else
    $\eta_t = \beta_3$
    if $e_c \geq e_o$ **then**
        $\eta_c = \beta_1$, $\eta_o = \beta_2$
    else
        $\eta_c = \beta_2$, $\eta_o = \beta_1$

Output: η_t, η_c and η_o

Aerial Relay Controller

The mobile relay controller must satisfy the objectives (*ii*) and (*iii*) enumerated in Section 3.1.1. Indeed, these objectives are satisfied if Definition 3.3 holds. Thus, the mobile relay r needs to maintain an adequate relative position respect to each one of the mobile sensors in $\mathcal{S}$. Therefore, we use potential function approach also for the aerial where

$$U_r(\mathbf{q}_r) = \sum_{i=1}^{N} U(\mathbf{q}_r, \mathbf{q}_{s_i}), \tag{3.12}$$

with

$$U(\mathbf{q}_r, \mathbf{q}_{s_i}) = \begin{cases} \frac{1}{2}\eta_r \varrho_{rs}^2(\mathbf{q}_r, \mathbf{q}_{s_i}) & \text{if } \varrho_{rs}(\mathbf{q}_r, \mathbf{q}_{s_i}) \geq \frac{\delta_r}{2}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

Here, $\varrho_{rs}(\mathbf{q}_r, \mathbf{q}_{s_i}) = \|\mathbf{q}_r^{(xy)} - \mathbf{q}_{s_i}^{(xy)}\|$. Thus, the controller law is defined as

$$\mathbf{u}_r = -k_3 \begin{bmatrix} \frac{\partial U_r(\mathbf{q}_r)}{\partial x_r} & \frac{\partial U_r(\mathbf{q}_r)}{\partial y_r} & 0 \end{bmatrix}^T + k_4 \dot{\mathbf{q}}_r, \tag{3.14}$$

where k_3 and k_4 are positive constants.

3.1.3 Simulation Results

We develop a 3-D environment to test the methodology proposed in Section 3.1.2. The sensor platform geometry as well as its FOV geometry are shown in Figure 3.2(a). On the other hand, the mobile relays are assumed as point robots, but they are represented as a thin cross with four circles at each side just for visualization purposes. Obstacle geometries and target geometries are assumed to be known *a priori*. The heterogeneous network considered for simulation is formed by 1 mobile relay and 3 mobile sensors. The network moves in an environment $\mathcal{W}$ of $4 \times 4 \times 1.5$ m^3 with 4 obstacles and 2 targets, see Figure 3.3(a). The base station is located at the position $(-1.75, 1.75)$ m closed to where the network starts moving. The communication coverage radius for the mobile relay is $\delta_r = 1$ m. The equations of motion for the ground sensors (3.3) and for the aerial relay (3.4) are implemented at each time step using ode45-differential equation solver. The step time is set up at 0.01 s.

We assume that $\beta_1 = 0.295, \beta_2 = 0.125$, and $\beta_3 = 0.105$ for Algorithm 3.1, while we set $k_0 = k_1 = 0.75$ for the sensors' control law (3.11). In the case of the mobile relay controller (3.14), we have that $k_3 = 0.95$ and $k_4 = 0.85$ and $\eta_r = 0.425$ for (3.13). After 35 seconds, the heterogeneous robotic network arrives at the first target and it is in the second target approximately at 47 seconds. These instants

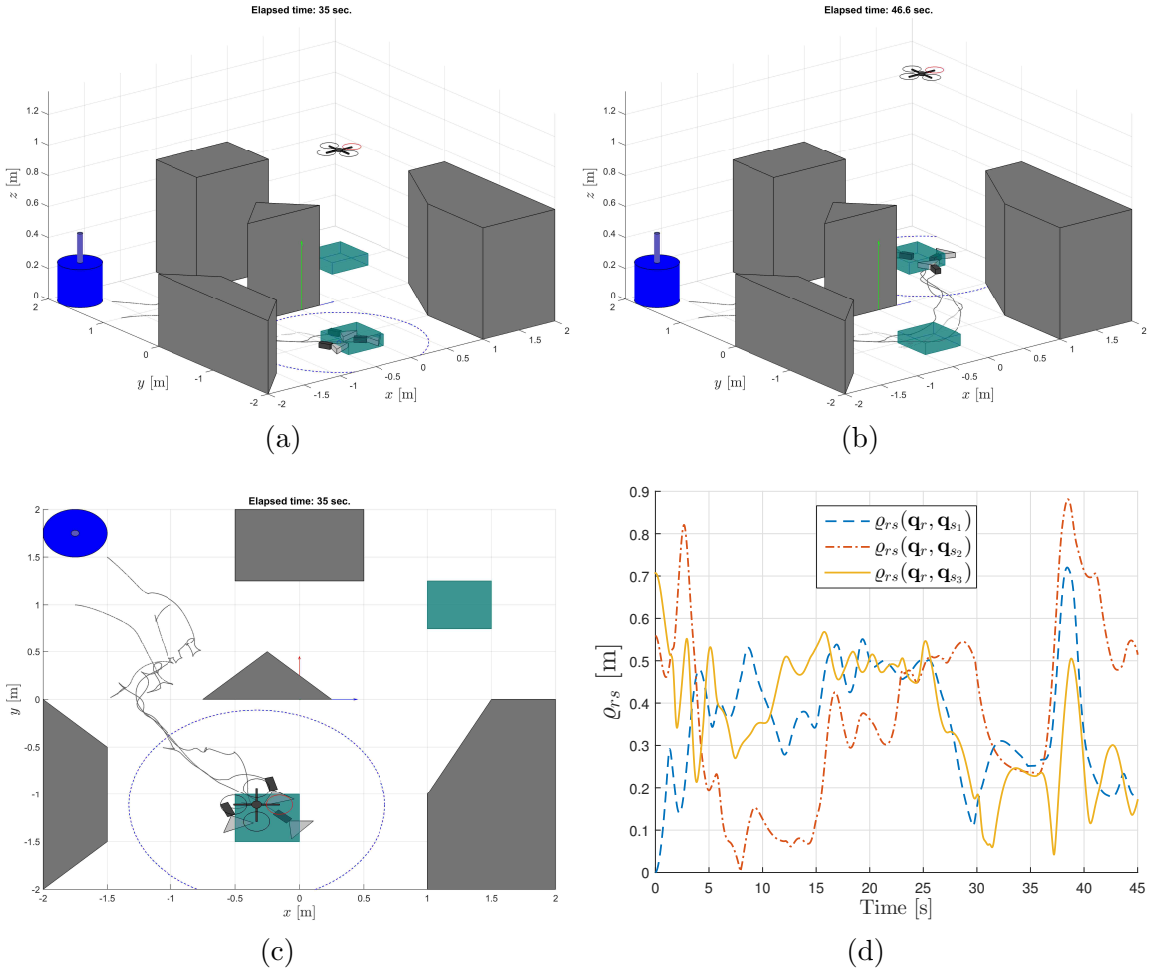


Figure 3.3: (a) and (b) 3-D view of the simulation environment: 4 obstacles, 5 targets, 1 aerial relay, and 3 ground sensors. The sensors arrive approximately after 35 s to the first target region; while approximately after 47 s to the second region. (b) 2-D view of the snapshot presented in figure (a). (c) Evolution of the relative distance of the aerial relay with respect to each ground sensor.

are illustrated in Figures 3.3(a) and 3.3(b), respectively. The distance of each one of the sensors with respect to the aerial relay are presented in Figure 3.3(d). For the simulation experiment, we assume that $\delta_r = 1$ m then the controller for the relay has to maintain the distance with all the ground sensors less than this value. The results in Figure 3.3(d) show that the controller is successfully keeping an adequate

relay-sensor distance for each one of the three .

3.2 Connectivity Maintenance - Multiple Aerial Communication Relays

In Section 3.1, we analyzed the case where the set of heterogeneous robots $\mathcal{A}$ is composed by a set of N ground mobile sensors denoted as $\mathcal{S} = \{s_1, ..., s_N\}$ and one mobile relay r. Here, we study the case when we have M aerial relays which form the set $\mathcal{R} = \{r_1, ..., r_M\}$ such that the set of heterogeneous robots is now given by $\mathcal{A} = \mathcal{S} \cup \mathcal{R}$. In addition, we assume a more realistic communication channel model between the robots than the traditional disc model adopted for the communication link definitions given in Section 3.1. The work presented in this section is based on our paper published in [92].

3.2.1 Received Signal Strength (RSS)

We consider as communication nodes: the base station b, the set of sensors $\mathcal{S}$, and the set of relays $\mathcal{R}$. Thus, we define $\mathcal{M} = \{b\} \cup \mathcal{S} \cup \mathcal{R}$ as the set of communication nodes. Notice that the cardinality of this set is $|\mathcal{M}| = N + M + 1$. Let $I_{\mathcal{M}}$ be the index set of $\mathcal{M}$, then $\forall i \in I_{\mathcal{M}}$, the set of neighbors at the vicinity of the communication node $n_i \in \mathcal{M}$ is given by

$$\mathcal{N}_{n_i} = \left\{ n_j \in \mathcal{M} \,|\, d(n_i, n_j) \leq \delta_{n_i} \right\}. \tag{3.15}$$

Here, $d(n_i, n_j)$ is the distance between the nodes n_i and n_j, and δ_{n_i} is the maximum vicinity range of node n_i.

Assuming the i^{th} node acts as receiver (Rx) and the j^{th} node as transmitter (Tx), we adopt a log-distance path-loss model [93] to estimate the power loss between Tx and Rx. For this model, the noise-free received power at the i^{th} node from the j^{th}

node is given by

$$p_{ij} = p_{\mathrm{TX}_j}\,\kappa\left[\frac{d_o}{d(n_i, n_j)}\right]^{\alpha}, \tag{3.16}$$

where p_{TX_j} is the power transmitted by the j^{th} node, d_o is a reference close-in distance for the antenna far field, α is the path-loss exponent, and κ is a unitless constant that depends on the antenna characteristics and the frequency of operation. We assume that p_{TX_j} is the same for all nodes, so $p_{\mathrm{TX}_j} = p_{\mathrm{TX}}$. The log-distance model is a simplified piecewise model with two segments: free-space propagation is assumed up to distance d_o and non-free-space propagation is assumed for distances grater than d_o. Typical values for d_o are 1-10 m indoors and 10-100 m outdoors. The value of the path-loss exponent α depends on the propagation environment. Its value ranges from 2 for free-space propagation up to 6 for heavy cluttered environments [93]. The constant κ can be set to the free-space path gain at distance d_o given by

$$\kappa = G_{n_i} G_{n_j} \left(\frac{\lambda}{4\pi d_o}\right)^2. \tag{3.17}$$

Here, G_{n_i} and G_{n_j} are the antenna gains for the nodes n_i and n_j, respectively, and $\lambda = c/f$ is the wavelength of the transmitted signal (c is the speed of light and f is the communication frequency). For the base station and the mobile sensors, we assume that they are equipped with unidirectional antennas. Thus, the antenna gain for these cases is 1. On the other hand, we consider that the aerial relays have directional antennas which means that the antenna gain take on a value ≤ 1.

Every received-transmitter link is affected by obstructions and clutter, so the effective received power at the i^{th} node from the j^{th} node can be expressed as

$$P_{ij} = p_{ij}\,\zeta_{ij}, \tag{3.18}$$

with $\zeta_{ij} = 10^{\frac{\chi_{ij}}{10}}$ being the model of a log-normal shadowing noise [93]. Each χ_{ij} is assumed to have a Gaussian distribution with zero mean and variance σ_{ij}^2. Furthermore, we assume independent and homogeneous log-normal shadowing across all

nodes, then $\sigma_{ij}^2 = \sigma^2$. The total received signal strength at the i^{th} node, RSS_{n_i}, represents an aggregate of the power emanating from all the transmitters in its vicinity. This parameter measured in dB can be expressed as [94]

$$\text{RSS}_{n_i} = 10 \log_{10} \sum_{j \in I_{\mathcal{N}_{n_i}}} P_{ij}, \tag{3.19}$$

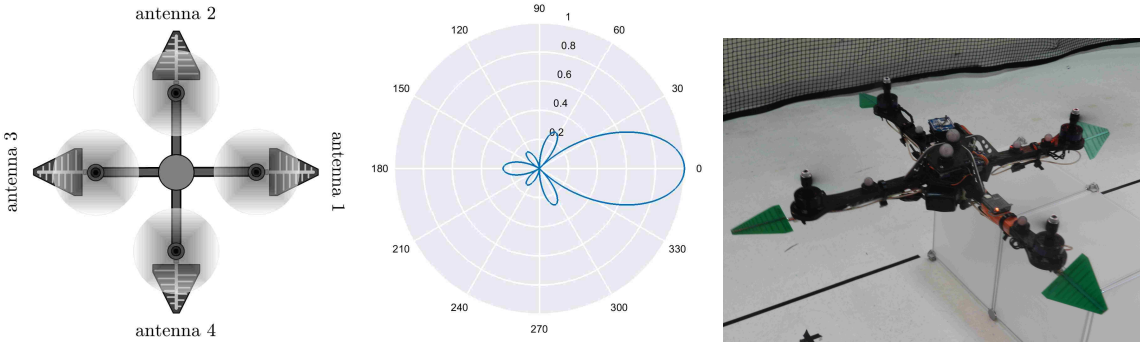
where $I_{\mathcal{N}_{n_i}}$ is the index set of $\mathcal{N}_{n_i}$. From (3.18), we also define the received signal strength at the i^{th} node with respect to the j^{th} node as

$$\text{RSS}_{n_i,n_j} = 10 \log_{10} \left(p_{ij} \, \zeta_{ij} \right). \tag{3.20}$$

3.2.2 Antenna Diversity

For the case of the aerial relays, we consider that each relay has four directional antennas. Each antenna is mounted at the end of each arm, see Figure 3.4. The use of multiple antennas, known as antenna diversity, improves wireless links and is especially effective at mitigating multipath. Antenna diversity also has the advantage that while one antenna is in a deep fade, it is probable that another antenna has a strong enough signal [93, 95]. We assume directional antennas since they radiate greater power in a specific direction as opposed to omnidirectional antennas that radiate power uniformly. This characteristic allows the quadrotor to sense the unknown RF environment and find a direction to move improving the communication performance.

We assume the incoming signal is received by all four antennas but only the signal taken from the antenna with the highest RSS is used. Therefore, four RSS values have to be calculated for every aerial relay $r_j \in \mathcal{R}$. These values can be found by applying (3.16) to (3.19) considering the distance with respect to each antenna as well as the individual antenna gain. Let $\text{RSS}_{r_j}^{(k)}$ be the received signal strength of the k^{th} antenna at the j^{th} aerial relay. Notice that $k = 1, \ldots, 4$. Then, the received



(a) (b) (c)

Figure 3.4: (a) Sketch showing four directional antennas mounted on a quadrotor vehicle. (b) Gain pattern of a directional antenna with a 45° width for its main lobe for both the x-y and x-z planes. (c) An aerial relay equipped with four directional antennas.

signal strength for the j^{th} relay is

$$\text{RSS}_{r_j} = \max\left(\text{RSS}_{r_j}^{(k)}\right). \tag{3.21}$$

3.2.3 Connectivity Matrix

In order to describe the connectivity of the network formed by the nodes in $\mathcal{M}$, we construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. This graph consists of the set of vertices $\mathcal{V} = \mathcal{M}$ and the set of edges $\mathcal{E}$ such that $\forall n_i, n_j \in \mathcal{M}$

$$(n_i, n_j) \in \mathcal{E} \text{ iff } i \neq j \text{ and } \text{RSS}_{n_i,n_j} \geq \upsilon_{ij}. \tag{3.22}$$

Here, υ_{ij} is the RSS threshold needed to maintain a minimal link quality between the nodes n_i and n_j. In this chapter, we consider that $\upsilon_{ij} = \upsilon$ for all nodes. We also assume that the links between all nodes are symmetric; *i.e.*, $\text{RSS}_{n_i,n_j} = \text{RSS}_{n_j,n_i}$. From this assumption and by condition (3.22), the graph $\mathcal{G}$ is undirected and has no self-loops. For this type of graph, $\mathcal{G}$ is connected if and only if there exists a path between any two vertices [96, 97]. Hence, we define a $|\mathcal{M}| \times |\mathcal{M}|$ matrix $\mathbf{C} = [C_{ij}]$

such that

$$C_{ij} = \begin{cases} 1 & \text{if } (n_i, n_j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, C_{ij} is nonzero just when RSS_{n_i,n_j} is sufficiently high to establish a link between n_i and n_j. For this reason, we call $\mathbf{C}$ the connectivity matrix. Let $\mathbf{L}$ be the Laplacian matrix associated with the graph $\mathcal{G}$ defined by

$$\mathbf{L} = \text{diag}(\delta_i) - \mathbf{C}, \tag{3.23}$$

with $\delta_i = \sum_{j=1}^{|\mathcal{M}|} C_{ij}$. From graph theory [96, 97], we know that

$$\mathcal{G} \text{ is connected if and only if } \text{rank}(\mathbf{L}) = |\mathcal{M}| - 1 = N + M.$$

Consequently, we know that the network formed by the base station b, the set of N mobile sensors $\mathcal{S}$ and the set of M serial relays $\mathcal{R}$ is connected if the rank of the Laplacian matrix $\mathbf{L}$ given by (3.23) is equal to $N+M$. From this results, the relative connectivity index [21] can be defined as

$$c = \frac{\text{rank}(\mathbf{L})}{N + M}. \tag{3.24}$$

The network is connected if and only if $c = 1$. Otherwise, it is broken and as c tends to zero the network has more and more disconnected components [97]. We compute this index during our simulation tests.

3.2.4 Coordinated Control

Given a heterogeneous robotic team $\mathcal{A}$ formed by a set of mobile sensors $\mathcal{S}$ with dynamics given by (3.3) and a set of aerial relays $\mathcal{R}$ with dynamics given by (3.4), our goal is to design controllers for these two sets such that the sensors get measurements of the target regions specified in the set $\mathcal{T}$ while the aerial relays maintain network connectivity between the mobile sensors and the base station b.

Mobile Sensor Controller

We consider that every mobile sensor in $\mathcal{S}$ has assigned a target region in $\mathcal{T}$ whose location is known *a priori*. We denote this target region as τ_{s_i} for all $i \in I_{\mathcal{S}}$. Similarly, as in Section 3.1, the workspace $\mathcal{W}$ is populated with L static obstacles. Therefore, each mobile sensor requires to be guided to obtain measurements of its preassigned target region avoiding collisions with obstacles and other mobile sensors. Summarizing, the control objectives for the case of the mobile sensors are: (i) obtaining measurements of the target region, (ii) avoiding obstacles, and (iii) avoiding collisions with other sensors. For every $i \in I_{\mathcal{S}}$, we use a similar potential function as the one given in (3.5) but redefining its components. Thus, we have

$$U_i = U_t + U_c + U_o, \tag{3.25}$$

where

$$U_t = \begin{cases} \frac{1}{2}\eta_t \varrho_t^2(\mathbf{q}_{s_i}, \tau_{s_i}) & \text{if } \varrho_t(\mathbf{q}_{s_i}, \tau_{s_i}) \geq d_{o_{sens}} \\ 0 & \text{otherwise} \end{cases},$$

$$U_c = \sum_{j=1, j\neq i}^{N} U(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}), \quad \text{and} \quad U_o(\mathbf{q}_{s_i}) = \sum_{j=1}^{L} U(\mathbf{q}_{s_i}, \boldsymbol{o}_j).$$

Here,

$$U(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) = \begin{cases} \frac{1}{2}\eta_c \left(\frac{1}{\varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j})} - \frac{1}{d_{o_{col}}}\right)^2 & \text{if } \varrho_s(\mathbf{q}_{s_i}, \mathbf{q}_{s_j}) \leq d_{o_{col}} \\ 0 & \text{otherwise} \end{cases}, \quad \text{and}$$

$$U(\mathbf{q}_{s_i}, \boldsymbol{o}_j) = \begin{cases} \frac{1}{2}\eta_{rep} \left(\frac{1}{\varrho_o(\mathbf{q}_{s_i}, \boldsymbol{o}_j)} - \frac{1}{d_{o_{obj}}}\right)^2 & \text{if } \varrho_o(\mathbf{q}_{s_i}, \boldsymbol{o}_j) \leq d_{o_{obj}} \\ 0 & \text{otherwise} \end{cases}.$$

Using the gradient of the potential function (3.25), ∇U_i, the control inputs for the i^{th} mobile sensor are given by (3.11) that is the same control law that we employed in Section 3.1.

Aerial Relay Controller

For the case of the aerial relays, we assume they maintain a safe height above the obstacles at all times. Thus, the aerial relays do not require obstacle avoidance, but they have to keep a safe distance among them. While the goal of the mobile sensors is to get measurements of the target regions, the flying relays have to keep the connectivity of the whole network, particularly between the base station and the mobile sensors. The proposed controller for the relays uses a cooperative learning technique [21] to maintain the RSS between relay and sensors above a threshold in order to have a good link quality.

We define $\mathcal{D}_{r_j} \subset \{b\} \cup \mathcal{R}$ as the set of neighbors for which the j^{th} relay has to keep a relative distance μ_r. This set for each relay is defined *a priori*. We assume that one of the relays has the base station as neighbor and that all the relays have at least one neighbor. Also, we consider that if r_j is a neighbor of r_k then r_k is also a neighbor of r_j. From our definition of $\mathcal{D}_{r_j}$, $\mathcal{D}_{r_j} \cap \mathcal{S} = \emptyset$ for every j. There is no consideration that a relay should maintain a sensor inside its vicinity since the sensors explore around $\mathcal{W}$ moving from the vicinity of one relay to another. However, we use the RSS among relays and the sensors inside their vicinity to maintain network connectivity.

We define the acceleration control input for the j^{th} aerial router as

$$\mathbf{u}_{r_j} = k_2 \sum_{i \in I_{\mathcal{D}_{r_j}}} (\ell_{ij} - \mu_r)\, \mathbf{e}_{ij} - k_3 \dot{\mathbf{q}}_{r_j} - k_4 \left(\mathbf{q}_{r_j} - \mathbf{q}_{\text{ref}}\right), \tag{3.26}$$

where $I_{\mathcal{D}_{r_j}}$ is the index set of $\mathcal{D}_{r_j}$. The first two terms in (3.26) create a virtual spring [3] among r_j and its neighbors, so the relative distance μ_r can be maintained. For these terms, k_2 and k_3 are the spring constant and the damping coefficient, respectively. Also, we define for every i^{th} neighbor of r_j in $\mathcal{D}_{r_j}$ the length and the direction of the force of the virtual spring as

$$\ell_{ij} = \|\mathbf{q}_i - \mathbf{q}_{r_j}\|, \quad \text{and} \quad \mathbf{e}_{ij} = \frac{\mathbf{q}_i - \mathbf{q}_{r_j}}{\|\mathbf{q}_i - \mathbf{q}_{r_j}\|},$$

respectively. Here, $\mathbf{q}_i$ is the position with respect to $\mathcal{F}_{\mathcal{W}}$ of the i^{th} neighbor of r_j.

The last term in (3.26) helps to move the aerial relay r_j such that the RSS of r_j is improved taking into account the RSS with respect to the mobile sensors in its vicinity. From Section 3.2.2 and using equation (3.19), we redefine the RSS for the j^{th} mobile relay considering just the mobile sensors in its vicinity. Thus, we get that

$$\text{RSS}_{r_j}^{(k)} = 10 \log_{10} \sum_{i, n_i \in \mathcal{N}_{r_j} \cap \mathcal{S}} P_{ij}, \tag{3.27}$$

where $k = 1, 2, 3, 4$ is the number of antenna. Then redefining (3.21), we obtain

$$[\text{RSS}_{r_j}, A_{r_j}] = \max \left(\text{RSS}_{r_j}^{(k)} \right), \text{ with } k = 1, 2, 3, 4. \tag{3.28}$$

Here, the max function returns not only the maximum RSS_{r_j} (the maximum received signal strength between all the four antennas), but also the number of the antenna with this received signal strength A_{r_j}. Using RSS_{r_j} and A_{r_j}, we select the reference position $\mathbf{q}_{\text{ref}}$ required by the last term of (3.26) base on a cooperative reinforcement learning technique.

Cooperative Q-learning Recently, reinforcement learning (RL) has grown as an effective approach for control applications [98] and in particular to allow multi-robot systems to learn cooperation [21, 99, 100]. Reinforcement learning refers to an actor, *e.g.*, a robotic agent, which interacts with its environment and modifies its control policy based on a reward received in response to its actions. Thus, RL algorithms seek to find the optical control policy that maximizes the future reward. In RL, the actor only knows its previous and current states and the reward of how good the previous action was. Let x^t and u^t be the agent state and the action at time t. Then, the action u^t produces a new state x^{t+1} from interacting with the environment and a reward ρ^{t+1} is observed. Applying RL, the agent then selects a new action base on x^t, x^{t+1}, and ρ^{t+1}. Typically, RL algorithms must balance the need to collect informative data (by *exploring* novel action choices) with the need to control the

process (by *exploiting* the currently available knowledge) [101–103]. Exploration is when the agent selects random actions while learning regardless the current state. In this way, the agent can explore the entire state-action space to avoid the problem of learning local maxima. In contrast, exploitation is the use of the agent's knowledge in selecting actions. The agents choose the action that will produce the maximum reward given the current state. The trade-off between exploration and exploitation is challenging. For example, using mostly exploration can cause the agent to take a long time to learn. Usually at the beginning of the learning, exploration is mainly used and then, the exploration diminishes over time and exploitation takes over. One approach to transition between exploration and exploitation is the ε-greedy technique [101]. We use this technique in our algorithm for implementing the exploration-exploitation trade-off.

Q-learning is a model-free value iteration RL algorithm which uses a state-action value function known as Q-function [101–103]. This function represents the expected utility of taking a given action in a given state and following the optimal policy thereafter. In fact, Q-learning starts from an arbitrary Q-function and updates it without requiring a model and using instead observed state transitions and rewards. Let x_j^t, u_j^t, and ρ_j^t be the state, action and reward of the j^{th} aerial relay r_j at time t, respectively. Then, we have the following models for the state, action and reward:

The state. For every r_j, the state is defined by the number of the antenna whose RSS is the highest. From (3.28), we have that $x_j = A_{r_j}$ and so $x_j \in \{1, 2, 3, 4\}$. Thus, the total number of states per aerial relay is four.

The action. We assume five possible actions: North, South, East, West, and Stay, which are associated with the unitary vectors

$$\mathbf{e}_1 = [1 \quad 0 \quad 0]^T,\ \mathbf{e}_2 = -\mathbf{e}_1,\ \mathbf{e}_3 = [0 \quad 1 \quad 0]^T,\ \mathbf{e}_4 = -\mathbf{e}_3,\ \text{and}\ \mathbf{e}_5 = \mathbf{0},$$

respectively. Here, $\mathbf{0} \in \mathbb{R}^3$ is the zero vector. Once an action is chosen, we use its associated vector to find the reference position $\mathbf{q}_{\text{ref}}$ required by (3.26). For

example, let u_j = East then

$$\mathbf{q}_{\text{ref}} = \mathbf{q}_{r_j} + \Delta\ell\mathbf{e}_3,$$

where $\Delta\ell$ is a reference distance. Since the total number of states is four and the number of actions is five, we have 20 state-action pairs per each aerial relay.

The reward. It is calculated based on RSS_{r_j} obtained by applying (3.28). Since our purpose is to maintain RSS_{r_j} above the threshold υ, we define the reward function as follows

$$\rho_j = \begin{cases} 100 & \text{if } \text{RSS}_{r_j} > \upsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{3.29}$$

Each aerial relay r_j updates an individual Q-function $\bar{Q}_j(x_j, u_j)$ as follows

$$\bar{Q}_j^{t+1}(x_j, u_j) = Q_j^t(x_j, u_j) + \eta \left[\rho_j^{t+1} + \gamma \max_{\hat{u}_j} Q_j^t(x_j^{t+1}, \hat{u}_j) - Q_j^t(x_j, u_j)\right], \tag{3.30}$$

where $\eta \in (0, 1]$ is the learning rate and $\gamma \in (0, 1]$ is the discount factor. However, our intention is to allow that each aerial relay aggregates information of its relay neighbors via their individual Q-functions. Let $\mathcal{B}_{r_j} = \mathcal{D}_{r_j} \setminus \{b\}$ be the set of the relay neighbors of relay r_j, then each r_j updated the total Q-function $Q_j(x_j, u_j)$ following the equation for the cooperative Q-learning technique described in [21]

$$Q_j^{t+1}(x_j, u_j) = \text{w}\bar{Q}_j^t(x_j, u_j) + (1 - \text{w}) \sum_{i \in I_{\mathcal{B}_{r_j}}} \bar{Q}_i^t(x_i, u_i), \tag{3.31}$$

where $I_{\mathcal{B}_{r_j}}$ is the index set of $\mathcal{B}_{r_j}$ and $\text{w} \in [0, 1]$ is the trust weight. If $\text{w} = 1$, the Q-function is updated by trusting only in the knowledge of the relay itself (independent learning). On the other hand, if $\text{w} = 0$, the aerial relay updates its Q-function based only on the Q-functions of its relay neighbors. When $0 < \text{w} < 1$, the learning is carried out in a cooperative fashion. The convergence of the cooperative Q-learning algorithm is analyzed in [21]. Q^t will converge if $\eta \in (0, 1)$ and $\gamma \in (0, \frac{1}{n})$ with n being the number of agents performing the cooperative learning. For a complete discussion, the reader is referred to [21].

Once the Q-function is updated, we use the ε-greedy technique to balance exploration with exploitation [101, 103]. This method selects actions according to

$$u_j^t = \begin{cases} \arg\max_{\hat{u}} Q_j^t(x_j, \hat{u}_j) & \text{with probability } 1-\varepsilon_t \text{ (exploit)}, \\ \text{uniformly random action} & \text{with probability } \varepsilon_t \text{ (explore)}. \end{cases} \tag{3.32}$$

Here $\varepsilon_t \in (0, 1]$ is the exploration probability at time step t. A typical value for the exploration probability is $\varepsilon_t = \frac{1}{t}$. Algorithm 3.2 summarizes the cooperative Q-learning method for every aerial relay.

Algorithm 3.2 Cooperative Q-learning $\forall r_j \in \mathcal{R}$

Require: learning rate η, discount factor γ, initial Q-function Q_j^0 (*e.g.*, $Q_j^0 \leftarrow 0$), initial state x_j^0, trust weight w

1: **for** every time step $t = 0, 1, 2, \ldots$ **do**

2: $u_j^t = \begin{cases} \arg\max_{\hat{u}} Q_j^t(x_j, \hat{u}_j) & \text{with prb. } 1 - \frac{1}{t+1} \\ \text{unif. rand. in } \{\text{Nort, South, East, West, Stay}\} & \text{with prb. } \frac{1}{t+1} \end{cases}$

3: Apply u_j^t

4: Find next state x_j^{t+1} and the reward ρ_j^{t+1}

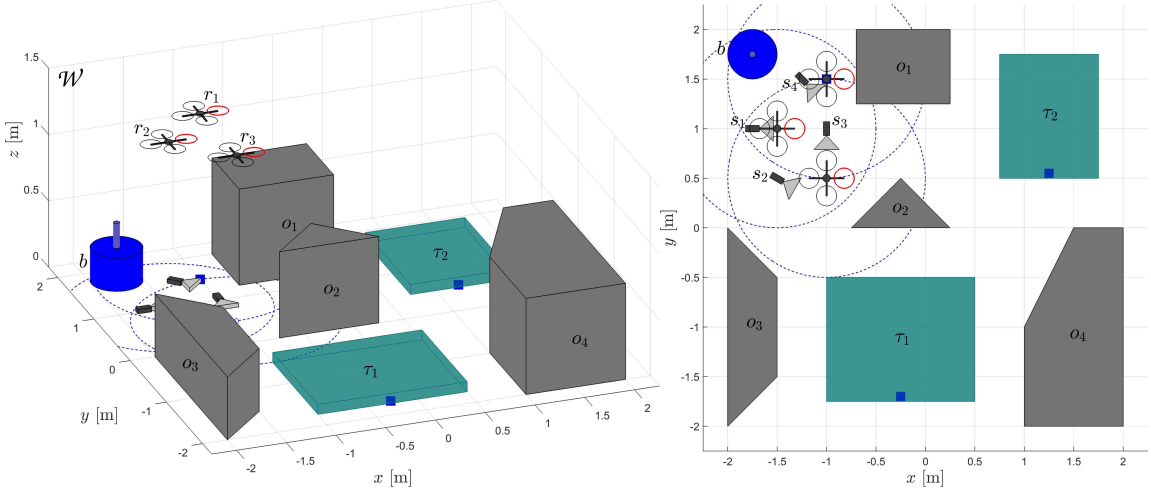
5: $\bar{Q}_j^{t+1}(x_j, u_j) \leftarrow Q_j^t(x_j, u_j) + \eta \left[\rho_j^{t+1} + \gamma \max_{\hat{u}_j} Q_j^t(x_j^{t+1}, \hat{u}_j) - Q_j^t(x_j, u_j)\right]$

6: $Q_j^{t+1}(x_j, u_j) \leftarrow \mathrm{w}\bar{Q}_j^t(x_j, u_j) + (1 - \mathrm{w}) \sum_{i \in I_{\mathcal{B}_{r_j}}} \bar{Q}_i^t(x_i, u_i)$

3.2.5 Simulation Results

In this section, we present simulation results of the proposed methodology for a network of mobile sensors and aerial relays. For the case of the cooperative Q-learning method (Algorithm 3.2), we compare the independent learning ($\mathrm{w} = 1$) with the cooperative learning (assuming $\mathrm{w} = 0.5$) in terms of the network connectivity.

We consider a heterogeneous team of robots made by four mobile sensors, *i.e.*, $N = 4$, and three aerial relays, *i.e.*, $M = 3$. The workspace $\mathcal{W}$ is of $4 \times 4 \times 1.5$ m^3 populated with four convex obstacles and two target regions. Figure 3.5 illustrates



(a) (b)

Figure 3.5: (a) 3D view and (b) 2D view of the workspace employed for the simulation tests. The workspace $\mathcal{W}$ has a base station b, four obstacles, $\mathcal{O} = \{o_1, o_2, o_3, o_4\}$, and two target regions $\mathcal{T} = \{\tau_1, \tau_2\}$. The heterogeneous robotic network is formed by four mobile sensors, $\mathcal{S} = \{s_1, s_2, s_3, s_4\}$, and three aerial relays, $\mathcal{R} = \{r_1, r_2, r_3\}$. The blue squares inside τ_1 and τ_2 denote the reference points used for the initial deployment for the aerial relays.

this workspace where the obstacles are in gray and the target regions are in green. The base station is represented in blue and its position is $\mathbf{q}_{\text{base}} = [-1.75 \quad 1.75 \quad 0]^T$ m. Also, the initial position of the ground sensors and the aerial relays are shown. The blue dashed circles denotes the vicinity range for each flying relay. We assume that for all the communication nodes the vicinity range is 1 m, *i.e.*, $\forall n_i \in \mathcal{M}$, $\varrho_{n_i} = 1$ m. For set of relays $\mathcal{R} = \{r_1, r_2, r_3\}$, we have that

$$\mathcal{D}_{r_1} = \{b, r_2, r_3\}, \quad \mathcal{D}_{r_2} = \{r_3\}, \quad \text{and} \quad \mathcal{D}_{r_3} = \{r_2\}.$$

Thus, r_1 has to keep a relative distance with the base station and with the other sensors, while r_2 and r_3 have to do the same with r_1. The time step is 0.005 s and we run the simulation for 30 s. The rest of the simulation parameters are summarized in Table 3.1.

Table 3.1: Simulation parameters

Network Connectivity	$p_{\text{TX}} = 60$ mW, $\alpha = 2.5$, $d_o = 0.5$ m, $f = 2.4$ GHz, $\sigma = 0.25$, $\upsilon = -70$ dB
Mobile Sensor Controller	$d_{o_{sens}} = 0.1$ m, $d_{o_{col}} = 0.2$ m, $d_{o_{obj}} = 0.3$ m, $\eta_t = 0.6$, $\eta_c = \eta_{rep} = 0.4$, $k_0 = 0.75$, $k_1 = 1.25$
Aerial Relay Controller	$\mu_r = 1.5$ m, $\eta = 0.2$, $\gamma = 0.25$, $\Delta L = 0.75$ m, $k_2 = 5.75$, $k_3 = 3.75$, $k_4 = 2.5$

At the beginning of the simulation, we deploy just the aerial routers such that r_2 gets closer to the first target τ_1, and r_3 to the second target τ_2. To achieve this, we use the control law (3.26), but $\mathbf{q}_{\text{ref}}$ is fixed to a position inside their respective target regions. The blue squares within τ_1 and τ_2 represent the reference positions. For the case of r_1, we also use (3.26), but without considering the third term, *i.e.*, $\text{k}_4 = 0$. Therefore, we do not use the learning algorithm during this initial deployment. Figure 3.6 shows a series of snapshots of the deployment of the aerial relays. Since we take into consideration damping effects in the controller (3.26), the mesh of virtual springs between the flying relays has similar behavior as a real spring system in which dissipative forces act against the movement of the springs ensuring velocity to eventually reach zero. This behavior guarantees that the network of aerial relays eventually reaches a rest state [3] that indicates the end of the initial deployment. Indeed, the initial deployment maneuver finishes, *i.e.*, the flying relays reach the rest state, around 4 s after it started. As can be seen in Figure 3.6, the relays move such that they maintain the relative distance with respect to their predefined neighbors closer to μ_r while they move towards their preassigned reference positions in the target regions, specially for the case of r_2 and r_3. Figure 3.6(c) shows the rest position that the aerial relays achieve.

Once the deployment of the aerial relays is complete, the mobile sensors are commanded to start moving around $\mathcal{W}$ such that they get measurements of the target regions τ_1 and τ_2 by applying the control law (3.11). Also, the aerial relays begin using the learning algorithm together with their controller (3.26) to enhance

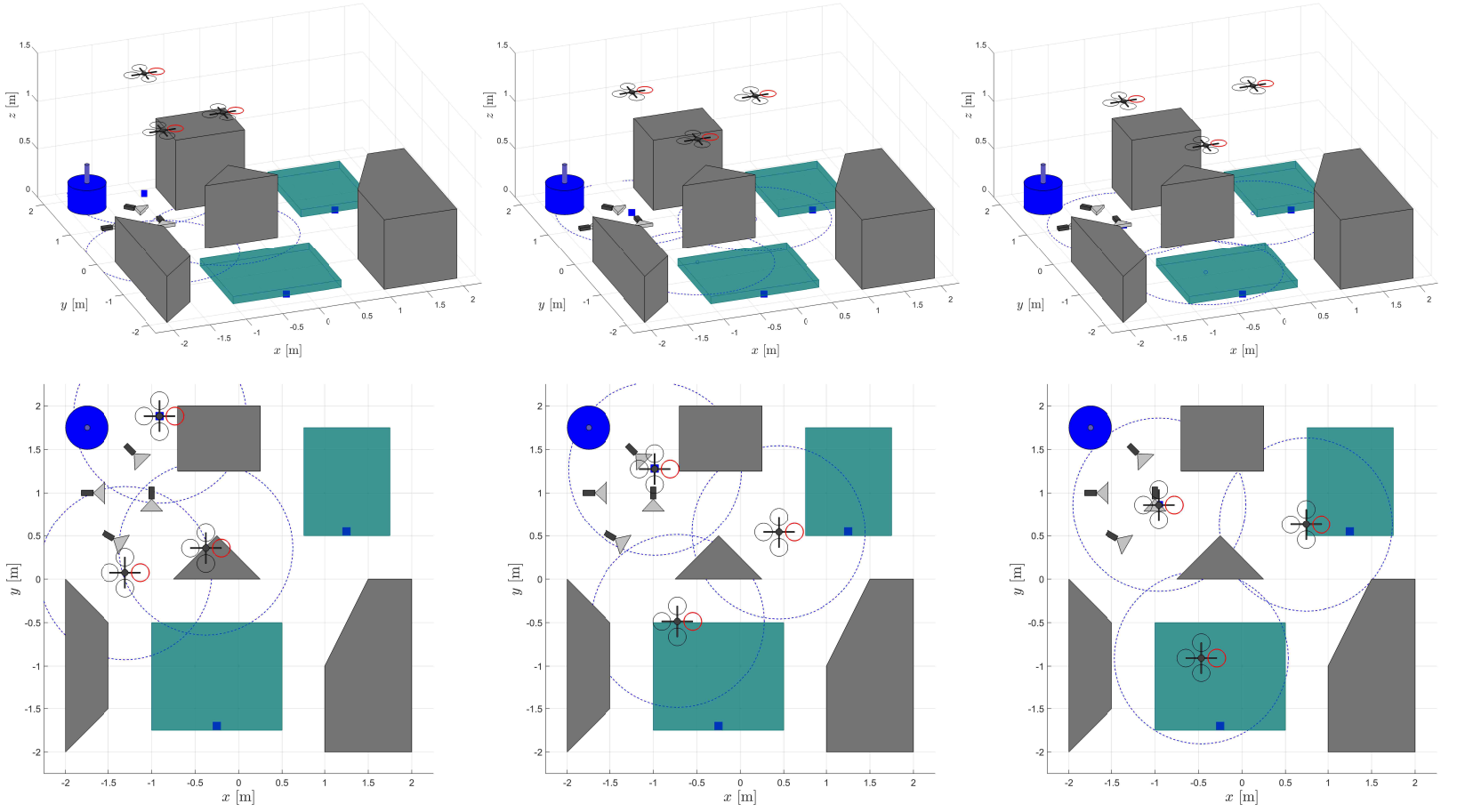


Figure 3.6: Snapshots of the initial deployment of the aerial routers. Time elapsed since the start of the simulation: (a) 1 s, (b) 2 s, and (c) 4 s. The 3D views are at the top and their corresponding 2D view are at the bottom. The maneuver finishes after around 4 s of starting the simulation.

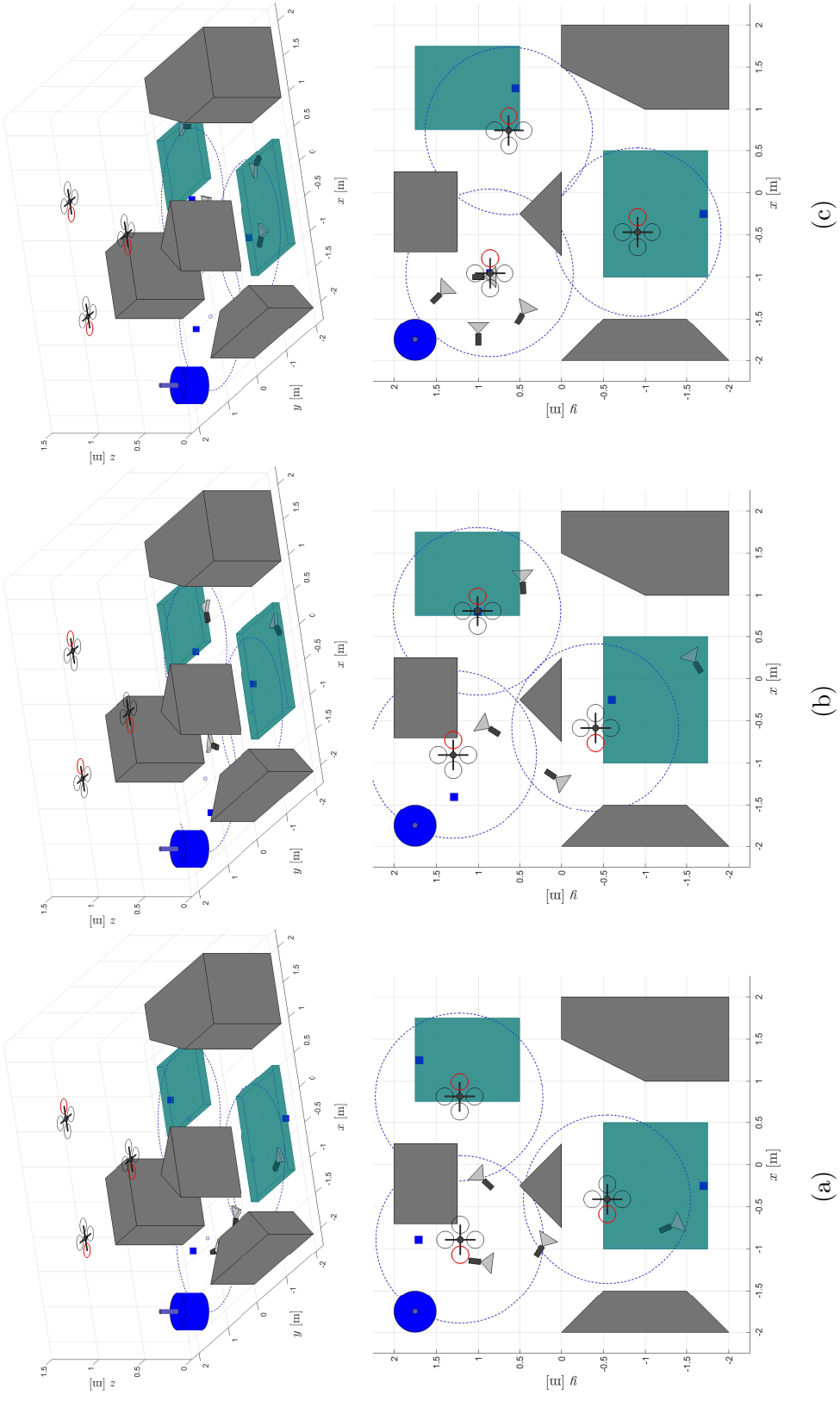


Figure 3.7: Snapshots of the simulation experiment assuming independent learning, *i.e.*, w = 1. Time elapsed since the start of the simulation: (a) 6 s, (b) 10 s, and (c) 20 s. The 3D views are at the top and their corresponding 2D view are at the bottom. In (b) and (c), the network is disconnected since one or more mobile sensors are out of the vicinity range of the mobile routers.

the network connectivity. As for the mobile sensors, two of then are assigned to τ_1 and the others to τ_2. Once a sensor gets to its predefined target region, it continues moving around within the region. The flying relays do not use the learning Algorithm 3.2 at every time step. Instead, Algorithm 3.2 is called every 0.25 s. We do this in order to allow that the aerial relays have enough time to move to the reference position $\mathbf{q}_{\text{ref}}$ computed based on this algorithm. Figure 3.7 shows different time instants of the simulation for the case of independent learning, *i.e.*, $\text{w} = 1$. For each relay, we indicate which antenna has the maximum RSS by changing to red the circle at the end of its corresponding arm.

From Figure 3.7, the mobile sensors move towards their preassigned targets evading obstacles and collisions with other sensors. At the same time, the aerial relays select the antenna with the maximum RSS and change its position according to the reference position computed based on Algorithm 3.2. In Figure 3.7(b), one of the mobile sensors assigned to τ_1 is out of the vicinity range of the relays. Indeed, it is not inside any of the blue-dashed circles associated with each flying relay. This causes the network to get disconnected making the relative connectivity index, defined by (3.24), be less than one. Furthermore, the evaluation of this index during the simulation experiments is shown in Figure 3.8(a) for the case of independent learning and in Figure 3.8(b) for cooperative learning. As one can see on these figures, the network connectivity is lost more times when the independent learning method is applied. For example, two mobile sensors are out of the communication vicinity range of the relays in Figure 3.7(c) decreasing its RSS and then having an effect on the relative connectivity index c. Even though c is not always one for the cooperative learning (meaning that the network is always connected), we see from our tests that in the worst case two of the mobile sensors gets disconnected for a short time. This is not the case for the independent learning where, at different time instants, three of the four sensors were disconnected.

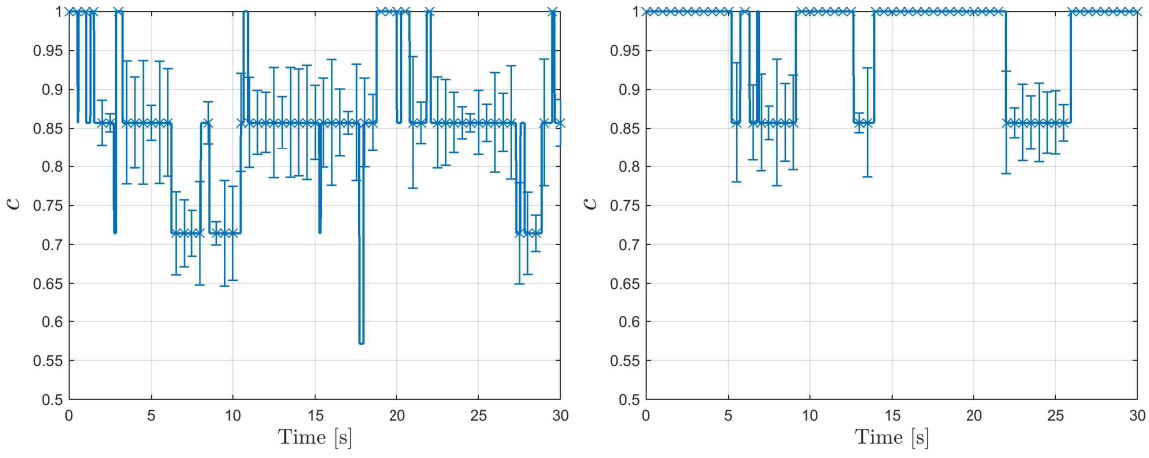


Figure 3.8: Relative connectivity index c for the 20 trials. (a) Results for the case of independent learning (w = 1). (b) Results for the case of cooperative learning ($0 < \mathrm{w} < 1$) with w = 0.5. The cross marker "×" represents the average value, while the bars denote the standard deviation at every 0.5 s.

3.3 Conclusions

In this chapter, we took advantage of the different dynamics, resources and capabilities of a heterogeneous robotic network made of ground and aerial agents. Exploiting their heterogeneity allows to effectively use their specialized abilities and therefore accomplish the overall mission, in our cause to take measurements of multiple targets in a cluttered environment while the network connectivity is maintained. For the case of the mobile ground sensors, we took into account their FOV and platform geometry as part of their communication and sensing models. Meanwhile, we assumed that the aerial agents are specialized communication relays. By using potential field methods, the ground mobile sensors were coordinated to sense regions of interest in the environment avoiding collision. At the same time, in the first section of this chapter, one aerial relay kept an adequate relative position with the ground sensors to maintain connectivity with a fixed base station. We extended our approach to

the case of multiple aerial relays in the second section of this chapter. Indeed, we combined antenna diversity and cooperative Q-learning to maintain a certain RSS level between the sensors and the aerial communication relays. For both cases, we analyzed the effectiveness of our proposed methodology in a 3-D simulation scenario obtaining promising results. In particular for the learning-based method, we considered for the relays the cases of independent and cooperative learning. Our results showed that the network achieved a better relative connectivity index when the learning was carried out in a cooperative fashion.

Since our results showed that the connectivity of the network is not maintained all the time even when the cooperative method is employed, future work will consist in expanding the learning method to the mobile sensors. Thus, they can react when their RSS drops under a minimum communication level. In our proposed method, only the aerial relays are sensing and maintaining the RSS between them and the sensors. We believe that adding the consideration of the minimum RSS communication level to the mobile sensor controller will help to get better results for the whole network connectivity.

Chapter 4

Enabling Optical Wireless Communications

Maintaining reliable communications on heterogeneous robotic networks is fundamentally important, especially for cooperative purposes. Radio communications is the common method which allows the robots to operate wirelessly. However, this technology has some limitations that affect the capacity of the robotic network. Optical wireless (OW) communication has been proposed as the perfect complement to mitigate some of the weaknesses of radio frequency systems [27–29]. Therefore, we are motivated to study designs for an optical wireless link between an aerial robotic platform and a mobile ground robot. Our intention is to combine radio frequency communications and optical data transfer to have robust connectivity in GPS-denied environments. This chapter is based on our work published in [104, 105] and it has been developed in collaboration with Dr. Brian M. Sadler from the Army Research Laboratory (ARL).

To the best of our knowledge, OW communications has not been proposed for the case of a robotic team of aerial and ground vehicles. Possibly the major shortcoming of OW technology that has delayed its mobile application is the LOS pointing and

tracking requirement. Indeed, this challenge has to be addressed to fully exploit the benefits of the optical link. In this chapter, we present our approach to tackle this problem. First, we introduce a model for a directed LOS optical link between an unmanned aerial robot and a ground mobile vehicle. Based on the link model, we define a *connectivity cone* over the receiver where a minimum transmission rate is guaranteed if the transmitter stays within. Second, we address the problem of tracking a ground receiver by an aerial transmitter in order to establish a point-to-point optical communication link. We consider that only noisy measurements of the receiver position are available for the transmitter. Using Bayesian methods, we estimate the location of the connectivity cone. Then, we develop a control to reduce the distance between the aerial transmitter and the cone. Once the transmitter is within the cone, the control acts to optimize the possible communications rate. This general approach is valid for a variety of platform to platform OW communications as we verify through numerical simulations that includes physics based optical channel modeling. In Section 4.3, we consider the problem of pairing a ground sensor with an aerial vehicle, both equipped with a hybrid communication system - radio frequency for non-line of sight transmission and optical for line-of-sight transmission. Once the quadrotor is within a desired range, the optical link can be employed to carry out bulk data transfer. Therefore, the flying robot has to localize the sensor in order to upload/download data while staying within optical communication range. We develop a solution to autonomously localize the ground sensor node relative to the aerial vehicle assuming that the sensor position is unknown at all times. We exploit the hybrid communication scheme in order to solve this problem. The control strategy is demonstrated through simulations that incorporate a realistic model for the hybrid communication link.

4.1 Optical Wireless (OW) Communication Link Model

The indoor optical link is shown in Figure 4.1(a). A hovering aerial vehicle, *e.g.*, a quadrotor, is equipped with an optical transmitter; while a ground vehicle has an optical receiver. Thus, it is possible to establish an optical wireless connection between both platforms. From now on, we will refer to the mobile ground robot as the ground optical receiver and to the aerial robot as the flying optical transmitter. We consider that the optical transmitter can be directed using a pan-tilt (PT) mechanism, so it is possible to point the light beam to the receiver. We assume the workspace spanned by the PT mechanism resembles half a sphere which is sufficient for reaching the ground receptor. The pan and tilt angles are illustrated in the inset in Figure 4.1(a). Conversely, we assume that the optical receiver does not have any directionality and it is always pointing up. With the addition of radio, this type of configuration forms a hybrid LOS optical link [27–29].

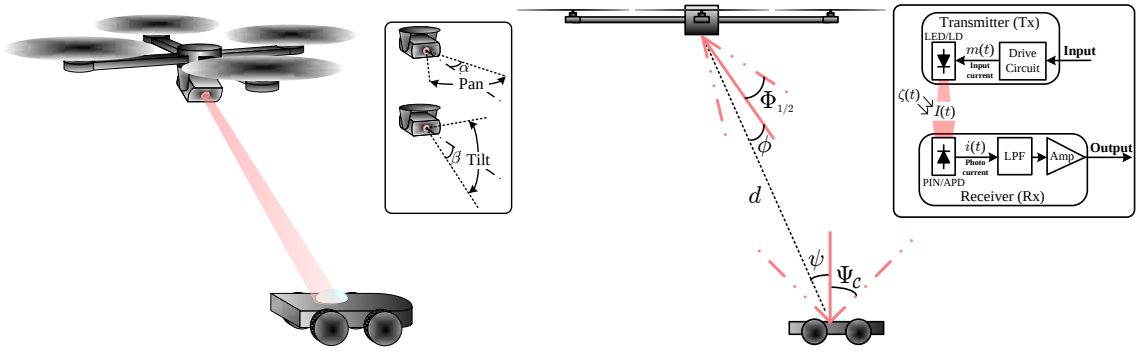


Figure 4.1: (a) Sketch of the OW link. The inset shows the pan and tilt angles of a gimbal mechanism. (b) Diagram of the LOS optical link with its main parameters. The inset presents a block diagram of the OW communication system.

We consider a link model of an intensity-modulation direct-detection (IM/DD)

LOS OW system [29, 106, 107]. The inset in Figure 4.1(b) is a block diagram of this indoor OW communication system. A light-emitting diode (LED) or a laser diode (LD) can be used as optical transmitter and a PIN photodiode or an avalanche photodiode (APD) can be employed as optical receiver. A modulation signal $m(t)$ drives the current of the optical source varying its intensity $I(t)$. The optical receiver integrates the incident optical signal generating a photocurrent $i(t)$ which is directly proportional to the instantaneous optical power incident on it. The signal $\zeta(t)$ represents noise added at the receiver due to ambient light within the pass band of the detector. In general, $\zeta(t)$ is modeled as white Gaussian noise independent of $I(t)$ [28, 29, 106].

A diagram of the optical link with the important parameters for modeling purposes is illustrated in Figure 4.1(b). The transmitter beam width is generally described by its half-angle at half-power denoted as $\Phi_{1/2}$. Similarly, the receiver effective collection area is described by the half-angle field-of-view (FOV) denoted as $\Psi_{\mathcal{C}}$ [29, 106]. In Figure 4.1(b), d is the distance between transmitter and receiver, ϕ is the pointing angle of the transmitter, and ψ is the angle-of-incidence with respect to the receiver axis. Notice that the transmitter is perfectly pointed at the receiver when $\phi = 0°$, so ϕ is also known as the *pointing error* [106]. Similarly, the receiver is perfectly pointed at the transmitter when $\psi = 0°$, so ψ is sometimes referred to as the *receiver pointing error* [106]. Even though all the angles in Figure 4.1(b) are in the same plane, the link model presented next is generally valid with circularly symmetric beam and FOV.

The emission pattern of the transmitter can be described by the irradiance function $I_s(d, \phi)$ [106, 107] given by

$$I_s(d, \phi) = P \frac{m+1}{2\pi d^2} \cos^m \phi, \tag{4.1}$$

where P is the average transmitted optical power and m is the Lambert's mode number expressing directivity of the source beam. The Lambert's mode number m

is related to $\Phi_{1/2}$ by [107]

$$m = \frac{-\ln 2}{\ln(\cos \Phi_{1/2})}.$$

The optical receiver is modeled as an effective area A_{eff} collecting the radiation incident at the angle ψ and is given by

$$A_{\text{eff}}(\psi) = g(\psi) A \cos \psi, \tag{4.2}$$

if no optical filter is used [29,107]. Here, A is the receiver active area and $g(\psi)$ is the light-concentrator gain. An ideal concentrator with internal refractive index n has gain

$$g(\psi) = \begin{cases} \frac{n^2}{\sin^2 \Psi_{\mathcal{C}}} & \text{if } |\psi| \leq \Psi_{\mathcal{C}}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

Based on (4.1) and (4.2), the received signal power is given by

$$P_{\text{Rx}} = I_s A_{\text{eff}}, \tag{4.4}$$

and the corresponding photocurrent is

$$I_p = R P_{\text{Rx}}, \tag{4.5}$$

where R is the responsivity of the photodiode.

The dominant noise source in short range indoor OW systems is the ambient light [27, 29, 107, 108]. Assuming the ambient light is isotropic with a spectral irradiance p_n, the receiver is typically equipped with a bandpass filter with bandwidth $\Delta\lambda$ and peak transmission T_0. Then, the ambient optical power incident at the receiver is equal to

$$P_n = p_n \Delta\lambda T_0 A n^2. \tag{4.6}$$

This ambient light noise is modeled as Gaussian [107] whose variance is approximated by

$$\sigma^2 = 2qRP_nB, \tag{4.7}$$

where q is the charge of an electron and B is the bit rate. Assuming the optical link uses on-off keying (OOK) [107–109], the SNR at the receiver is given by

$$\text{SNR} = \frac{I_p^2}{\sigma^2}. \tag{4.8}$$

The bit-error rate (BER) for OOK is related with the SNR by the equation $\text{BER} = Q(\sqrt{\text{SNR}})$ [107], where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp\left(-\frac{u^2}{2}\right) du$$

is the tail probability of the standard normal distribution (the *Q-function*). For example, to achieve a $\text{BER} = 10^{-4}$ requires $\text{SNR} \approx 11.4$ dB. Therefore, for a given BER and combining (4.7) and (4.8), the corresponding bit rate B can be computed by

$$B = \frac{1}{2qRP_n} \left[\frac{I_p}{Q^{-1}(\text{BER})}\right]^2, \tag{4.9}$$

where P_n is given by (4.6) and I_p can be calculated applying (4.1) - (4.5). Furthermore, assuming a bit rate B, it is possible to solve for the range d by manipulating equations (4.1) - (4.5) and (4.9), yielding

$$d = \left(\frac{R}{2qP_nB}\right)^{\frac{1}{4}} \left[\frac{A_{\text{eff}} P(m+1)\cos^m \phi}{2\pi Q^{-1}(\text{BER})}\right]^{\frac{1}{2}}. \tag{4.10}$$

Table 4.1: Optical link parameters.

Transmitter	$P = 0.03$ W, $\Phi_{1/2} = 45°$
Receiver	$A = 1\text{cm}^2, \Psi_{\mathcal{C}} = 45°, n = 1.5, R = 0.6$ A/W
Noise	$p_n = 5.8\mu\text{W/nm} \cdot \text{cm}^2, \Delta\lambda = 100$ nm, $T_0 = 0.8$

4.1.1 Connectivity Cone

Based on the link model explained in Section 4.1 and using (4.9), we can plot the bit rate of the optical link as a function of the transmitter position relative to the

receiver. Figure 4.2 shows contour maps of the optical link bit rate assuming that the receiver is static at the position $[0 \quad 0 \quad 0]^T$ and assuming a BER $= 10^{-4}$. The contours represent the logarithm of the bit rate. For example, "7" represents 10 Mbps. The parameters used for creating the plots in Figure 4.2 are summarized in Table 4.1 which are adopted from [106]. The optical source power is consistent with a single LED. These plots are obtained considering different values of the pointing error ϕ. The contours in Figure 4.2(a) are for the case when the optical beam is perfectly pointed at the receiver, *i.e.*, $\phi = 0°$. In contrast, we assume for Figures 4.2(b) and 4.2(c) that the transmitter has a constant maximum misalignment with the receiver of $\phi = 25°$ and $\phi = 50°$, respectively.

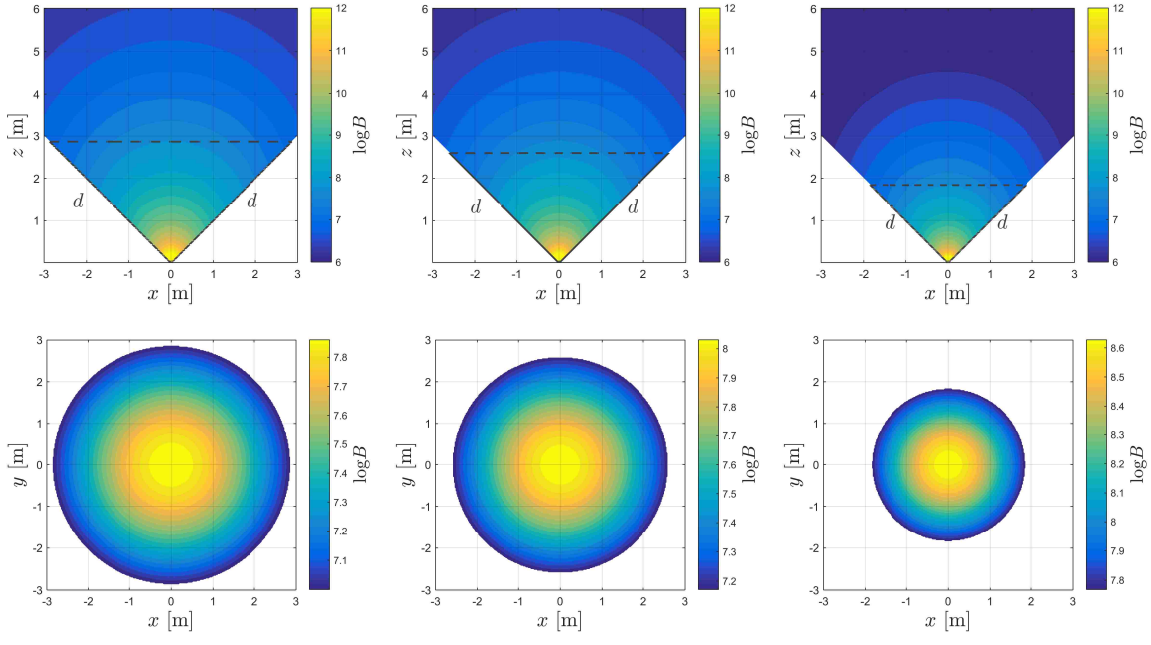


Figure 4.2: Contour maps of the logarithm of the bit rate assuming: (a) no pointing error, *i.e.*, $\phi = 0°$, (b) a misalignment of $\phi = 25°$, and (c) a misalignment of $\phi = 50°$. The range d for a bit rate of 10 Mbps computed applying (4.10) is shown for the three cases. The range d decreases as the pointing error ϕ increases. This effect can be seen in the upper plots. Also, as shown in the lower plots, the (x, y) area coverage decreases with increasing the pointing error.

Using (4.10), it is possible to find the range d for a desired bit rate B. Having this range d, we can calculate the height of the transmitter when the receiver pointing error is equal to its maximum, *i.e.*, $\psi = \Psi_{\mathcal{C}}$, given by

$$h = d \cos \Psi_{\mathcal{C}}. \tag{4.11}$$

Assuming a bit rate B equal to 10 Mbps, we compute the range d applying (4.10) and the height h from (4.11) for each case of pointing error, *i.e.*, $\phi = 0°, \phi = 25°$ and $\phi = 50°$. We assume that the height h between transmitter and receiver is fixed for generating the contours shown in the bottom plots of Figure 4.2 for each case.

In this scenario, we find that it is possible to achieve a bit rate of 10 Mbps at a range of $d \approx 4.05$ m for the case of $\phi = 0°$, see Figure 4.2(a). Using this range, we obtain $h \approx 2.87$ m. Notice that the bit rate increases as the transmitter becomes more directly aligned above the receiver. Perfect pointing results in the maximum bit rate achievable for the given parameters and desired maximum BER.

On the other hand, $d \approx 3.67$ m for the case of $\phi = 25°$ (Figure 4.2(b)) and $d \approx 2.61$ m assuming $\phi = 50°$ (Figure 4.2(c)). Then, we obtain $h \approx 2.59$ m and $h \approx 1.84$ m for $\phi = 25°$ and $\phi = 50°$, respectively. Thus, the model quantifies the trade-off in pointing accuracy and optical parameters such as power, effects of ambient light interference, range, and desired communications performance.

Implementing an OW link that supports a minimum data rate $B_{\min}$ requires a tradeoff between the maximum pointing error allowed $\phi_{\max}$ and the transmitter-to-receiver distance d. Moreover, $B_{\min}$ will be obtained at a different range d if the pointing error is zero, *i.e.*, $\phi = 0°$, or if a maximum pointing error is assumed, *i.e.*, $\phi = \phi_{\max} > 0°$. The shortest range $d_{\min}$ corresponds to $\phi = \phi_{\max}$, *i.e.*, to the worst case pointing error. Combining this range with the half-angle FOV of the receiver, $\Psi_{\mathcal{C}}$, it is possible to describe a right circular cone as shown in Figure 4.3(a). Hence, connectivity at a bit rate greater or equal to $B_{\min}$ would be sufficient guarantee if the transmitter lies in this cone assuming a pointing error less than $\phi_{\max}$. We call this

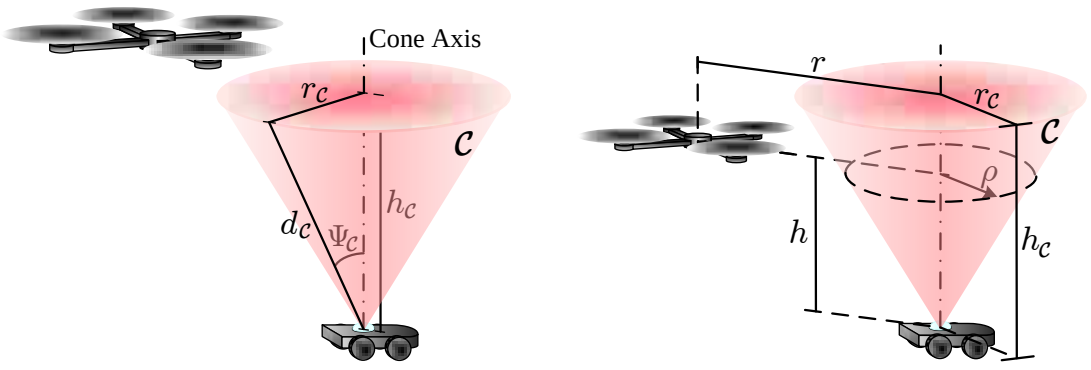


Figure 4.3: (a) Connectivity cone $\mathcal{C}$ and its parameters. (b) Comparing h with $h_{\mathcal{C}}$ and r with $r_{\mathcal{C}}$ is it possible to determine if the aerial transmitter is inside the connectivity cone $\mathcal{C}$.

the connectivity cone and denote it as $\mathcal{C}$. So, the tracking control problem addressed in this work requires a control law for the aerial transmitter such that it approaches and stays inside of $\mathcal{C}$. The apex or vertex of $\mathcal{C}$ is given by the position of the ground receiver $\mathbf{p}_{\text{Rx}}$. Furthermore, $\mathcal{C}$ can be defined by its slant height $d_{\mathcal{C}}$ and its apex angle $\Psi_{\mathcal{C}}$. Employing these parameters, it is easy to find the base radius of the cone $r_{\mathcal{C}}$ and the height of the cone $h_{\mathcal{C}}$, see Figure 4.3(a). Also, notice that the normal vector of the cone axis is given by $\mathbf{e}_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$.

4.2 Target Tracking to Establish an OW Communication Link

In this section, we consider the problem of controlling the flying optical transmitter such that it tracks the ground optical receiver and points the light beam to establish a temporary point-to-point OW communication link. The controller estimates the receiver state and tries to stay inside of a connectivity cone. While we focus on this particular set of assumptions, the control methods developed are readily adapted to

other cases. Next, we introduce the models for the ground optical receiver, the flying transmitter, and the noisy measurements.

4.2.1 Ground Optical Receiver (Rx)

Let $\mathbf{x} = \begin{bmatrix} x_{\text{Rx}} & y_{\text{Rx}} & \theta \end{bmatrix}^T \in \text{SE}(2)$ be the state of the ground receiver, where $\mathbf{p}_{\text{Rx}} = \begin{bmatrix} x_{\text{Rx}} & y_{\text{Rx}} & 0 \end{bmatrix}^T$ is the 3D position vector, see Figure 4.4(a), and θ is its heading angle. We assume that the motion of the ground receiver is given by the discrete-time nonlinear model

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \omega_t), \tag{4.12}$$

with

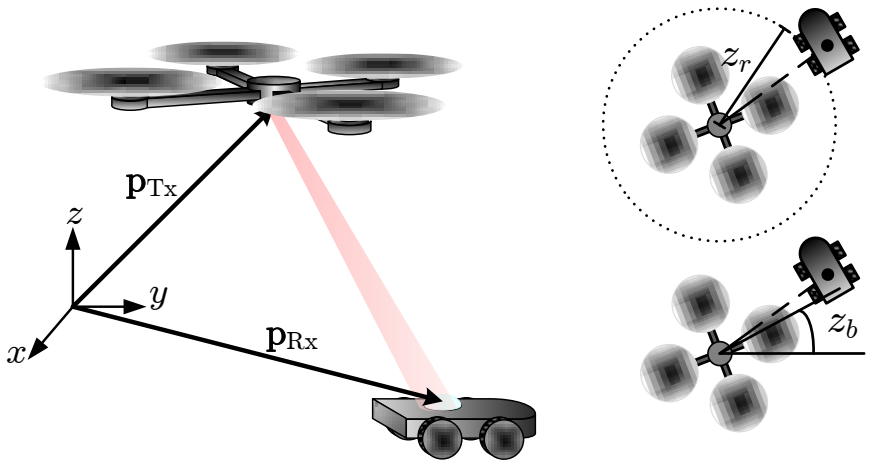
$$f(\mathbf{x}_t, \omega_t) = \mathbf{x}_t + \tau \begin{bmatrix} \text{v} \cos\theta_t \\ \text{v} \sin\theta_t \\ \omega_t \end{bmatrix}.$$

Here t is the time index, τ is the time step, v is the linear velocity that we assume is constant, and ω is the angular velocity control input. Thus, the mobile ground receiver moves with a constant linear velocity and can change its heading according to the control input ω.

4.2.2 Flying Optical Transmitter (Tx)

Let $\mathbf{p}_{\text{Tx}} = \begin{bmatrix} x_{\text{Tx}} & y_{\text{Tx}} & z_{\text{Tx}} \end{bmatrix}^T \in \mathbb{R}^3$ be the position of the flying transmitter, see Figure 4.4(a). We define its state as $\mathbf{y} = \begin{bmatrix} \mathbf{p}_{\text{Tx}} & \dot{\mathbf{p}}_{\text{Tx}} \end{bmatrix}^T \in \mathbb{R}^6$. We approximate the dynamics of the quadrotor as those of a point mass capable of accelerating in any direction [110], so the flying transmitter dynamics can be written in discrete time as

$$\mathbf{y}_{t+1} = \mathbf{A}\mathbf{y}_t + \mathbf{B}\mathbf{u}_t, \tag{4.13}$$



(a) (b)

Figure 4.4: (a) Position vectors of the aerial transmitter and of the ground receiver. (b) Noisy measurement of the distance to the ground receiver (Top) and noisy measurement of the direction to the ground receiver (Bottom).

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 & \tau & 0 \\ 0 & 0 & 1 & 0 & 0 & \tau \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \tau & 0 & 0 \\ 0 & \tau & 0 \\ 0 & 0 & \tau \end{bmatrix},$$

and $\mathbf{u} \in \mathbb{R}^3$ is the acceleration control input.

4.2.3 Measurement Model

We assume that the flying transmitter knows its own state without any error at all times. However, it obtains noisy sensor measurements at a frequency of $1/\tau$ Hz for estimating the state of the ground receiver. The sensor measurements are the ground

receiver heading angle, the distance and the direction to the ground receiver. The angular velocity measurement is modeled by

$$z_\omega = \omega + \eta_\omega, \tag{4.14}$$

where η_ω is Gaussian noise with zero mean and standard deviation σ_ω, *i.e.*, $\eta_\omega \sim \mathcal{N}(0, \sigma_\omega^2)$. Following [110], the measurement model for the range to the receiver can be written as

$$z_r = \sqrt{(x_{\mathrm{Rx}} - x_{\mathrm{Tx}})^2 + (y_{\mathrm{Rx}} - y_{\mathrm{Tx}})^2} + \eta_r, \tag{4.15}$$

where η_r is the range measurement noise such that $\eta_r \sim \mathcal{N}(0, \sigma_r^2)$. The model for the direction to the receiver is given by

$$z_b = \arctan\left(\frac{y_{\mathrm{Rx}} - y_{\mathrm{Tx}}}{x_{\mathrm{Rx}} - x_{\mathrm{Tx}}}\right) + \eta_b, \tag{4.16}$$

where $\eta_b \sim \mathcal{N}(0, \sigma_b^2)$ is the bearing measurement noise. We assume that the Gaussian random noise terms η_θ, η_r and η_b are mutually independent. The range measurement model given by (4.15) and the bearing measurement model given by (4.16) are depicted in Figure 4.4(b).

Letting $\mathbf{z} = \begin{bmatrix} z_r & z_b \end{bmatrix}^T$ and combining (4.15) and (4.16), we can write the following nonlinear measurement model

$$\mathbf{z} = h(\mathbf{x}, \mathbf{y}) + \boldsymbol{\eta}, \tag{4.17}$$

with

$$h(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \sqrt{(x_{\mathrm{Rx}} - x_{\mathrm{Tx}})^2 + (y_{\mathrm{Rx}} - y_{\mathrm{Tx}})^2} \\ \arctan\left(\frac{y_{\mathrm{Rx}} - y_{\mathrm{Tx}}}{x_{\mathrm{Rx}} - x_{\mathrm{Tx}}}\right) \end{bmatrix},$$

and $\boldsymbol{\eta} = \begin{bmatrix} \eta_r & \eta_b \end{bmatrix}^T$. Notice that by our definition of η_r and η_b, $\boldsymbol{\eta}$ is a Gaussian random vector with zero mean and covariance matrix

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_b^2 \end{bmatrix}.$$

4.2.4 Estimation and Control

Because of the noisy measurements, a solution to this pointing and tracking problem requires an estimation step before the computation of the transmitter motion control. We perform this estimation using particle filters [111–113] which have been successfully applied to tracking applications [110, 114].

Particle Filter

Particle filters are a Monte Carlo method to perform Bayesian estimation [113]. These recursive methods allow direct use of nonlinear motion dynamics, highly nonlinear sensor models, and arbitrary noise probability distributions. We employ a sampling-importance-resampling (SIR) particle filter [111, 112]. The application of this filter for our case is presented in Algorithm 4.1. We use the symbol "$\triangleright$" to denote comments in the algorithm statements.

The transmitter estimates the state of the ground receiver using

$$\tilde{\mathbf{x}} = \sum_{i=1}^{N} \mathrm{w}_i \hat{\mathbf{x}}_i, \tag{4.18}$$

where $\tilde{\mathbf{x}}$ is the estimate of $\mathbf{x}$, and $\{\hat{\mathbf{x}}_i, \mathrm{w}_i\}_{i=1}^{N}$ is the set of N particles. For this set, $\hat{\mathbf{x}}_i \in \mathrm{SE}(2)$ is the state of the i^{th} particle and $\mathrm{w}_i \in \mathbb{R}_{>0}$ is its importance weight. Equation (4.18) is the particle filter minimum mean square estimate of $\mathbf{x}$ [110, 111]. The set of particles is updated using Algorithm 4.1 where the new measurement $\mathbf{z}$ is incorporated into the collection of particles $\{\hat{\mathbf{x}}_i, \mathrm{w}_i\}_{i=1}^{N}$. This algorithm predicts the state of each particle (line 2), updates and normalizes the importance weights with the likelihood of new observations (lines 3 to 7), and then resamples the particles (lines 8 to 10).

For predicting the state of each particle, we use the model of the ground receiver given in (4.12) using the noise measurement of the angular velocity z_ω. To update and normalize the importance weights in the SIR particle filter, we require the probability

$p(\mathbf{z}|\mathbf{x},\mathbf{y})$. From (4.17), this probability is given by

$$p(\mathbf{z}|\mathbf{x},\mathbf{y}) = \mathcal{N}(\mathbf{z}; h(\mathbf{x},\mathbf{y},\mathbf{\Sigma})), \tag{4.19}$$

where $\mathcal{N}(\mathbf{s};\boldsymbol{\mu},\mathbf{\Gamma})$ is a multivariate Gaussian distribution with argument $\mathbf{s}$, mean $\boldsymbol{\mu}$, and covariance $\mathbf{\Gamma}$. The argument $\mathbf{s}$ and the mean $\boldsymbol{\mu}$ are $1 \times \mathrm{d}$ vectors, and the covariance $\mathbf{\Gamma}$ is a $\mathrm{d} \times \mathrm{d}$ matrix. This Gaussian probability distribution is

$$\mathcal{N}(\mathbf{s};\boldsymbol{\mu},\mathbf{\Gamma}) = \frac{1}{\sqrt{(2\pi)^{\mathrm{d}}|\mathbf{\Gamma}|}} \exp\left(-\frac{1}{2}(\mathbf{s}-\boldsymbol{\mu})^T\mathbf{\Gamma}^{-1}(\mathbf{s}-\boldsymbol{\mu})\right), \tag{4.20}$$

where $|\mathbf{\Gamma}|$ is the determinant of the matrix $\mathbf{\Gamma}$.

Algorithm 4.1 Particle Filter

Require: $\{\hat{\mathbf{x}}_i, \mathrm{w}_i\}_{i=1}^N$ the set of particles, z_ω and $\mathbf{z}$ the noisy measurements, $\mathbf{\Sigma}$ the covariance matrix, $\mathbf{y}$ the state of the flying transmitter, $\mathrm{N_{Th}}$ the resampling threshold

1: **for** $i = 1 : N$ **do**
2: $\quad \hat{\mathbf{x}}_i \leftarrow f(\hat{\mathbf{x}}_i, z_\omega)$ ▷ update particle state using (4.12)
3: $\quad p(\mathbf{z}_t|\hat{\mathbf{x}}_i,\mathbf{y}) \leftarrow \mathcal{N}(\mathbf{z}; h(\hat{\mathbf{x}}_i,\mathbf{y}),\mathbf{\Sigma})$ ▷ from (4.19) and (4.20)
4: $\quad \mathrm{w}_i \leftarrow \mathrm{w}_i p(\mathbf{z}|\hat{\mathbf{x}}_i,\mathbf{y})$
5: $\mathrm{W} \leftarrow \sum_{i=1}^N \mathrm{w}_i$
6: **for** $i = 1 : N$ **do**
7: $\quad \mathrm{w}_i \leftarrow \mathrm{w}_i/\mathrm{W}$ ▷ normalization
8: $\mathrm{N_{eff}} \leftarrow 1/\sum_{i=1}^N (\mathrm{w}_i)^2$
9: **if** $\mathrm{N_{eff}} < \mathrm{N_{Th}}$ **then**
10: $\quad \{\hat{\mathbf{x}}_i, \mathrm{w}_i\}_{i=1}^N \leftarrow \mathrm{resample}\{\hat{\mathbf{x}}_i, \mathrm{w}_i\}_{i=1}^N$

Output: $\tilde{\mathbf{x}} \leftarrow \sum_{i=1}^N \mathrm{w}_i \hat{\mathbf{x}}_i$ ▷ estimated state of the receiver

The use of resampling is required to mitigate the degeneracy problem of particle filters [111,112]. Broadly speaking, the degeneration phenomenon occurs when a few particles have a high concentration of the probability mass. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the estimation of $\mathbf{x}_t$ in (4.18) is almost zero. To estimate the level of degeneracy, we use the effective sample size $\mathrm{N_{eff}} = \frac{1}{\sum_{i=1}^N (\mathrm{w}_i)^2}$ [112]. If $\mathrm{N_{eff}}$ is less than a threshold $\mathrm{N_{th}}$ then a severe degeneracy occurs and resampling is required (line 10). In our simulations, we use systematic sampling [112]. For completeness, the steps for this

method are reproduced in Algorithm 4.2. For more details about particle filters and resampling methods, the reader is referred to [111–113].

Algorithm 4.2 Resample (adapted from [112])

Require: $\{\hat{\mathbf{x}}_i, \mathrm{w}_i\}_{i=1}^{N}$ the set of particles
1: Initialize CDF, $c_1 = 0$
2: **for** $i = 2 : N$ **do**
3: $\quad c_i \leftarrow c_{i-1} + \mathrm{w}_i \triangleright$ construct CDF
4: Initialize index, $i = 1 \triangleright$ start at the bottom of CDF
5: Generate $u_1 \sim \mathcal{U}[0, \frac{1}{N}] \triangleright$ uniform distributed initial value
6: **for** $j = 1 : N$ **do**
7: $\quad u_j \leftarrow u_1 + \frac{j-1}{N} \triangleright$ Move along the CDF
8: $\quad$ **while** $u_j > c_i$ **do**
9: $\quad\quad i \leftarrow i + 1$
10: $\quad \hat{\mathbf{x}}_j \leftarrow \hat{\mathbf{x}}_i \triangleright$ assign sample
11: $\quad \mathrm{w}_j \leftarrow \mathrm{w}_i \triangleright$ assign weight
Output: $\{\hat{\mathbf{x}}_j, \mathrm{w}_j\}_{j=1}^{N} \triangleright$ resampled set of particles

Tracking and Pointing

After applying the particle filter detailed in Algorithm 4.1, the transmitter has an estimate of the state of the ground receiver

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{x}_{\mathrm{Rx}} & \tilde{y}_{\mathrm{Rx}} & \tilde{\theta} \end{bmatrix}^T . \tag{4.21}$$

Thus, $\tilde{\mathbf{p}}_{\mathrm{Rx}} = \begin{bmatrix} \tilde{x}_{\mathrm{Rx}} & \tilde{y}_{\mathrm{Rx}} & 0 \end{bmatrix}^T$ is the estimated position of the receiver and $\tilde{\theta}$ is its estimated heading angle. Given $\tilde{\mathbf{p}}_{\mathrm{Rx}}$, the parameters of the optical transmitter and receiver (such as in Table 4.1), and the desired bit rate, we can estimate $\mathcal{C}$. We define the feedback control law for the aerial transmitter Tx as

$$\mathbf{u} = \nu \left(\dot{\mathbf{p}}_{\mathrm{Tx}} - \mathbf{v}_{\mathrm{ref}} \right) + \mathcal{F} \left(\mathbf{p}_{\mathrm{Tx}} \right), \tag{4.22}$$

where ν is a control gain, $\mathbf{v}_{\mathrm{ref}} = \begin{bmatrix} \mathrm{v} \cos\tilde{\theta} & \mathrm{v} \sin\tilde{\theta} & 0 \end{bmatrix}^T$ is the reference velocity, and $\mathcal{F}\left(\mathbf{p}_{\mathrm{Tx}}\right)$ is the artificial force to ensure that Tx approaches and stays inside the cone.

This force is given by

$$\mathcal{F}\left(\mathbf{p}_{\mathrm{Tx}}\right) = -\nabla \mathcal{U}\left(\mathbf{p}_{\mathrm{Tx}}\right), \tag{4.23}$$

with $\mathcal{U}\left(\mathbf{p}_{\mathrm{Tx}}\right)$ being an attractive potential function defined as

$$\mathcal{U}\left(\mathbf{p}_{\mathrm{Tx}}\right) = \frac{1}{2}\kappa \left\|\mathbf{p}_{\mathrm{Tx}} - \mathbf{p}_{\mathrm{Ref}}\right\|^2. \tag{4.24}$$

This type of potential function is widely used in robotics for controlling an agent to go to a desired goal position [20, 89, 114, 115]. Here, κ is a scaling factor, $\|\cdot\|$ is the Euclidean norm, and $\mathbf{p}_{\mathrm{ref}} \in \mathbb{R}^3$ is a reference point which is estimated depending if Tx is inside the connectivity cone or not. In order to determine this, we need to first compute two parameters: a) the height of Tx denoted as h, and b) the distance between Tx and the cone axis denoted as r. These parameters are illustrated in Figure 4.3(b) and they can be estimated by

$$\begin{aligned} h &= \left(\mathbf{p}_{\mathrm{Tx}} - \tilde{\mathbf{p}}_{\mathrm{Rx}}\right) \cdot \mathbf{e}_3, \text{ and} \\ r &= \left\|\mathbf{p}_{\mathrm{Tx}} - \tilde{\mathbf{p}}_{\mathrm{Rx}} - h\mathbf{e}_3\right\|. \end{aligned} \tag{4.25}$$

When h is greater than the height of the connectivity cone $h_{\mathcal{C}}$, then it is clear that Tx is outside of the cone. Otherwise, we need to compare r with the cone radius at the height h to check if Tx is inside the cone. This cone radius, denoted ρ, is given by

$$\rho = \frac{h}{h_{\mathcal{C}}} r_{\mathcal{C}}, \tag{4.26}$$

and is depicted in Figure 4.3(b). Recall that $r_{\mathcal{C}}$ is the base radius of the cone. If r is greater than ρ then the transmitter is outside the cone. Algorithm 4.3 summarizes the steps to determine if Tx is inside $\mathcal{C}$ or not.

Once we determine if the transmitter is outside or inside of $\mathcal{C}$, we calculate the reference position $\mathbf{p}_{\mathrm{ref}}$ required by the potential function (4.24). From our discussion in Section 4.1.1, we know that the OW link has a bit rate at least equal to $B_{\min}$ when Tx is within $\mathcal{C}$. Furthermore, having Tx inside of $\mathcal{C}$, the bit rate is enhanced as Tx

Algorithm 4.3 Inside Connectivity Cone

Require: $r_{\mathcal{C}}$ base radius of the cone, $h_{\mathcal{C}}$ height of the cone, h height of Tx respect to Rx, r distance between Tx and the cone axis

if $h \leq h_{\mathcal{C}}$ **then**
  $\rho \leftarrow (h/h_{\mathcal{C}})\, r_{\mathcal{C}}$ ▷ cone radius at height h
  if $r \leq \rho$ **then**
    InCone ← TRUE ▷ inside the cone
  else
    InCone ← FALSE ▷ outside the cone
else
  InCone ← FALSE ▷ outside the cone

Output: InCone

gets closer to the cone axis. The reference position in the cone axis can be estimated as

$$\mathbf{p}_{\text{axis}} = \tilde{\mathbf{p}}_{\text{Rx}} + h\mathbf{e}_3. \tag{4.27}$$

This point is shown in Figure 4.5(a) and we use it as the reference point when the transmitter is inside the connectivity cone, *i.e.*, $\mathbf{p}_{\text{ref}} = \mathbf{p}_{\text{axis}}$ if Tx $\in \mathcal{C}$. For this case, the pan and tilt angles are

$$\begin{aligned} \alpha &= \arctan \tfrac{\tilde{y}_{\text{Rx}} - y_{\text{Tx}}}{\tilde{x}_{\text{Rx}} - x_{\text{Tx}}}, \text{ and} \\ \beta &= \arcsin \tfrac{h}{\|\tilde{\mathbf{p}}_{\text{Rx}} - \mathbf{p}_{\text{Tx}}\|}. \end{aligned} \tag{4.28}$$

When Tx is outside of $\mathcal{C}$, we assume that the pan and tilt angles are both equal 0°. For this case, we also assume that the reference position is given by the closest point in the cone with respect to the position of Tx. We denote this point as $\mathbf{p}_{\mathcal{C}}$. Algorithm 4.4 describes the procedure to find $\mathbf{p}_{\mathcal{C}}$. When Tx is above the base of $\mathcal{C}$, *i.e.*, $h > h_{\mathcal{C}}$ and $r < r_{\mathcal{C}}$ (see Figure 4.5(b)), it is easy to compute the closest point

$$\mathbf{p}_{\mathcal{C}} = \mathbf{p}_{\text{Tx}} - (h - h_{\mathcal{C}})\, \mathbf{e}_3. \tag{4.29}$$

There does not appear to be a direct way to find the closest point in the cone if Tx is not above its base. However, we can determine the closest point and the shortest distance from a point to a triangle in 3D, *e.g.*, see [116]. Consequently, we decompose

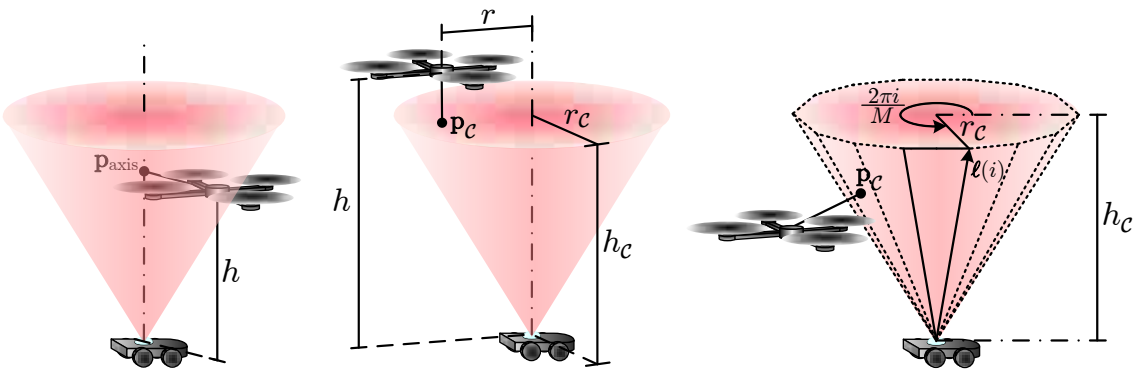


Figure 4.5: (a) Reference point is in the cone axis, $\mathbf{p}_{\text{axis}}$, when the aerial transmitter is inside the connectivity cone. (b) Closest point respect to the aerial transmitter, $\mathbf{p}_{\mathcal{C}}$, when it is right on top the base of the cone. (c) Closest point respect to the aerial transmitter, $\mathbf{p}_{\mathcal{C}}$, obtained by decomposing the cone into M triangles.

the cone into M triangles as shown in Figure 4.5(c). Then, we can find the closest point and its corresponding distance from $\mathbf{p}_{\text{Tx}}$ to each of the triangles by applying the algorithm in Section 10.3.2 of [116] and implemented in [117]. Let $i = 1, ..., M$,

Algorithm 4.4 Closest Point Cone

Require: $r_{\mathcal{C}}$ base radius of the cone, $h_{\mathcal{C}}$ height of the cone, $\tilde{\mathbf{p}}_{\text{Rx}}$ estimated position of Rx, $\mathbf{p}_{\text{Tx}}$ position of Tx, h height of Tx respect to Rx, r distance between Tx and the cone axis, M number of triangles which the cone is decomposed into

if $h > h_{\mathcal{C}}$ and $r < r_{\mathcal{C}}$ **then**
  $\mathbf{p}_{\mathcal{C}} \leftarrow \mathbf{p}_{\text{Tx}} - (h - h_{\mathcal{C}})\,\mathbf{e}_3$
else
  for $i = 1 : M$ **do**
    $\mathbf{a}(i) \leftarrow \tilde{\mathbf{p}}_{\text{Rx}}$
    $\mathbf{b}(i) \leftarrow \mathbf{a}(i) + \boldsymbol{\ell}(i-1)$
    $\mathbf{c}(i) \leftarrow \mathbf{a}(i) + \boldsymbol{\ell}(i)$
    $\mathbf{t}_i \leftarrow \{\mathbf{a}(i), \mathbf{b}(i), \mathbf{c}(i)\}$ ▷ i^{th} triangle
    $[\varrho_i, \mathbf{q}_i] \leftarrow$ Point Triangle Distance $(\mathbf{t}_i, \mathbf{p}_{\text{Tx}})$
  $[\varrho_{\text{close}}, \text{i}_{\text{close}}] \leftarrow \min(\varrho_1, \ldots, \varrho_i, \ldots, \varrho_M)$
  $\mathbf{p}_{\mathcal{C}} \leftarrow \mathbf{q}_{\text{i}_{\text{close}}}$

Output: $\mathbf{p}_{\mathcal{C}}$ ▷ closest point in the cone from Tx

then we define the i^{th} triangle $\mathbf{t}_i$ by the position of its vertices $\mathbf{a}(i), \mathbf{b}(i)$, and $\mathbf{c}(i)$ that are given by

$$\begin{aligned} \mathbf{a}(i) &= \tilde{\mathbf{p}}_{\text{Rx}}, \\ \mathbf{b}(i) &= \mathbf{a}(i) + \boldsymbol{\ell}(i-1), \text{ and} \\ \mathbf{c}(i) &= \mathbf{a}(i) + \boldsymbol{\ell}(i), \end{aligned} \tag{4.30}$$

where

$$\boldsymbol{\ell}(i) = \begin{bmatrix} r_{\mathcal{C}} \cos\left(\frac{2\pi i}{M}\right) & r_{\mathcal{C}} \sin\left(\frac{2\pi i}{M}\right) & h_{\mathcal{C}} \end{bmatrix}^T. \tag{4.31}$$

The i^{th} triangle is illustrated in Figure 4.5(c). Employing the algorithm in [117], we find the shortest distance ϱ_i and the closest point $\mathbf{q}_i$ from $\mathbf{p}_{\text{Tx}}$ to the triangle $\mathbf{t}_i = \{\mathbf{a}(i), \mathbf{b}(i), \mathbf{c}(i)\}$ for every $i = 1, \ldots, M$ (see line 9 in Algorithm 4.4). We then determine the index of the smallest element in the set $\{\varrho_1, \ldots, \varrho_M\}$ which we denote as $\mathrm{i}_{\text{close}}$ and the closest point for this case is

$$\mathbf{p}_{\mathcal{C}} = \mathbf{q}_{\mathrm{i}_{\text{close}}}. \tag{4.32}$$

Summarizing, the closest point in $\mathcal{C}$ with respect to the transmitter when it is outside of $\mathcal{C}$ is given by

$$\mathbf{p}_{\mathcal{C}} = \begin{cases} \mathbf{p}_{\text{Tx}} - (h - h_{\mathcal{C}})\,\mathbf{e}_3, & \text{if } h > h_{\mathcal{C}} \text{ and } r < r_{\mathcal{C}}, \\ \mathbf{q}_{\mathrm{i}_{\text{close}}}, & \text{otherwise.} \end{cases} \tag{4.33}$$

We take this point as the reference point when the transmitter is outside the connectivity cone, *i.e.*, $\mathbf{p}_{\text{ref}} = \mathbf{p}_{\mathcal{C}}$ if $\text{Tx} \notin \mathcal{C}$.

Considering (4.27) and (4.33), we can define the reference point $\mathbf{p}_{\text{ref}}$ required for the calculation of the artificial attractive force $\mathcal{F}(\mathbf{p}_{\text{Tx}})$ by applying (4.23) and (4.24), and therefore for computing the transmitter control input $\mathbf{u}$ by (4.22). Therefore, this reference point is given by

$$\mathbf{p}_{\text{ref}} = \begin{cases} \mathbf{p}_{\text{axis}} \text{ given by (4.27)}, & \text{if } \text{Tx} \in \mathcal{C}, \\ \mathbf{p}_{\mathcal{C}} \text{ given by (4.33)}, & \text{otherwise.} \end{cases} \tag{4.34}$$

The steps to generate the control input $\mathbf{u}$ for the aerial transmitter are summarized in Algorithm 4.5.

Algorithm 4.5 Tracking and Pointing Controller

Require: $\mathbf{p}_{\mathrm{Tx}}$ position of Tx, $\dot{\mathbf{p}}_{\mathrm{Tx}}$ velocity of Tx, z_ω and $\mathbf{z}$ noisy measurements, κ scaling factor, ν control gain

$\tilde{\mathbf{x}} \leftarrow$ Particle Filter $(z_\omega, \mathbf{z})$
Get $\tilde{\mathbf{p}}_{\mathrm{Rx}}$ and $\tilde{\theta}$ from $\tilde{\mathbf{x}}$
$h \leftarrow (\mathbf{p}_{\mathrm{Tx}} - \tilde{\mathbf{p}}_{\mathrm{Rx}}) \cdot \mathbf{e}_3$ ▷ height of Tx respect to Rx
$r \leftarrow \|\mathbf{p}_{\mathrm{Tx}} - \tilde{\mathbf{p}}_{\mathrm{Rx}} - h\mathbf{e}_3\|$ ▷ distance between Tx and the cone axis
InCone $\leftarrow$ Inside Connectivity Cone $(\tilde{\mathbf{p}}_{\mathrm{Rx}}, \mathbf{p}_{\mathrm{Tx}})$
if InCone = TRUE **then** ▷ tracking and pointing
    $\mathbf{p}_{\mathrm{ref}} \leftarrow \tilde{\mathbf{p}}_{\mathrm{Rx}} + h\mathbf{e}_3$
    $\alpha \leftarrow \arctan \frac{\tilde{y}_{\mathrm{Rx}} - y_{\mathrm{Tx}}}{\tilde{x}_{\mathrm{Rx}} - x_{\mathrm{Tx}}}$ ▷ pan angle
    $\beta \leftarrow \arcsin \frac{h}{\|\tilde{\mathbf{p}}_{\mathrm{Rx}} - \mathbf{p}_{\mathrm{Tx}}\|}$ ▷ tilt angle
else▷ no pointing and get closer to the cone
    $\mathbf{p}_{\mathrm{ref}} \leftarrow$ Closest Point Cone $(\tilde{\mathbf{p}}_{\mathrm{Rx}}, \mathbf{p}_{\mathrm{Tx}})$
    $\alpha, \beta \leftarrow 0^\circ$ ▷ since we assume the beam is deactivated
$\mathcal{U}(\mathbf{p}_{\mathrm{Tx}}) \leftarrow \frac{1}{2}\kappa \|\mathbf{p}_{\mathrm{Tx}} - \mathbf{p}_{\mathrm{Ref}}\|^2$
$\mathcal{F}(\mathbf{p}_{\mathrm{Tx}}) \leftarrow -\nabla\mathcal{U}(\mathbf{p}_{\mathrm{Tx}})$
$\mathbf{v}_{\mathrm{ref}} \leftarrow \begin{bmatrix} \mathrm{v}\cos\tilde{\theta} & \mathrm{v}\sin\tilde{\theta} & 0 \end{bmatrix}^T$
Output: $\mathbf{u} \leftarrow \nu(\dot{\mathbf{p}}_{\mathrm{Tx}} - \mathbf{v}_{\mathrm{ref}}) + \mathcal{F}(\mathbf{p}_{\mathrm{Tx}})$

Table 4.2: Simulation parameters

Measurements	$\sigma_r = \sigma_b = \sigma_\omega = \sqrt{0.5}$
Particle Filter	$N = 2000$, $N_{\mathrm{Th}} = \frac{1}{2}N$
Controller	$\mathrm{K_p} = \mathrm{diag}(30, 30, 35)$, $\mathrm{K_v} = \mathrm{diag}(8.5, 8.5, 8.5)$

4.2.5 Simulation Results

The methodology and algorithms proposed in the previous section are validated by running numerical simulations using MATLAB®. A 3D environment developed for visualization purposes is shown in Figure 4.6. The aerial transmitter is represented as a thin cross with four circles at each side, while the mobile ground receiver is depicted as a cuboid with a sensing field-of-view in its front. We have drawn in red the connectivity cone $\mathcal{C}$ on top of the receiver, the reference position $\mathbf{p}_{\mathrm{ref}}$ used in Algorithm 4.5 by an asterisk marker "*", and the distance between the aerial transmitter and $\mathbf{p}_{\mathrm{ref}}$ by a dash-dot line "–·", see Figure 4.6(a). When the flying

transmitter is within $\mathcal{C}$, the transmit light beam is activated and depicted in the 3D environment, see Figure 4.6(b).

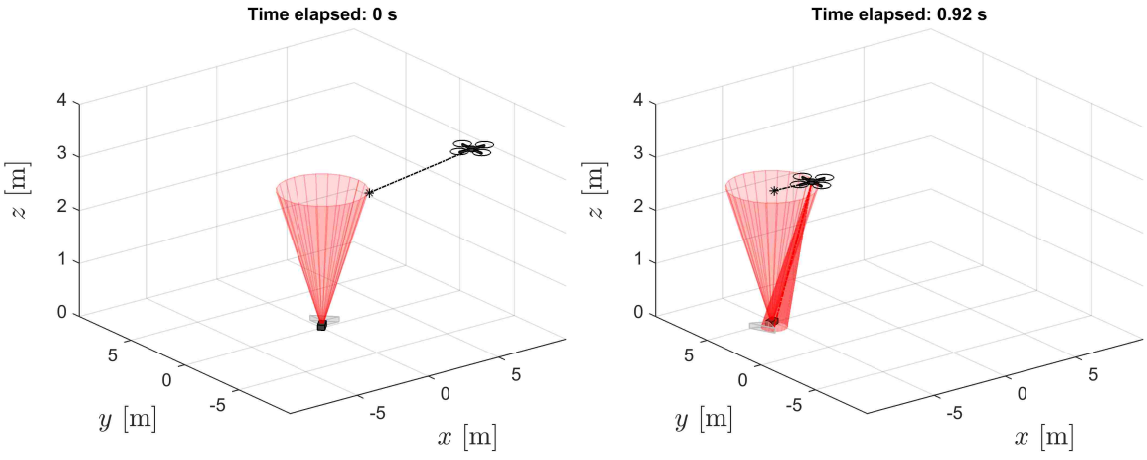
Estimation and Control Results

We first run a set of simulations where the initial position of the aerial robot is $\begin{bmatrix} 6 & -6 & 3.5 \end{bmatrix}^T$ m and the mobile ground robot starts at the origin with a heading angle $\theta = 45°$. Thus, the transmitter-receiver distance at the beginning is approximately 9.18 m. The 3D simulation environment at time zero is illustrated in Figure 4.6(a). The parameters for the optical link used for simulation are given in Table 4.1 that we employed to plot the contour maps in Figure 4.2. From our discussion in Section 4.1.1, we know that the parameters for $\mathcal{C}$ assuming a minimum desired bit rate $B_{\min} = 10$ Mbps and a maximum pointing error $\phi_{\max} = 25°$ are

$$\Psi_{\mathcal{C}} = 45° \quad \triangleright \text{apex angle},$$
$$d_{\mathcal{C}} \approx 3.67\,\text{m} \quad \triangleright \text{slant height},$$
$$r_{\mathcal{C}} = d_{\mathcal{C}} \sin \Psi_{\mathcal{C}} \approx 2.59\,\text{m} \quad \triangleright \text{base radius},$$
$$h_{\mathcal{C}} = d_{\mathcal{C}} \cos \Psi_{\mathcal{C}} \approx 2.59\,\text{m} \quad \triangleright \text{cone height}.$$

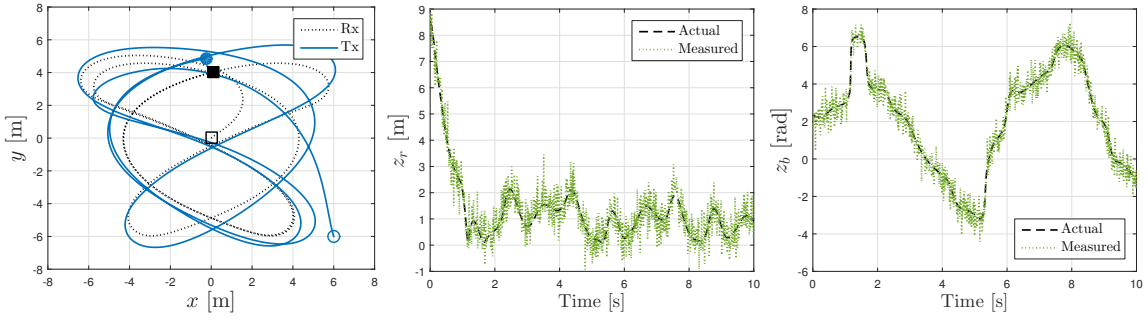
These are the values used to draw the connectivity cone in the 3D environment (Figure 4.6). The constant linear velocity of the mobile ground robot is assumed equal to 12 m/s. Also, we divide $\mathcal{C}$ into $M = 25$ triangles for Algorithm 4.4 which finds the closest point in $\mathcal{C}$ with respect to the flying transmitter. The time step τ is equal to 0.01 s and we run the simulations for 10 s. The rest of the parameters for the simulation are summarized in Table 4.2.

For the simulations, the ground receiver follows the trajectory shown in Figure 4.7(a) by the black dotted line. This figure also illustrates the xy trajectories of the optical receiver (Rx) and the optical transmitter (Tx). The empty square and circle denotes the initial xy position of Rx and Tx, respectively. The filled square and circle indicates the position of Rx and Tx after 10 seconds, respectively.



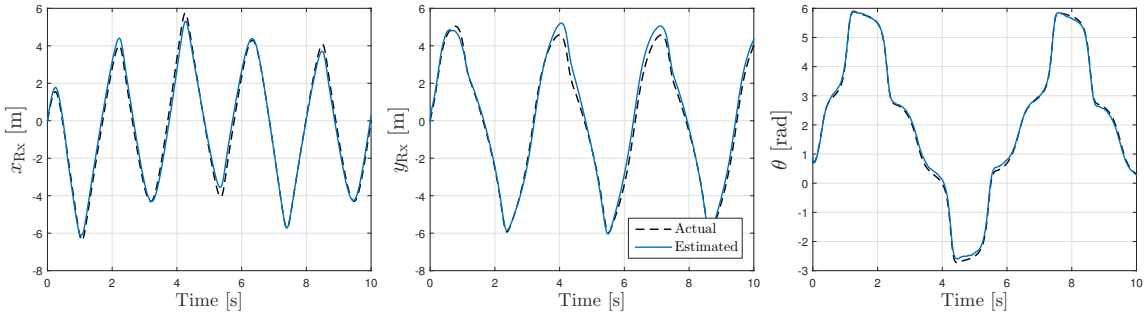
(a) (b)

Figure 4.6: 3D environment for numerical simulations. The connectivity cone $\mathcal{C}$ on top of the receiver is shown in red, the reference position $\mathbf{p}_{\text{ref}}$ is denoted by an asterisk marker "*", and the distance between the aerial transmitter and $\mathbf{p}_{\text{ref}}$ by a dash-dot line "–·". (a) The aerial transmitter starts outside of $\mathcal{C}$. (b) The transmit beam is activated once the transmitter is inside of $\mathcal{C}$.



(a) (b) (c)

Figure 4.7: (a) xy trajectories described by the mobile ground receiver (Rx) and the aerial transmitter (Tx). The empty square and circle denote the initial xy position of Rx and Tx, respectively. The filled square and circle indicate the position of Rx and Tx after 10 seconds, respectively. (b) The estimated and actual range z_r, *i.e.*, the distance from Tx to Rx. (c) The estimated and actual bearing z_b, *i.e.*, the direction from TX to the receiver.

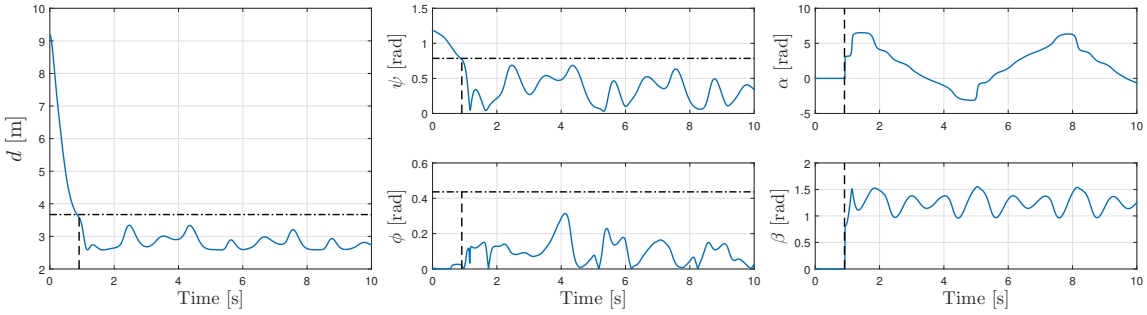


(a) (b) (c)

Figure 4.8: Actual and estimated values of the receiver state $\mathbf{x} = [x_{\text{Rx}} \quad y_{\text{Rx}} \quad \theta]^T$. The estimation is via the particle filter summarized in Algorithm 4.1. The maximum errors between the actual and estimated values are: 0.66 m for the x coordinate, 0.91 m for the y coordinate, and 0.15 rad for θ.

We generate the noisy measurements z_r, z_b and z_ω, feeding these to the tracking and pointing controller given in Algorithm 4.5 to estimate the state of the receiver and control the aerial transmitter. The range z_r given by (4.15) is depicted in Figure 4.7(b) and the bearing given by (4.16) is shown in Figure 4.7(c). For these figures, the actual values (without noise) are plotted with the black dashed line while the noisy measurements are drawn with the green dotted line. We employ the particle filter given in Algorithm 4.1 to estimate the state of the receiver. The results obtained from applying the particle filter are illustrated in Figure 4.8. The black dashed line denotes the actual values while the blue continuous line shows the estimated values. For the position of the receiver, the maximum errors between the actual and estimated values are around 0.66 m for the x coordinate and 0.91 m for the y coordinate. For the heading angle θ, the max error is 0.15 rad. These errors, especially for the position, will cause an error in pointing at the receiver which will be reflected in the pointing angle ϕ.

The transmitter-receiver distance d is shown in Figure 4.9(a), and the receiver pointing error ψ and the pointing error ϕ are given in Figure 4.9(b). The slant height



(a) (b) (c)

Figure 4.9: (a) The distance d between transmitter and receiver. The dash-dot line is the value of $d_{\mathcal{C}}$. (b) (Upper) The receiver pointing error ψ. The dash-dot line is the value of $\Psi_{\mathcal{C}}$. (Lower) The pointing error ϕ. The dash-dot line is the maximum value of pointing error assumed $\phi_{\max}$. (c) The pan α and tilt β angles. In all the figures, the dashed line indicates the elapsed time of 0.92 s that is the instant when the transmitter reaches the connectivity cone and therefore stays inside.

of the connectivity cone $d_{\mathcal{C}}$ is drawn by the dash-dot line in Figure 4.9(a). Similarly, the apex angle value $\Psi_{\mathcal{C}}$ and the maximum pointing error $\phi_{\max}$ are represented in the upper and lower plots of Figure 4.9(b), respectively. Notice that after around 0.92 s, d is less than $d_{\mathcal{C}}$ and it remains less than this value during the rest of the simulation. Also, ψ stays less than $\Psi_{\mathcal{C}}$ after 0.92 s. From this time on, the aerial transmitter remains within the connectivity cone. This time is outlined in the plots by a dashed black line and Figure 4.6(b) is a snapshot of the 3D simulation environment at this time. From the plot of ϕ in Figure 4.9(b), it is clear that this angle is always less than the assumed maximum pointing error $\phi_{\max} = 25°$. Figure 4.9(c) shows the pan and tilt angles, α and β respectively. As we explain in Section 4.2.4, these angles are assumed equal to zero when the flying transmitter is outside of $\mathcal{C}$. Once the transmitter is within $\mathcal{C}$, these angles are computed according to (4.28) to point the light beam at the receiver.

Because the transmitter remains inside the cone after 0.92 s, the bit rate should exceed the minimum desired bit rate of 10 Mbps after this time. The bit rate

B is illustrated in Figure 4.10(a) where the desired minimum limit of 10 Mbps is represented by the dash-dot line. B stays over the 10 Mbps once the 0.92 s have elapsed. The minimum bit rate after this time is 17.38 Mbps with an average bit rate of 78.31 Mbps. Consequently, our goal of establishing and maintaining an optical link of at least 10 Mbps is accomplished after 0.92 s.

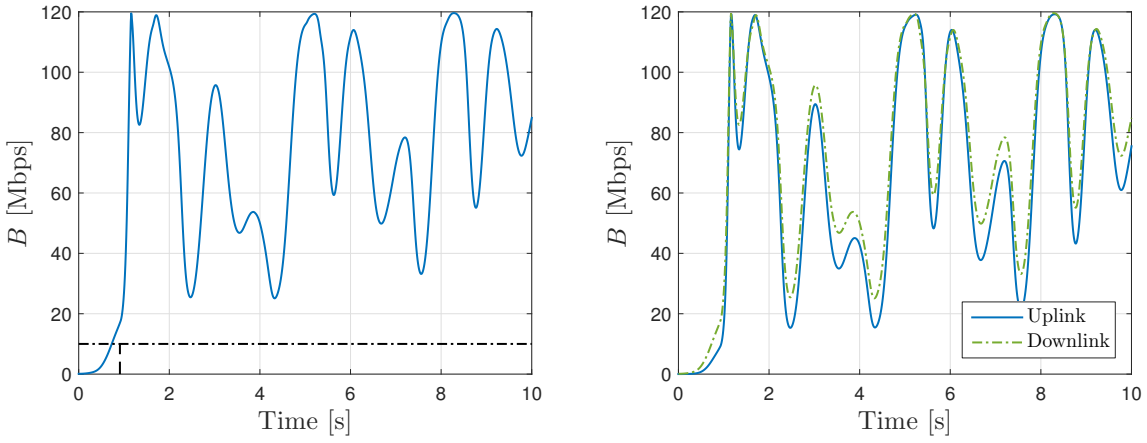


Figure 4.10: (a) The bit rate in Mbps calculated applying (4.9). Once the transmitter remains inside the connectivity cone $\mathcal{C}$, which is after 0.92 s, the minimum bit rate is 17.38 Mbps. Also, the average bit rate is 78.31 Mbps with a standard deviation of 28.469 Mbps. (b) The bit rate in Mbps of the uplink and downlink. Once the transmitter remains inside the connectivity cone, the minimum bit rate for the uplink is $\approx$ 9 Mbps, and the average bit rate is 71.29 Mbps with a standard deviation of 32.44 Mbps. The results for the downlink are given in Figure 4.10(a).

Uplink Bit Rate

So far, we have assumed that a directed optical transmitter is mounted on the aerial robot, while the ground vehicle is equipped with a fixed optical receiver always pointing up. Assuming that a fixed optical transmitter is now in the ground vehicle and a directed optical receiver in the aerial robot, the only modifications required in the optical link illustration given in Figure 4.1(b) are to interchange the angles $\Phi_{1/2}$

with Ψ_c and ϕ with ψ. Taking into account these modifications, it is straightforward to use the equations presented in Section 4.1 which characterize the model of the optical link. Therefore, we can compute the bit rate of the potential uplink between the agents using (4.9). The result for the case of the simulation experiment explained in the previous section is shown in Figure 4.10(b).

In general, the uplink has a lower bit rate than the downlink. For example, the uplink has a minimum bit rate of 9.04 Mbps once the aerial robot enters and remains within the connectivity cone. Thus, the uplink minimum bit rate is lower than the one for the downlink and it does not satisfy the requirement of minimum 10 Mbps assumed for defining the cone. However, this only occurs for a short time when the flying vehicle enters the cone. While it stays in the cone, the average bit rate is 71.29 Mbps with a standard deviation of 32.44 Mbps. Consequently, our approach can maintain not only a potential high bit rate downlink but also a reliable uplink.

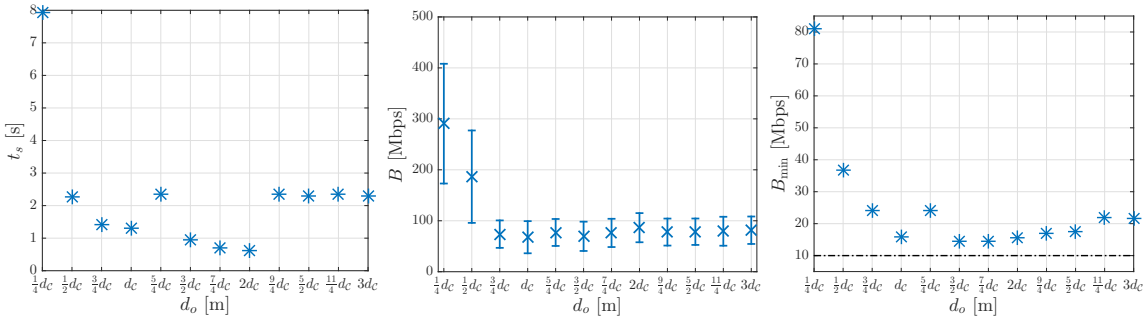
Changing the Transmitter Initial Position

In order to test the robustness of our approach with respect to the initial position of the flying transmitter, we run a second set of simulations where we generate randomly different initial positions. To generate these positions, we assume that the initial transmitter-receiver distance, d_o, goes from $\frac{1}{4}d_{\mathcal{C}}$ up to $3d_{\mathcal{C}}$ in steps of $\frac{1}{4}d_{\mathcal{C}}$. Recall that $d_{\mathcal{C}}$ is the slant height of the connectivity cone. This assumption gives us a total of 12 random starting positions. None of the positions has to be inside the initial cone whose vertex is located at the origin. The rest of the simulation parameters are the same as the ones used for the first set of simulations. For each one of the 12 cases, we compute:

- t_s, the time at which the transmitter enters the connectivity cone without leaving it again,
- B_{avg} and B_{std}, the average and the standard deviation of the bit rate, and

- B_{min}, the minimum bit rate.

We calculate B_{avg}, B_{std} and B_{min} considering the outcomes from t_s up to the final simulation time of 10 s. The results for the 12 cases are illustrated in Figure 4.11. We show B_{avg} and B_{std} in the same plot, see Figure 4.11(b). The cross marker "$\times$" represents B_{avg} while the bar denotes B_{std}.

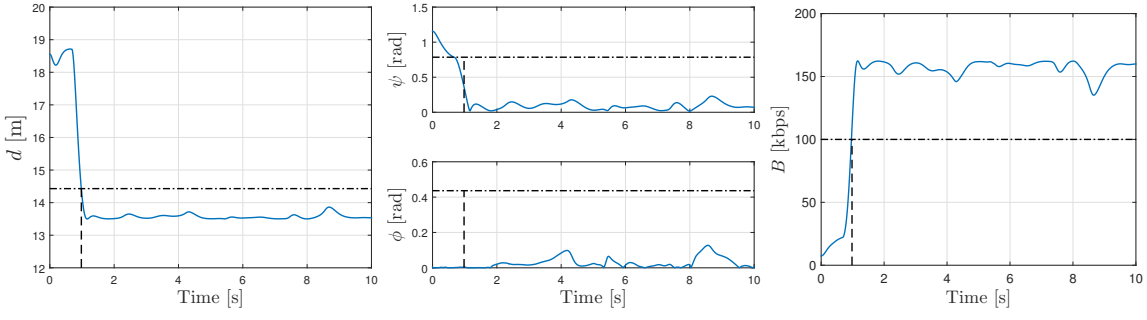


(a) (b) (c)

Figure 4.11: Results obtained generating random starting positions for the aerial transmitter such that they have an initial distance d_o respect to the ground receiver. (a) Time at which the transmitter stays inside $\mathcal{C}$ without leaving it again. (b) Average and standard deviation of the bit rate. The cross marker "$\times$" represents the average value and the bar denotes the standard deviation. (c) Minimum bit rate for each case. The desired minimum bit rate of 10 Mbps is illustrated by the dash-dot line.

From Figure 4.11(a), the transmitter needs more time to get and stay inside the connectivity cone when it starts from a position closer to the receiver than when it starts from a distance far from the cone. For example, $t_s = 7.92$ s for an initial transmitter-receiver distance of $\frac{1}{4}d_{\mathcal{C}}$ while $t_s = 0.63$ s for an initial distance of $2d_{\mathcal{C}}$. This behavior is because the volume of the connectivity cone $\mathcal{C}$ is smaller as we get closer to its apex, so the transmitter more easily departs from the cone when the receiver turns. After leaving the cone, Algorithm 4.4 computes a reference point $\mathbf{p}_{\text{ref}}$ so that the transmitter approaches to $\mathcal{C}$ reaching a higher altitude. After some iterations, this calculation of $\mathbf{p}_{\text{ref}}$ ensures that the transmitter stays within $\mathcal{C}$.

In contrast, the average bit rate and the minimum bit rate are higher when the transmitter starts at a position closer to the cone. For example, $B_{\text{avg}} = 290.58$ Mbps and $B_{\text{min}} = 81.01$ Mbps for $\frac{1}{4}d_{\mathcal{C}}$ whereas that $B_{\text{avg}} = 86.41$ Mbps and $B_{\text{min}} = 15.76$ Mbps for $2d_{\mathcal{C}}$. We expect to get this result according to the contour maps shown in Figure 4.2. Notice that the minimum bit rate is never less than 10 Mbps for any of the 12 cases, see Figure 4.11(c). Consequently, an intermittent high rate optical communication will be obtained if the aerial transmitter tries to stays closer to the mobile ground receiver. Or, a more continuous optical wireless communication but with low rate can be established if the transmitter-receiver distance is kept close to the slant height of the connectivity cone.



(a) (b) (c)

Figure 4.12: Results assuming a beacon optical link, *i.e.*, a low-rate optical link. (a) The transmitter-receiver range. (b) The receiver pointing error ψ (Upper) and the pointing error ϕ (Lower). (c) The potential bit rate B. The transmitter remains inside the connectivity cone after 1 s of starting the simulation (dashed black line). Notice that the bit rate is more stable than for the case of high-rates shown in Figure 4.10(a).

Low Rate Optical Link

The use of a wide-beamwidth optical link acting as a support for a more focused, narrow-beam link has been proposed as a OW communication system [106]. The wide-beam link, known also as *beacon link*, is designed to provide low-data-rate con-

nectivity that can be used for providing positioning and alignment control of a high data throughput narrow-beam link. To consider a similar scenario, we assume that the proposed OW link is used as a lower rate beacon link to enable control and feedback for some other autonomous task. Consequently, we need to select an acceptable low beacon bit rate for control purposes. A common wireless communication for mobile robotic operations is ZigBee [118]. This technology is based on the IEEE 802.15.4 standard [119] and its bit rate is in the range of 10 kbps to 250 kbps. Thus, we choose 100 kpbs as acceptable data rate for the acquisition and positioning control. Assuming a $\phi_{\max} = 25°$ and the specifications in Table 4.1, we can find the parameters of the connectivity cone to be

$$\begin{aligned}
&\Psi_{\mathcal{C}} = 45° && \triangleright \text{ apex angle,}\\
&d_{\mathcal{C}} \approx 14.43\,\text{m} && \triangleright \text{ slant height,}\\
&r_{\mathcal{C}} = d_{\mathcal{C}} \sin\Psi_{\mathcal{C}} \approx 10.20\,\text{m} && \triangleright \text{ base radius,}\\
&h_{\mathcal{C}} = d_{\mathcal{C}} \cos\Psi_{\mathcal{C}} \approx 10.20\,\text{m} && \triangleright \text{ cone height.}
\end{aligned}$$

In the following, all the other parameters are the same as those in Section 4.2.5. The results for the transmitter-receiver range d, the pointing error ϕ, the receiver pointing error ψ, and the potential bit rate B are shown in Figure 4.12.

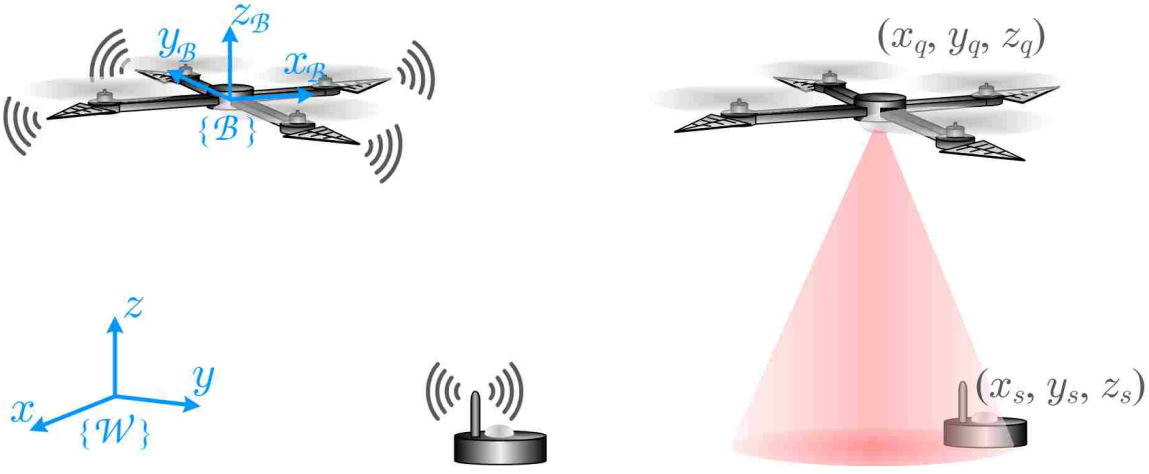
The transmitter enters the connectivity cone $\mathcal{C}$ after 1 s. This time instant is indicated by the dashed black line in all the plots of Figure 4.12. From the results for the range d (Figure 4.12(a)) and the receiver pointing error ψ (Figure 4.12(b)), the transmitter stays inside the cone after 1 s. Furthermore, the pointing error ϕ is kept below the desired maximum error of 25°. The bit rate B shown in Figure 4.12(c) remains above 100 kbps once the transmitter is within $\mathcal{C}$. Note that the potential bit rate has much less variation than in the previous results (compare with Figure 4.10(a)). The average bit rate is 156.75 kbps with a standard deviation of 6.50 kbps. This example illustrates how our tracking and pointing can be used to maintain a very stable low rate beacon link, that could be employed to provide control feedback for other autonomy tasks, such as sensing or networking optimization.

4.3 Sensor Localization Using a Hybrid RF/OW Communication Link

A common assumption for a wireless sensor network (WSN) is that the sensors are fully connected, *i.e.*, there is a communication path between any two nodes in the network. However, this type of connectivity generally requires a very dense number of sensors or additional communication relays which can be impractical or expensive to implement [120]. Furthermore, the available communication rate may be insufficient to deliver the information collected by the sensors when the amount of data is large, such as the case of high-resolution images or video.

Aerial robotic vehicles have the potential to alleviate these communication challenges by exploiting their mobility to gather information from in-situ wireless sensors. These information collectors are sometimes referred as *data mules* [121]. Also, large-scale data transfers can be achieved by combining the broadcast capability of radio frequency (RF) with the high rate capacity of optical wireless (OW) links. Hybrid RF/OW links can offer temporary high throughput point-to-point communications within fixed and mobile wireless networks [28, 122]. In this hybrid scheme, the RF component is generally employed for link control, maintenance and backup functions, while the OW component is used for bulk data transfer. Thus, RF can be exploited to assist in the pointing and acquisition of the OW link because it is both pervasive and independent.

We consider that both the sensor and the aerial data mule are equipped with a hybrid RF/optical communication system - RF for low bandwidth transmission and optical for high rate transfer. Our concept is illustrated in Figure 4.13(a), depicting a quadrotor employing radio signal strength to move towards a sensor node. Once the quadrotor is within a desired range, see Figure 4.13(b), the optical link can be employed to carry out bulk data transfer. Therefore, the flying robot has to localize the sensor in order to upload/download data while staying within



(a) (b)

Figure 4.13: (a) Radio-based communication is used by a quadrotor to move closer to a fixed sensor. (b) Once within range, optical wireless communication can be employed for high-data transfer.

optical communication range. We develop a solution to autonomously localize the ground sensor node relative to the aerial vehicle assuming that the sensor position is unknown at all times. We exploit the hybrid communication scheme in order to solve this problem.

We assume that a sensor node deployed in an open environment is equipped with a hybrid RF/optical communication system. The RF component is used for relatively low data rate, while the optical component can manage high data transmission. We assume that a quadrotor has a similar hybrid wireless transceiver and it is able to communicate with the sensor when in range. The sensor uses an omnidirectional antenna, while the quadrotor has one directional antenna at the end of each one of its arms, see Figure 3.4(a). Figure 3.4(b) shows the gain pattern of a directional antenna with a 45° main lobe. With this sectorization, the RSS can be measured at each antenna, a rough angle of arrival estimated, and the quadrotor can move towards the source.

Starting with the quadrotor at a significant distance from the sensor but within the RF range, our goal is that the aerial robot comes close to the sensor node such that the optical link can be employed to upload the data collected by the sensor. We assume that the exact location of the sensor is unknown to the quadrotor at all times. Our approach uses the RSS measurements of the four antennas to guide the flying vehicle towards the sensor. Once the quadrotor comes close enough to the sensor, it is able to detect the optical signal and hover within optical communication range. At this stage, the data transfer can be performed employing the optical link. In the next section, we detail the channel model for the hybrid RF/OW communication system.

4.3.1 Hybrid RF/OW Channel Model

Given their independent transmission modalities, we consider separate channel models for the RF and OW links to predict the hybrid system performance.

RF link

We assume an open environment, so that it is possible to achieve LOS between the air vehicle and the sensor node. We consider a multipath fading environment, *e.g.*, with microwave radio. Therefore, we adopt a log-distance path-loss model, see Section 3.2.1, to describe the power loss between the sensor node (acting as transmitter Tx) and the quadrotor (acting as receiver Rx). For this model, the received signal strength of the i^{th} quadrotor's antenna expressed in dB is given by

$$P_{\text{RF}_i} = P_{\text{Tx}} - P_{o_i} - 10\kappa \log_{10}\left(\frac{d_i}{d_o}\right) - \chi, \tag{4.35}$$

where $i = 1, \ldots, 4$ indexes the four antennas. In (4.35), P_{Tx} is the power transmitted by the sensor node, d_o is the reference distance, d_i is the distance between the sensor node and the i^{th} quadrotor antenna, κ is the path-loss exponent, P_{o_i} is the reference

path loss, and χ is a zero-mean Gaussian random variable reflecting the attenuation (in dB) caused by flat fading.

Typical values for d_o are 1-10 m indoors and 10-100 m outdoors. The value of the path-loss exponent κ depends on the propagation environment. Its value ranges from 2 for free-space up to 6 for heavily cluttered environments. The reference path loss P_{o_i} can be calculated at the reference distance d_o by applying the Friis transmission equation

$$P_{o_i} = 10 \log_{10} \left[\frac{1}{G_{\mathrm{Tx}} G_{\mathrm{Rx}_i}} \left(\frac{4\pi d_o}{\lambda} \right)^2 \right], \tag{4.36}$$

where G_{Tx} is the gain of the sensor's antenna, G_{Rx_i} is the gain of the i^{th} quadrotor's antenna, and $\lambda = c/f$ is the wavelength of the transmitted signal (c is the speed of light and f is the communication frequency in Hertz). We set $G_{\mathrm{Tx}} = 1$ since we assume an omnidirectional antenna for the sensor node. On the other hand, G_{Rx_i} can take on a value ≤ 1 to account for the directivity of the quadrotor's antennas as in Figure 3.4(b).

One way to mitigate fading is to increase the transmission power, but this is not always possible due to size, weight and energy limitations. An alternative is the use of multiple transmitting and/or receiving antennas. This technique is known as *antenna diversity*, see Section 3.2.2. Multiple antennas with sufficient spacing provide the advantage that while one receiving antenna is in a deep fade, it is probable that another one has sufficient signal strength. At microwave frequencies, an antenna spacing of at least one wavelength λ has been experimentally shown to result in approximately decorrelated fading [123]. For example, the operating frequency for Zigbee is 2.4 GHz, so its wavelength is $\lambda = 12.5$ cm. An AscTec Hummingbird quadrotor [77] has an arm length of 17 cm. Thus, the minimum separation between antennas placed in each arm for this quadrotor is ≈ 24 cm $\approx 2\lambda$. This spacing will yield roughly independent RSS measurements.

Optical Link

In Section 4.1, we introduced a model of a directional LOS optical link between an aerial and a ground robot. Based on this model, we designed a controller to place the aerial vehicle in optical range to maintain a minimum bit rate. We defined the area within optical range as a *connectivity cone*. A diagram showing the main parameters for the optical link model is illustrated in Figure 4.1(b). Using (4.1) and (4.2), the optical signal strength in dB at the receiver is given by

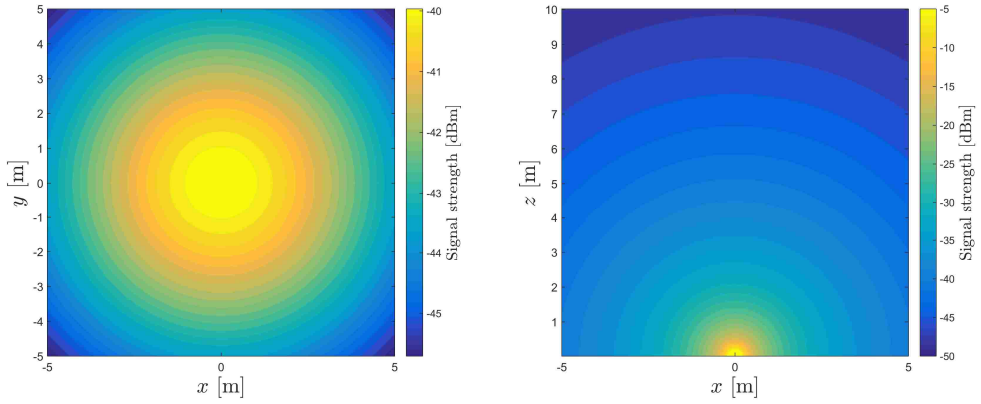
$$P_{\text{OW}} = 10 \log_{10} (I_s A_{\text{eff}}) + \zeta, \tag{4.37}$$

where ζ represents the average optical power (in dB) produced by natural and artificial light sources. This background radiation is generally modeled as being spectrally constant, Gaussian, and independent of the received signal [29]. In addition, we assume that the optical transceiver mounted on the quadrotor is always pointing down, while the one in the sensor node is always pointing up. Under this assumption, $\phi = \psi$ at all times. A generalization enables mechanical pointing, such as with a gimbal as it is illustrated in Figure 4.1(a).

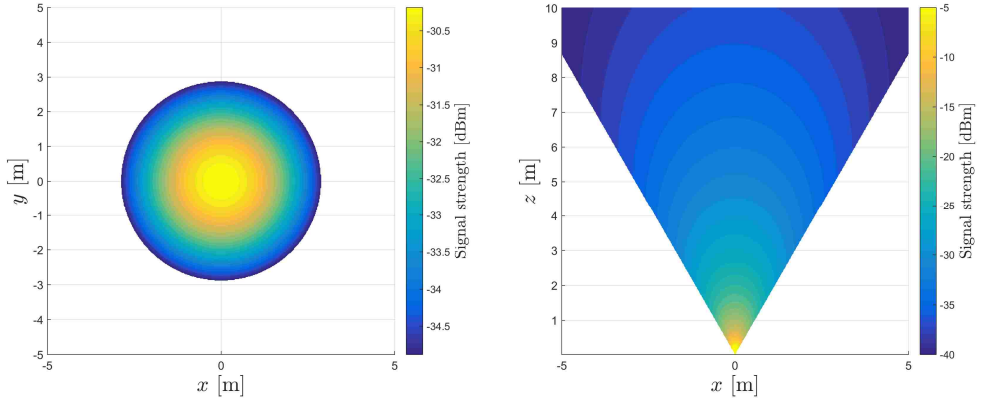
Contour Maps

Using (4.35) for the RF link and (4.37) for the OW link, we can plot the signal strength as a function of the receiver position relative to the transmitter. Contour maps of the signal strength are shown in Figure 4.14(a) and in Figure 4.14(b) for the RF and OW components, respectively. For each component, we plot the signal strength for the azimuth plane (x-y plane) and for the elevation plane (x-z plane) assuming that the transmitter is fixed at the origin. For illustration, these plots are noise free, *i.e.*, we do not consider the term χ for (4.35) and ζ for (4.37). The rest of the parameters are summarized in Table 4.3.

The white regions in the contour maps for the case of the OW link (Figure 4.14(b)) indicate that no optical signal can be detected in these areas. These regions appear



(a)



(b)

Figure 4.14: Signal strength contour maps: (a) for the RF link, and (b) for the OW link. The x-y plane is the azimuth plane, while the x-z is the elevation plane. For the case of the azimuth plane, we assume the receiver maintains a fixed height of 5 m.

Table 4.3: Parameters for the RF and OW links

RF link	$P_{\text{Tx}} = -12.22$ dB (60 mW), $f = 2.4$ GHz, $\kappa = 2.5$, $d_o = 1$ m, $G_{\text{Tx}} = G_{\text{Rx}} = 1$
OW link	$P = -15.23$ dB (30 mW), $A = 1\text{cm}^2$, $n = 1.5$, $\Phi_{1/2} = \Psi_{\mathcal{C}} = 30°$

because the concentrator gain $g(\psi)$ becomes zero, according to (4.3), if the incidence angle ψ is greater than the half-angle FOV $\Psi_{\mathcal{C}}$. Therefore, the receiver does not

detect any optical signal when it is within these areas. In contrast, it can detect the RF-based communication signal inside these white regions as seen in Figure 4.14(a).

4.3.2 Control Stategy

First, we introduce the dynamic model of the aerial vehicle. Then, we present a controller for the quadrotor such that it follows a desired direction which should point towards the sensor node. We detail in Section 4.3.3 how this direction is estimated based on the RSS measured by the four antennas.

Quadrotor Model

Let $\{\mathcal{W}\}$ be an inertial frame such that its unit vectors along the axes are given by $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$, with

$$\mathbf{u_1} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T, \mathbf{u_2} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T, \mathbf{u_3} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T.$$

Let $\{\mathcal{B}\}$ represent a fixed-body frame attached to the center of mass of the aerial vehicle. Both frames $\{\mathcal{W}\}$ and $\{\mathcal{B}\}$ are illustrated in Figure 4.13(a). The position vector of $\{\mathcal{B}\}$ with respect to $\{\mathcal{W}\}$ is denoted by $\mathbf{r} = \begin{bmatrix} x_q & y_q & z_q \end{bmatrix}^T$, while its orientation is expressed by $\boldsymbol{\vartheta} = \begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}^T$. The terms α, β, and γ are the roll, pitch, and yaw Euler angles, respectively. We assume that the orientation of the rigid body with respect to the inertial frame is given by the z-x-y rotation matrix

$$\mathbf{R} = \begin{pmatrix} c\gamma c\beta - s\alpha s\gamma s\beta & -c\alpha s\gamma & c\gamma s\beta + c\beta s\alpha s\gamma \\ c\beta s\gamma + c\gamma s\alpha s\beta & c\alpha c\gamma & s\gamma s\beta - c\gamma c\beta s\alpha \\ -c\alpha s\beta & s\alpha & c\alpha c\beta \end{pmatrix},$$

where c and s are shorthand forms for cosine and sine, respectively. The full nonlinear dynamics of the quadrotor can be expressed as (see Section 2.1.1)

$$\begin{aligned} m_q\ddot{\mathbf{r}} &= -m_q g\mathbf{u}_3 + F\mathbf{R}\mathbf{u}_3, \\ \dot{\mathbf{R}} &= \mathbf{R}\hat{\boldsymbol{\Omega}}, \\ \mathbf{J}\dot{\boldsymbol{\Omega}} &= -\boldsymbol{\Omega}\times\mathbf{J}\boldsymbol{\Omega} + \mathbf{M}. \end{aligned} \tag{4.38}$$

Here, m_q is the mass of the aerial vehicle, g is the gravitational constant, $\boldsymbol{\Omega}$ is the angular velocity of the vehicle expressed in the fixed-body frame, and $\mathbf{J}$ is a constant inertia matrix. F and $\mathbf{M}$ are the total thrust and the torque control inputs applied to the quadrotor. The *hat map* $\hat{\cdot}:\mathbb{R}^3\to \mathrm{SO}(3)$ is defined by the condition that $\hat{\mathbf{a}}\mathbf{b} = \mathbf{a}\times\mathbf{b}$ for all $\mathbf{a},\mathbf{b}\in\mathbb{R}^3$.

Quadrotor Control

A controller that guarantees stability for small deviations from the hover position is presented in [67]. We adapt this controller to guide the quadrotor to follow a desired direction vector $\mathbf{v}$ expressed with respect to $\{\mathcal{W}\}$. Let $\mathbf{u}_\mathbf{v}$ be the unitary vector of $\mathbf{v}$, then we define the following vector

$$\mathbf{a} = k_a\mathbf{u}_\mathbf{v} - \dot{\mathbf{r}}, \tag{4.39}$$

where k_a is a scalar gain. Using $\mathbf{a} = \begin{bmatrix} a_x & a_y & a_z \end{bmatrix}^T$ defined by (4.39), we can find the appropriate roll $\alpha^\star$ and pitch $\beta^\star$ angles according to

$$\begin{aligned} \alpha^\star &= \frac{1}{g}\left(a_x\sin\gamma^\star - a_y\cos\gamma^\star\right), \text{ and} \\ \beta^\star &= \frac{1}{g}\left(a_x\cos\gamma^\star + a_y\sin\gamma^\star\right). \end{aligned} \tag{4.40}$$

We assume that the quadrotor has to keep its initial yaw angle, so $\gamma^\star = \gamma_o$. Then, the control law for the torque input is

$$\mathbf{M} = -\mathbf{K}_\mathrm{R}\mathbf{e}_\mathrm{R} - \mathbf{K}_\Omega\boldsymbol{\Omega}, \tag{4.41}$$

where $\mathbf{e}_{\mathrm{R}} = \begin{bmatrix} \alpha - \alpha^{\star} & \beta - \beta^{\star} & \gamma - \gamma^{\star} \end{bmatrix}^{T}$, and $\mathbf{K}_{\mathrm{R}}$, $\mathbf{K}_{\Omega}$ are diagonal gain matrices. In addition, the control law for the total thrust is

$$F = -m_q \left(g + k_v a_z \right), \tag{4.42}$$

where k_v is a control gain.

4.3.3 Direction Estimate

For each antenna, we make N RSS measurements per T seconds as the quadrotor moves. Let $P_{\mathrm{RF}_i}(t)$ be the RSS measurement at time t for the i^{th} antenna given according to (4.35). Then, we maintain an average RSS value for each antenna which we denote as Q_i. Using an exponentially weighted average, we update Q_i according to

$$Q_i \leftarrow \eta Q_i + (1 - \eta) P_{\mathrm{RF}_i}(t), \tag{4.43}$$

where $i = 1, \ldots, 4$ indexes the four antennas and $\eta \in (0, 1]$ is a forgetting factor.

Every T seconds, we proceed to find the direction estimate vector $\boldsymbol{\mu}$ in the following manner:

1) Using the average RSS for the four antennas, we form the vector of average RSS measurements as

$$\mathbf{Q}_{\mathrm{RF}} = \begin{bmatrix} Q_1 & Q_2 & Q_3 & Q_4 \end{bmatrix}^{T}. \tag{4.44}$$

2) From $\mathbf{Q}_{\mathrm{RF}}$, we form its min-max normalization which is given by

$$\bar{\mathbf{Q}}_{\mathrm{RF}} = \frac{1}{\Upsilon - \upsilon} (\mathbf{Q}_{\mathrm{RF}} - \upsilon \mathbf{1}), \tag{4.45}$$

where $\Upsilon = \max(\mathbf{Q}_{\mathrm{RF}})$, $\upsilon = \min(\mathbf{Q}_{\mathrm{RF}})$, and $\mathbf{1}$ is a 4×1 vector of ones.

3) We compare the average RSS of antennas 1 and 3 to find the largest between them. Notice that antennas 1 and 3 (and similarly 2 and 4) have maximum spacing with respect to the center of mass of the quadrotor, see Figure 3.4(a).

4) Let ${}^{\mathcal{B}}\boldsymbol{\ell}_{13}$ be the location of the antenna selected in Step 3 with respect to the fixed-body frame $\{\mathcal{B}\}$. Also, let $\bar{Q}_{13}$ be the normalized average RSS associated with it.

5) We simultaneously repeat Steps 3 and 4 for antennas 2 and 4. As result, we obtain ${}^{\mathcal{B}}\boldsymbol{\ell}_{24}$ and $\bar{Q}_{24}$.

6) We determine the direction estimate vector as

$$\boldsymbol{\mu} = \mathbf{R}\left(\bar{Q}_{13}{}^{\mathcal{B}}\boldsymbol{\ell}_{13} + \bar{Q}_{24}{}^{\mathcal{B}}\boldsymbol{\ell}_{24}\right). \tag{4.46}$$

Notice that this vector is defined with respect to $\{\mathcal{W}\}$.

We also maintain an exponentially weighted average for the OW signal strength. Let $P_{\mathrm{OW}}(t)$ be the optical signal strength at time t found by (4.37), then we obtain its average O by

$$O \leftarrow \eta O + (1-\eta)P_{\mathrm{OW}}(t). \tag{4.47}$$

If the average optical signal strength is lower than a predefined threshold τ_{OW}, the desired direction $\mathbf{v}$ for the control strategy (Section 4.3.2) is the vector μ given by (4.46). Otherwise, $\mathbf{v}$ is the zero vector to command the quadrotor to hover. Next, we present numerical simulations of the proposed approach.

Table 4.4: Simulation parameters

RF Link	$P_{\mathrm{Tx}} = 60$ mW, $f = 2.4$ GHz, $d_o = 1$ m, $\kappa = 2.5$
Quadrotor Model	$m_q = 0.531$ kg, $\mathbf{J} = \mathrm{diag}(3.6, 3.6, 7.1)10^{-3}$ kg·m^2, dimensions $= 54 \times 54 \times 5.5$ cm
Controller	$k_a = 0.85$, $k_v = 0.01$, $\mathbf{K}_{\mathrm{R}} = 0.25\,\mathbf{I}_{3\times3}$, $\mathbf{K}_{\Omega} = 0.17\,\mathbf{I}_{3\times3}$

4.3.4 Simulation Results

We report two sets of simulations to demonstrate the validity of the proposed approach. The time step is 0.005 s and the forgetting factor η is 0.9. The rest of the

simulation parameters are summarized in Table 4.4. In this table, $\mathbf{I}_{3\times3}$ denotes a 3 by 3 identity matrix and the values for the quadrotor model corresponds to the AscTec Hummingbird quadrotor [77].

In the first simulation set, we begin with the aerial vehicle at the position $\begin{bmatrix}-5 & 5 & 5\end{bmatrix}^T$ m, and the sensor node at $\begin{bmatrix}3 & -2 & 0\end{bmatrix}^T$ m. The quadrotor gathers 100 samples of the signal strength for the RF and OW link every second as the quadrotor moves, so $N = 100$ and $T = 1$ s. We update the average signal strength for each antenna Q_i by (4.43) and the optical signal strength O according to (4.47). Every second, we find the direction vector $\boldsymbol{\mu}$ following the steps detailed in Section 4.3.3. As indicated at the end of this section, if $O \leq \tau_{\text{OW}}$ (for our case $\tau_{\text{OW}} = -25$ dBm) then the desired direction for the control strategy is $\mathbf{v} = \boldsymbol{\mu}$. Otherwise, $\mathbf{v}$ is the zero vector commanding the quadrotor to hover. Once the desired direction vector $\mathbf{v}$ is chosen, we find the quadrotor control inputs by applying (4.39) to (4.42). Then, we update the state of the aerial robot according to the model in (4.38).

The vector $\boldsymbol{\mu}$ estimated according to the procedure detailed in Section 4.3.3 is shown at different time instants in Figure 4.15(a). In this figure, we also illustrate the contour isolines for the RF link in the azimuth plane. The RF contour lines are the average signal strength without random fading, and are included as a background reference. In this simulation, the source direction vector $\mathbf{v}$ generally points towards the RF source enabling the quadrotor to approach the sensor node. We depict in Figure 4.15(b) the path described by the quadrotor as well as its final position. For these figures, we also plot the contour isolines for the OW link in the azimuth plane. As in the RF contours, the optical contour lines are plotted without considering noise and are included only for visual reference. The quadrotor converges to a position above the sensor node such that the optical link is above the predefined power threshold enabling optical communications, see Figure 4.16(b).

We plot the signal strength received by the four antennas in Figure 4.16(a). Roughly speaking, the direction estimate vector $\boldsymbol{\mu}$ is generally determined based on

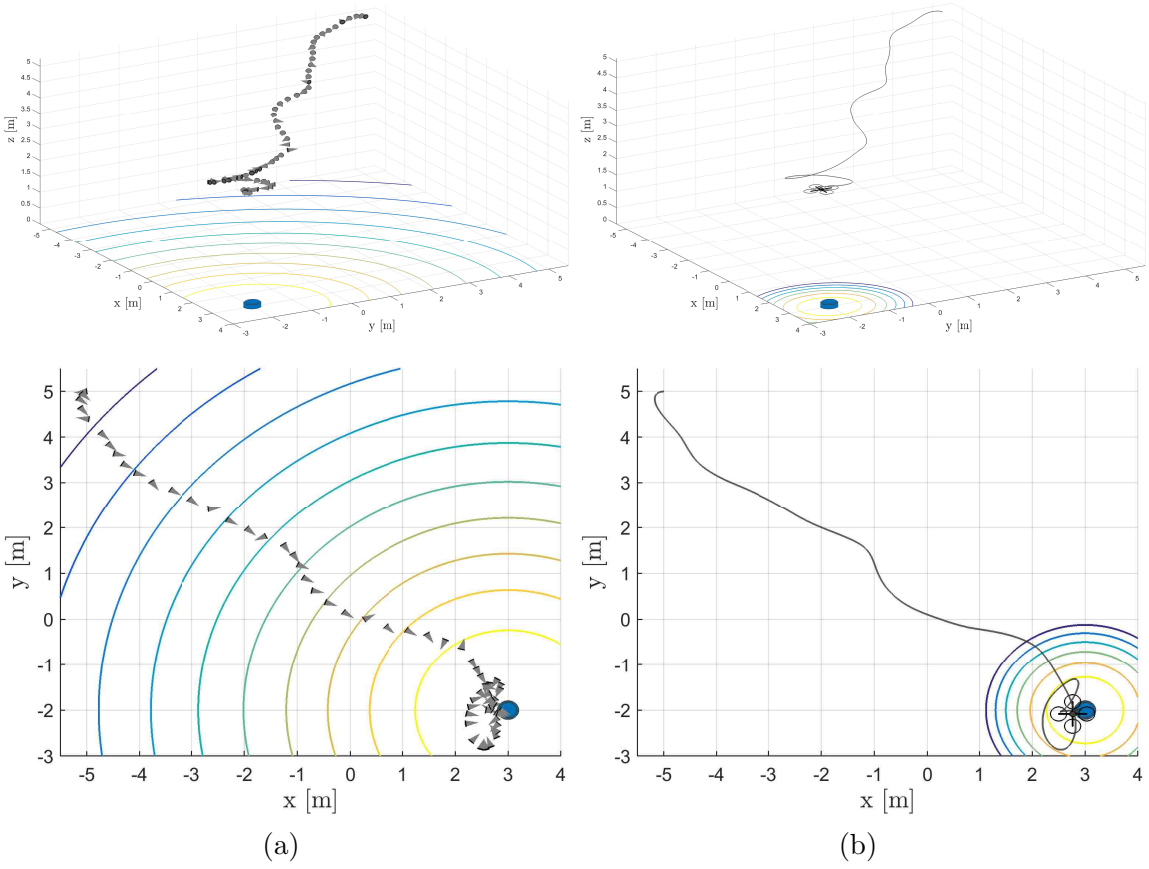


Figure 4.15: (a) Direction estimates at different time instants. (b) Trajectory described by the quadrotor. (Top) 3D-view, and (Bottom) 2D-view. The contour lines shown in these figures are averages drawn without considering noise or fading fluctuation and they are included only for visual reference.

antennas 1 and 4 for the first 15 s. This occurs because of the initial orientation of the quadrotor with respect to the source bearing. After initial approach, the control maintains a good relative position within optical range, see Figure 4.15(b). As seen in Figure 4.16(a), the RF signal strength has some significant drops. This occurs when the quadrotor is generally above the RF source, such that the source bearing no longer aligns with the main lobe of the antenna. We also plot the signal strength for the optical link, see Figure 4.16(b). The optical signal is detected approximately 13 s after starting the simulation. Then, the optical signal strength grows surpassing

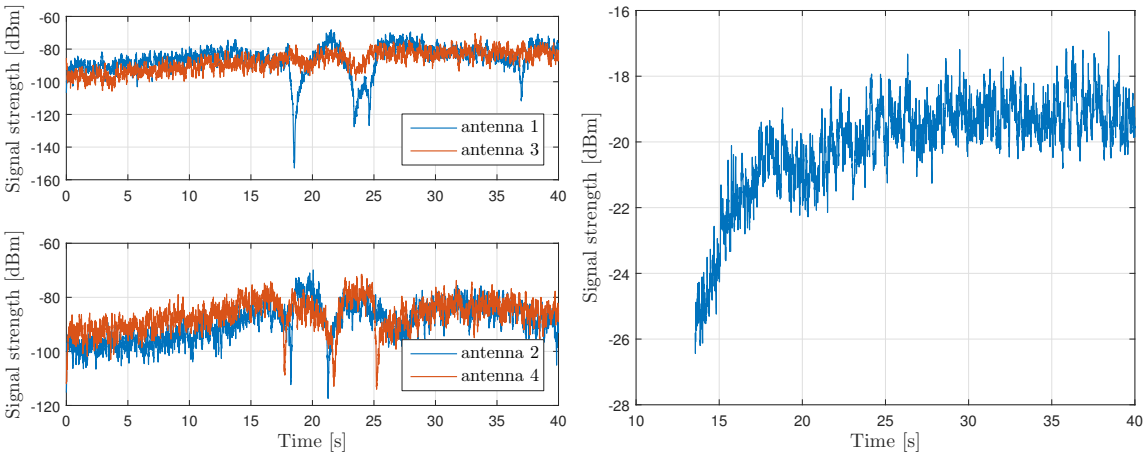


Figure 4.16: (a) Received signal strength for the four antennas: (Top) antennas 1 and 3, and (Bottom) antennas 2 and 4. (b) Signal strength for the optical link.

the prescribed threshold τ_{OW}. After convergence, the average optical signal strength is approximately -20.1 dBm.

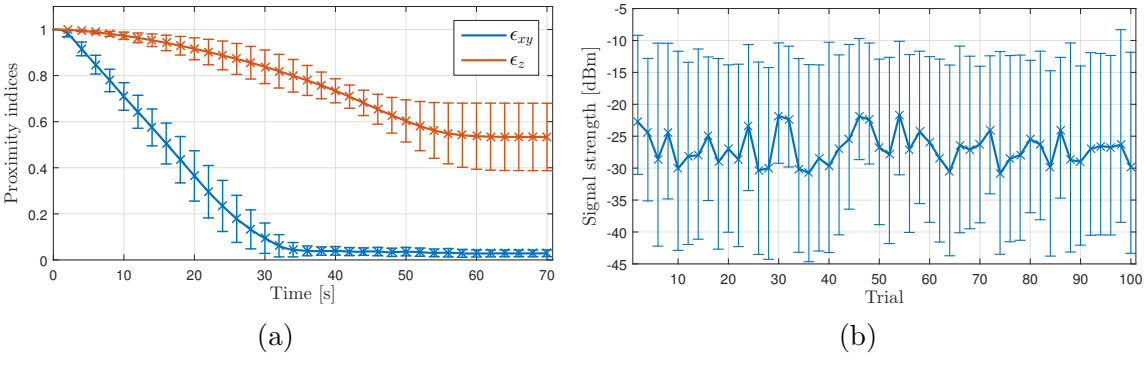


Figure 4.17: (a) Proximity indices. The cross marker "×" represents the average value, while the bars denote the standard deviation. (b) Optical signal strength for 50 of the 100 trials. The cross marker "×" represents the average value, while the bars connect the minimum and maximum values for each trial.

For the second set of simulations, we randomly generate 100 initial positions for

the quadrotor and the sensor. We assume a workspace area of $40 \times 40 \times 20$ m with the quadrotor initial height between 5 and 20 m. We set the initial quadrotor-sensor relative distance to 25 m, and test our approach over the 100 trials. In order to check convergence of quadrotor position relative to the sensor location, we compute the x-y and z proximity indices defined as

$$\epsilon_{xy} = \frac{\Delta_{xy}}{\Delta_{xy_0}} \text{ and } \epsilon_z = \frac{\Delta_z}{\Delta_{z_0}}, \tag{4.48}$$

respectively. Here, $\Delta_{xy} = \sqrt{(x_q - x_s)^2 + (y_q - y_s)^2}$ and $\Delta_z = |z_q - z_s|$. The initial value (at time $t = 0$) of Δ_{xy} is Δ_{xy_0} and similarly for Δ_{z_0}. The values x_s, y_s, and z_s are the x, y, and z position coordinates of the sensor node, respectively. Notice that as both proximity indices decrease, the quadrotor moves toward the sensor location. The evolution of these indices are shown in Figure 4.17(a). In this figure, the cross marker "$\times$" represents the average value, while the bars denote the standard deviation. Applying the proposed approach, the quadrotor comes close to the sensor for all trials such that the quadrotor-sensor distance is reduced to on average $< 10\%$ of the initial Δ_{xy}, *i.e.*, $\epsilon_{xy} < 0.1$, and $< 60\%$ of the initial Δ_z, *i.e.*, $\epsilon_z < 0.6$. In Figure 4.17(b), we show the optical signal strength for 50 of the 100 trials. For this figure, The cross marker "$\times$" represents the average value, while the bars connect the minimum and maximum values for each trial. The average optical signal strength is in the range of -22 to -32 dBm.

4.4 Conclusions

In this chapter, we studied an optical wireless link between aerial and ground robotic vehicles. The combination of optical and RF has the potential to dramatically expand communications rates. Using a model of a directed LOS optical link, we defined a connectivity cone on top of the mobile ground optical receiver based on the requirement of keeping the bit rate over a minimum desired value, for a fixed maximum bit error rate. Thus, the aerial transmitter has to approach and remain within this cone

in order to establish the optical link. A stochastic non-linear model for the measurement of the location of the ground receiver with respect to the flying transmitter was assumed. Our method employs particle filters to estimate the receiver state and therefore the position of the cone. An algorithm to find the closest point at the cone with respect to the transmitter was created. Using this reference point, a controller for the aerial transmitter to remain inside of the connectivity cone was designed. The simulation results show that the proposed approach is able to maintain an adequate relative position between receiver and transmitter in order to create a point-to-point optical wireless link. Based on these results, we envision an scenario where an autonomous aerial platform equipped with hybrid radio-optical communications can be employed to gather sensor node data. This bi-modal communication scheme can eliminate the necessity of precise relative localization between the sensors and the flying robot. We proposed and demonstrated through numerical simulations a control strategy for the flying robot in order that it moves towards the sensor node using RF measurements. The key idea is to find an estimate of the source direction using the RSS measured by four directional antennas installed on the aerial vehicle. Once the optical signal strength is above a predefined threshold, the quadrotor hovers close to the sensor node in order that the optical link can be maintained at a desired signal strength.

Some avenues for further research include tracking of the optical SNR and adjusting the data rate accordingly. Our simulations reveal a large swing in the potential bit rate, so tracking the optical SNR would help to increase the average communication rate. Relaxing the desired communications rate results in a very stable link, that may be useful for collaborative control tasks. For the case of the hybrid RF/OW communication scheme, it will be beneficial to couple the RSS-assisted direction estimation with other sensors, *e.g.*, a camera, to improve the sensor localization for accurate pointing of the optical beam. In addition, the experimental validation of our approach for micro-sized robotic platforms working in indoor environments is an important topic for future work.

Chapter 5

Building Coalitions of Heterogeneous Agents

Teams of agents with different skills can solve critical missions by efficiently joining their complementary abilities. One critical step to exploit the distinct resources available on a set of agents is to form a coalition, *i.e.*, an alliance that satisfies the requirements imposed by a mission. In this work, we represent the relation between agent capabilities and required resources for executing a given task by a weighted bipartite graph. Using this graph, we find an assignment between agents and resource capabilities such that the total weight of capabilities is maximized. From this assignment, also known as matching in Graph Theory, we compute a coalition of agents whose total resource capabilities can satisfy the task resource requirements. Finally, we measure the heterogeneity of the computed coalition and analyze how it is affected by the task constraints and the amount of resources present in the agents. This chapter is based on our work published in [124].

5.1 Preliminaries

We provide some definitions about bipartite graphs and its application in solving assignment problems [125–127]. Also, we introduce the metric of complexity and disparity for measuring the heterogeneity of a multi-agent system [128].

5.1.1 Bipartite Graphs and Assignment Problems

Definition 5.1 (Bipartite Graph)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a weighted graph where $\mathcal{V}$ is the set of vertices or nodes, $\mathcal{E}$ is the set of edges, and $\mathcal{W}$ is the set of weights associated with each edge. Then, $\mathcal{G}$ is a bipartite graph if the following conditions are satisfied:*

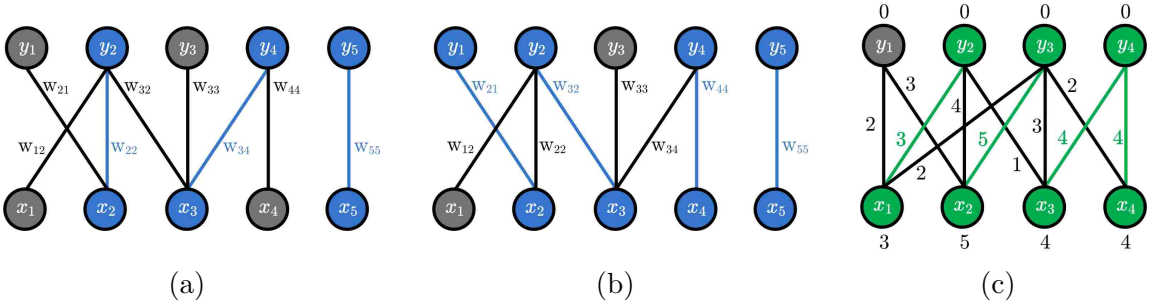
1. *$\mathcal{V}$ can be expressed as the union of two sets $\mathcal{V} = \mathcal{X} \cup \mathcal{Y}$ such that $\mathcal{X} \cap \mathcal{Y} = \emptyset$, and*
2. *$\mathcal{G}[\mathcal{X}]$ and $\mathcal{G}[\mathcal{Y}]$ are null graphs.*

In this definition, $\mathcal{G}[\mathcal{X}]$ is the subgraph of $\mathcal{G}$ induced by $\mathcal{X}$ [126], *i.e.*, $\mathcal{G}[\mathcal{X}]$ has as vertex set $\mathcal{X}$, the edge set is given by all the edges of $\mathcal{G}$ having end vertices in $\mathcal{X}$, and the weight set contains the weights associated with these edges. Examples of weighted bipartite graphs are shown in Figure 5.1. For instance, $\mathcal{G}[\mathcal{X}] = (\{x_1, x_2, x_3, x_4\}, \emptyset, \emptyset)$ for the bipartite graph in Figure 5.1(a). In particular, Figure 5.1(c) is an example of a weighted bipartite graph with labels. Next, we provide the definitions of labeling and neighbors.

Definition 5.2 (Labeling)**.** *A labeling of a graph is a real-valued label $\ell(v)$ assigned to every $v \in \mathcal{V}$. For a weighted bipartite graph, a labeling is feasible if and only if*

$$\ell(x) + \ell(y) \geq \mathrm{w}_{xy} \quad \forall x \in \mathcal{X}, \forall y \in \mathcal{Y}, \tag{5.1}$$

where w_{xy} denotes the weight of the edge connecting x and y.



(a) (b) (c)

Figure 5.1: Bipartite graph examples. (a) The blue edges are forming a matching $\mathcal{M}$ which is not perfect for this case. Nodes x_1, x_4, y_1 and y_3 are free. $p = \{y_1, \mathrm{w}_{21}, x_2, \mathrm{w}_{22}, y_2, \mathrm{w}_{32}, x_3, \mathrm{w}_{34}, y_4, \mathrm{w}_{44}, x_4\}$ is an alternating path. Indeed, this path is an augmenting path. (b) Using p the size of the matching is increased by one. (c) An example of a feasible labeling for a bipartite graph. The tight edges are shown in green.

Definition 5.3 (Neighbors)**.** *The edge between x and y for which (5.1) is an equality is known as tight. If x is linked to y by a tight edge then x is called a neighbor of y. All the neighbors of x are denoted by $\mathcal{N}_{\mathcal{E}}(x)$ and the set $\mathcal{N}_{\mathcal{E}}(\mathcal{S})$ defined as*

$$\mathcal{N}_{\mathcal{E}}(\mathcal{S}) = \bigcup_{s \in \mathcal{S}} \mathcal{N}_{\mathcal{E}}(s), \tag{5.2}$$

includes all the neighbors of any node s that belongs to $\mathcal{S}$.

A feasible labeling example for a weighted bipartite graph is given in Figure 5.1(c) where tight edges are shown in green. For this example,

$$\mathcal{N}_{\mathcal{E}}(\mathcal{S}) = \{x_2, x_3, x_4\} \text{ for } \mathcal{S} = \{y_3, y_4\}.$$

Bipartite graphs have extensively been used for describing *assignment problems* [125,127]. These problems deal with the question of how to assign n items, *e.g.*, tasks, to m other items, *e.g.*, machines. When a bipartite graph is employed for modeling an assignment problem, the vertices in $\mathcal{X}$ represent the tasks and the vertices in $\mathcal{Y}$ represent the machines which are linked to the tasks by means of edges between $\mathcal{X}$

and $\mathcal{Y}$. The weight associated with an edge can represent, for instance, the reward of using a machine to carry out the task that the edge is connecting.

Definition 5.4 (Matching and Perfect Matching)**.** *A* matching $\mathcal{M}$ *in a graph (not just for a bipartite graph) is a subset of* $\mathcal{E}$ *where no two edges share a node. A matching* $\mathcal{M}$ *is* perfect *if every vertex in* $\mathcal{V}$ *is a vertex of some edge in* $\mathcal{M}$*.*

Definition 5.5 (Free Vertex and Matching Edge)**.** *If a vertex is not linked to any edge in a matching* $\mathcal{M}$ *then it is called* free*. An edge of* $\mathcal{E}$ *is called a* matching edge *if it belongs to* $\mathcal{M}$*. Otherwise, it is a* non-matching edge*.*

In Figure 5.1(a), the blue edges form a non-perfect matching $\mathcal{M}$ of the weighted bipartite graph and vertices y_1 and x_4 are free.

When a weighted bipartite graph describes an assignment problem, the general goal is to find a matching such that the total weight of all the assignments is maximized if the weights represent for example, rewards. Hence, the famous *Hungarian Algorithm* [125, 126], also known as the *Kuhn-Munkres algorithm* or just *Kuhn algorithm*, is the preferred method to compute a matching with maximum weight from a weighted bipartite graphs associated to an assignment problem. We will use this algorithm in Section 5.3 to find a coalition based on the computed matching. The Hungarian Algorithm employs *alternating* and *augmenting paths* to add edges to the matching. Next, we present the definitions of these two fundamental concepts.

Definition 5.6 (Alternating and Augmenting Paths)**.** *An alternating path is formed by edges that alternatively are matching and non-matching edges. When an alternating path has the property that its both endpoints are free, it is called an augmenting path.*

In Figure 5.1(a), $p = \{y_1, \mathrm{w}_{21}, x_2, \mathrm{w}_{22}, y_2, \mathrm{w}_{32}, x_3, \mathrm{w}_{34}, y_4, \mathrm{w}_{44}, x_4\}$ forms an alternating path because the edges alternate between being part of the matching $\mathcal{M}$ (blue) and not being part of $\mathcal{M}$ (black). Furthermore, this path is an augmenting

path since y_1 and x_4 are free. The path is called *augmenting* since if we "toggle" the blue and black edges, we increase the size of $\mathcal{M}$ by one. Figure 5.1(b) shows the augmented matching produced by the path p. The augmentation procedure is described in the following definition.

Definition 5.7 (Augmentation)**.** *Let p be an augmenting path with respect to a matching $\mathcal{M}$, then the matching augmented by p is obtained by the following:*

1. *all previously non-matching edges of p now become matching and all previously matching edges of p become non-matching, and*
2. *all matching edges of $\mathcal{M}$ which are not in p remain as matching edges.*

The basic principle of the Hungarian Algorithm is to iteratively either increase the size of the current matching estimate $\mathcal{M}$ or improve the labeling of the bipartite graph such that new tight edges appear. In this fashion, the $\mathcal{M}$ can be enlarged in later iterations. The previous definitions will allow us to summarize in Section 5.3 how the Hungarian Algorithm works and how we adapt it for our case.

5.1.2 Heterogeneity of Multi-agent Systems

Recently, a measure of heterogeneity for multi-agent systems has been proposed in the literature [128]. In this work, we present this metric since we are interested in computing the heterogeneity of a multi-agent coalition formed to execute a given task. For complete details about the following definitions, please refer to [128].

Let $\mathcal{A}$ be a set of N distinct agents such that all of them have different resources, for instance different sensing and end-effector capabilities for the case of robots. For $i = 1, \ldots, N$, let $p_i \in [0 \;\; 1]$ be the probability of randomly chosen agent $a_i \in \mathcal{A}$ such that

$$\sum_{i=1}^{N} p_i = 1.$$

Let $\mathcal{C}$ be a subset of $\mathcal{A}$, *i.e.*, $\mathcal{C} \subseteq \mathcal{A}$, whose cardinality is denoted by $|\mathcal{C}|$. It is clear that the agents in $\mathcal{C}$ form a multi-agent system. The variety of a multi-agent system describes how well dispersed are its agents with respect to all the available agents. This measure of disorder can be captured by the *entropy* of the multi-agent system [128, 129].

Definition 5.8 (Entropy)**.** *The entropy of the system formed by the agents in $\mathcal{C}$ is given by*

$$E(\mathcal{C}) = -\sum_{i,a_i \in \mathcal{C}} p_i \log(p_i). \tag{5.3}$$

Traditionally, entropy is denoted with H. However, we will later use this symbol for heterogeneity.

Meanwhile, the disparity of a multi-agent system measures how diverse the agents are from one to another and it can be computed using the *Rao's quadratic entropy* [128, 130].

Definition 5.9 (Rao's quadratic entropy)**.** *The Rao's quadratic entropy of the system formed by the agents in $\mathcal{C}$ is given by*

$$Q(\mathcal{C}) = \sum_{i,a_i \in \mathcal{C}} \sum_{j,a_j \in \mathcal{C}} p_i p_j d_{ij}^2, \tag{5.4}$$

where d_{ij} is a metric of the difference between $a_i, a_j \in \mathcal{C}$.

Using the definitions of entropy and the Rao's quadratic entropy given before, the heterogeneity H of the multi-agent system formed by the agents in $\mathcal{C}$ can be formulated.

Definition 5.10 (Heterogeneity)**.** *The heterogeneity of the system formed by the agents in $\mathcal{C}$ is given by*

$$H(\mathcal{C}) = E(\mathcal{C})Q(\mathcal{C}), \tag{5.5}$$

where $E(\mathcal{C})$ and $Q(\mathcal{C})$ are given by (5.3) and (5.4), respectively.

As an example of the concept of heterogeneity for multi-agent systems, consider that we have available the six species of agents illustrated in Figure 5.2(a). Within these species, we have two types of agents: aerial robotic platforms and ground vehicles. There are three different sizes of robots for each type that create the six available species. Assume we select a group as the one shown in Figure 5.2(b). Clearly, this group forms an homogeneous multi-agent system, so we denoted it as *Group Ho*. Now suppose we select two more groups, the ones in Figures 5.2(c) and 5.2(d) denoted as *Group He1* and *Group He2*, respectively. It is clear that both groups, *Group He1* and *Group He2*, are more heterogeneous than *Group Ho*. For *Group He1*, it is more heterogeneous than *Group Ho* since its members are distributed among three of the six available species. Thus, *Group He1* shows variety. On the other hand, the members of *Group He2* are distributed only among two of the six available species. Therefore, *Group He2* has more variety than *Group Ho* but less than *Group He1*. However, the agents in *Group He1* are only of one type: aerial robots, while the agents in *Group He2* are from the two available types: aerial and ground robots. Consequently, *Group He2* shows more disparity than *Group He1*.

From the previous discussion, *Group He1* and *Group He2* are more heterogeneous than *Group Ho*, but in different ways. *Group Ho* has no variety or disparity. *Group He1* is more heterogeneous than *Group Ho* since it has a higher variety. In contrast, *Group He2* is more heterogeneous than *Group Ho* because it exhibits a higher disparity. Therefore, in general, a heterogeneous multi-agent system has to be present these two characteristics: variety and disparity, as it can be seen in *Group He3* illustrated in Figure 5.2(e).

After giving the background related to this chapter, we introduce the coalition formation problem.

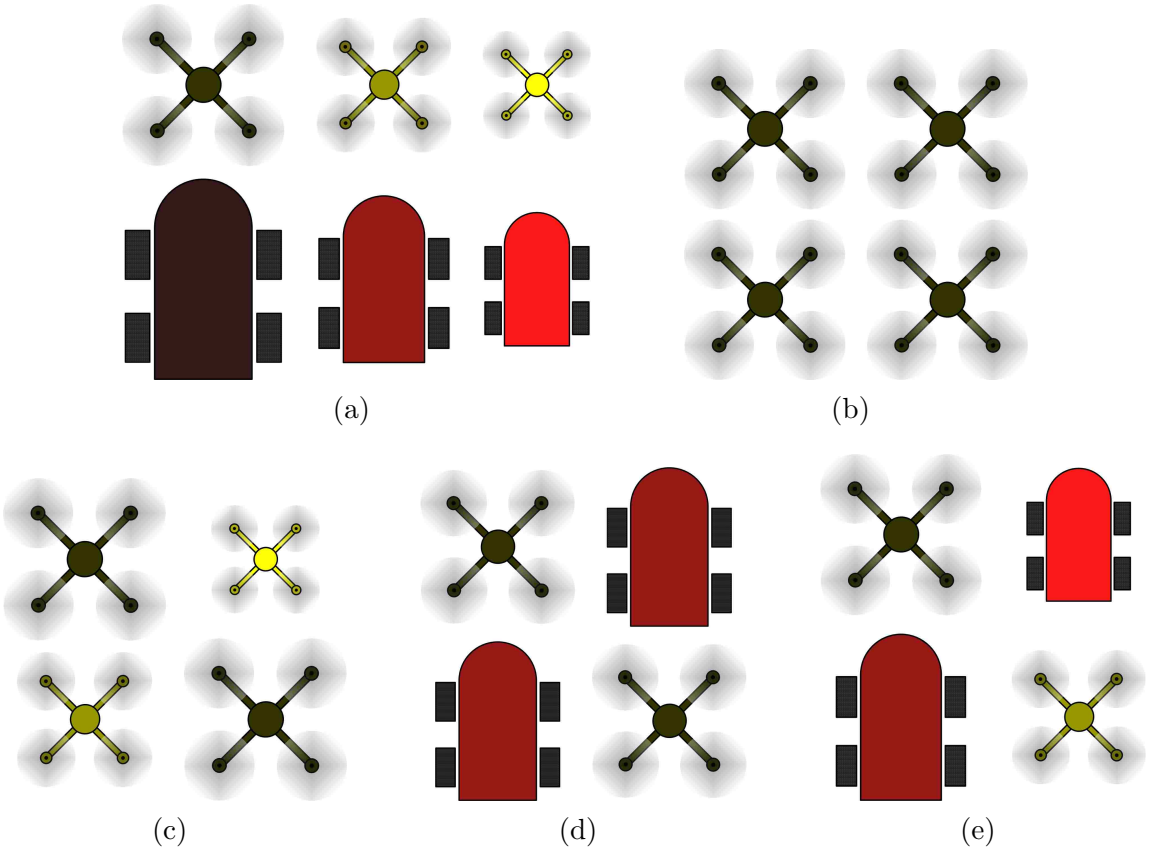


Figure 5.2: Heterogeneity example. (a) Available species. (b) *Group Ho* Homogeneous group, no variety or disparity. (c) *Group He1*, Heterogeneous group showing variety but little disparity. (d) *Group He1*, Heterogeneous group showing low variety but high disparity. (e) *Group He3* Heterogeneous group showing both variety and disparity.

5.2 The Coalition Formation Problem

We adopt a formulation for the coalition formation problem similar to the one described in [44, 48]. However, we state the problem more formally and add conditions to force that a coalition which can execute a given task must be formed by more than one agent.

Let $\mathcal{R} = \{r_1, r_2, \ldots, r_M\}$ be the set of M resources needed to execute a given

task τ. In addition, the task τ has an M-dimensional vector $\mathbf{w}^\tau$ representing the resource requirements and it is defined by

$$\mathbf{w}^\tau = \begin{bmatrix} \mathrm{w}^\tau_{r_1} & \mathrm{w}^\tau_{r_2} & \dots & \mathrm{w}^\tau_{r_M} \end{bmatrix}, \tag{5.6}$$

where $\mathrm{w}^\tau_{r_j}$ for all $j \in I_{\mathcal{R}}$ quantifies the amount of resource j required to accomplish the given task τ. Here, $I_{\mathcal{R}} = \{1, \dots, M\}$ is the index set of $\mathcal{R}$. We also assume that $\mathrm{w}^\tau_{r_j} > 0 \ \forall j \in I_{\mathcal{R}}$.

Now, let $\mathcal{A} = \{a_1, a_2, \dots, a_N\}$ be the set of N agents. In general, $N \neq M$. Each agent a_i for $i \in I_{\mathcal{A}} = \{1, \dots, N\}$ has its own capability resource vector given by

$$\mathbf{w}^{a_i} = \begin{bmatrix} \mathrm{w}^{a_i}_{r_1} & \mathrm{w}^{a_i}_{r_2} & \dots & \mathrm{w}^{a_i}_{r_M} \end{bmatrix}, \tag{5.7}$$

where $\mathrm{w}^{a_i}_{r_j}$ for all $j \in I_{\mathcal{R}}$ quantifies the amount of resource j available at agent a_i. We can arrange all the capability resource vectors as the $N \times N$ matrix defined by

$$\mathbf{W}^{\mathcal{A}} = \begin{bmatrix} \mathbf{w}^{a_1} \\ \vdots \\ \mathbf{w}^{a_N} \end{bmatrix}. \tag{5.8}$$

We call $\mathbf{W}^{\mathcal{A}}$ as the capability resource matrix. Also, we assume the following two conditions,

$$\forall i \in I_{\mathcal{A}} \text{ and } \forall j \in I_{\mathcal{R}}, 0 \leq \mathrm{w}^{a_i}_{r_j} \leq \mathrm{w}^\tau_{r_j}, \text{and} \tag{5.9}$$

$$\forall i \in I_{\mathcal{A}} \ \exists j \in I_{\mathcal{R}} \text{ such that } 0 \leq \mathrm{w}^{a_i}_{r_j} < \mathrm{w}^\tau_{r_j}. \tag{5.10}$$

Condition (5.9) indicates that any resource capability for any robot is non-negative and its maximum value is the value of the resource required for executing the given task. Meanwhile, condition (5.10) guarantees that no agent in $\mathcal{A}$ can perform the given task alone. Therefore, an alliance or coalition of agents is needed in order to execute the task.

Definition 5.11 (Coalition)**.** *Let $\mathcal{C}$ be a subset of $\mathcal{A}$, then we say that $\mathcal{C}$ is a* coalition *of agents. Also, we define the resource capability vector of the coalition $\mathcal{C}$ as*

$$\mathbf{w}^{\mathcal{C}} = \begin{bmatrix} \mathrm{w}^{\mathcal{C}}_{r_1} & \mathrm{w}^{\mathcal{C}}_{r_2} & \dots & \mathrm{w}^{\mathcal{C}}_{r_M} \end{bmatrix}, \tag{5.11}$$

where

$$\forall j \in I_{\mathcal{R}},\ \mathrm{w}_{r_j}^{\mathcal{C}} = \sum_{i \in I_{\mathcal{C}}} \mathrm{w}_{r_j}^{a_i}, \tag{5.12}$$

with $I_{\mathcal{C}} = \{i \in I_{\mathcal{A}} \mid a_i \in \mathcal{C}\}$ *being the index set of* $\mathcal{C}$.

Remark 5.1. *From this definition, the set of all the agents is a coalition, i.e.,* $\mathcal{C} = \mathcal{A}$. *In this case,* $\mathbf{w}^{\mathcal{C}} = \mathbf{w}^{\mathcal{A}}$ *which means that* $\forall j \in I_{\mathcal{R}}, \mathrm{w}_{r_j}^{\mathcal{C}} = \mathrm{w}_{r_j}^{\mathcal{A}}$. *On the other hand, if a coalition is a proper subset of the set of all the agents, i.e.,* $\mathcal{C} \subsetneq \mathcal{A}$, *then it is clear from (5.12) that* $\forall j \in I_{\mathcal{R}}, \mathrm{w}_{r_j}^{\mathcal{C}} \leq \mathrm{w}_{r_j}^{\mathcal{A}}$. *Consequently for any coalition* $\mathcal{C}$, $\mathrm{w}_{r_j}^{\mathcal{C}} \leq \mathrm{w}_{r_j}^{\mathcal{A}}\ \forall j \in I_{\mathcal{R}},$.

Definition 5.12 (Task Execution)**.** *We say that a coalition* $\mathcal{C}$ *can execute a given task* τ *if and only if*

$$\forall j \in I_{\mathcal{R}},\ \mathrm{w}_{r_j}^{\mathcal{C}} \geq \mathrm{w}_{r_j}^{\tau}. \tag{5.13}$$

Remark 5.2. *By condition (5.10), the cardinality of a coalition* $\mathcal{C}$ *that can execute a task is always greater than 1, i.e.,* $|\mathcal{C}| > 1$. *Consequently, the elements in* $\mathcal{C}$ *form a multi-agent system.*

Under the previous assumptions, we can formulate the multi-agent coalition formation problem.

Problem 5.1. Given $\mathcal{R}$ a set of M resources required to execute a task τ whose resource requirement vector is defined by (5.6), also given $\mathcal{A}$ a set of N robotic agents whose resource capability vectors are defined by (5.7) under the conditions (5.9) and (5.10), then our goal is to find a multi-agent coalition $\mathcal{C} \subseteq \mathcal{A}$ such that $\mathcal{C}$ satisfies condition (5.13), *i.e.*, $\mathcal{C}$ can execute the task τ.

Next, we present the methodology to solve this problem. Also, we apply the definitions given in Section 5.1.2 to measure the heterogeneity of the computed coalition.

5.3 Methodology

First, we represent the relation between the set of resources $\mathcal{R}$ and the set of agents $\mathcal{A}$ as a graph and then we show that it is a bipartite graph. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a weighted graph where

- $\mathcal{V} = \mathcal{R} \cup \mathcal{A}$ is the set of vertices,
- $\mathcal{E} = \{[a_i \;\; r_j] \mid \mathbf{W}^{\mathcal{A}}(i,j) > 0, i \in I_{\mathcal{A}}, j \in I_{\mathcal{R}}\}$ is the set of edges, and
- $\mathcal{W} = \{\mathrm{w}_{ij} = \mathbf{W}^{\mathcal{A}}(i,j) \mid \mathbf{W}^{\mathcal{A}}(i,j) > 0, i \in I_{\mathcal{A}}, j \in I_{\mathcal{R}}\}$ is the set of weights.

Proposition 5.1. *The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ that represents the relation between the sets $\mathcal{R}$ and $\mathcal{A}$ is a bipartite graph.*

Proof. From our definition of $\mathcal{V}$, we know that $\mathcal{V} = \mathcal{R} \cup \mathcal{A}$. Since $\mathcal{R}$ is the set of resources and $\mathcal{A}$ is the set of agents, it is clear that $\mathcal{R} \cap \mathcal{A} = \emptyset$. Also, we have that the subgraph $\mathcal{G}[\mathcal{R}] = (\mathcal{R}, \emptyset, \emptyset)$. Similarly, $\mathcal{G}[\mathcal{A}] = (\mathcal{A}, \emptyset, \emptyset)$. Thus, $\mathcal{G}[\mathcal{R}]$ and $\mathcal{G}[\mathcal{A}]$ are both null graphs. By Definition 5.1, we can conclude that $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ is a bipartite graph. □

An illustration of the bipartite graph $\mathcal{G}$ is shown in Figure 5.3(a). Using $\mathcal{G}$ we can get a matching $\mathcal{M}$ that maximizes the total weight of assigning resources with agents. In order to find $\mathcal{M}$, we employ the Hungarian Algorithm which is presented and adapted to our case.

5.3.1 The Hungarian Algorithm

Taking into account the definitions given in Section 5.1.1, we are in the position of explaining how the Hungarian Algorithm works. The pseudo-code of this method adjusted for our case is presented in Algorithm 5.1. For a complete discussion about

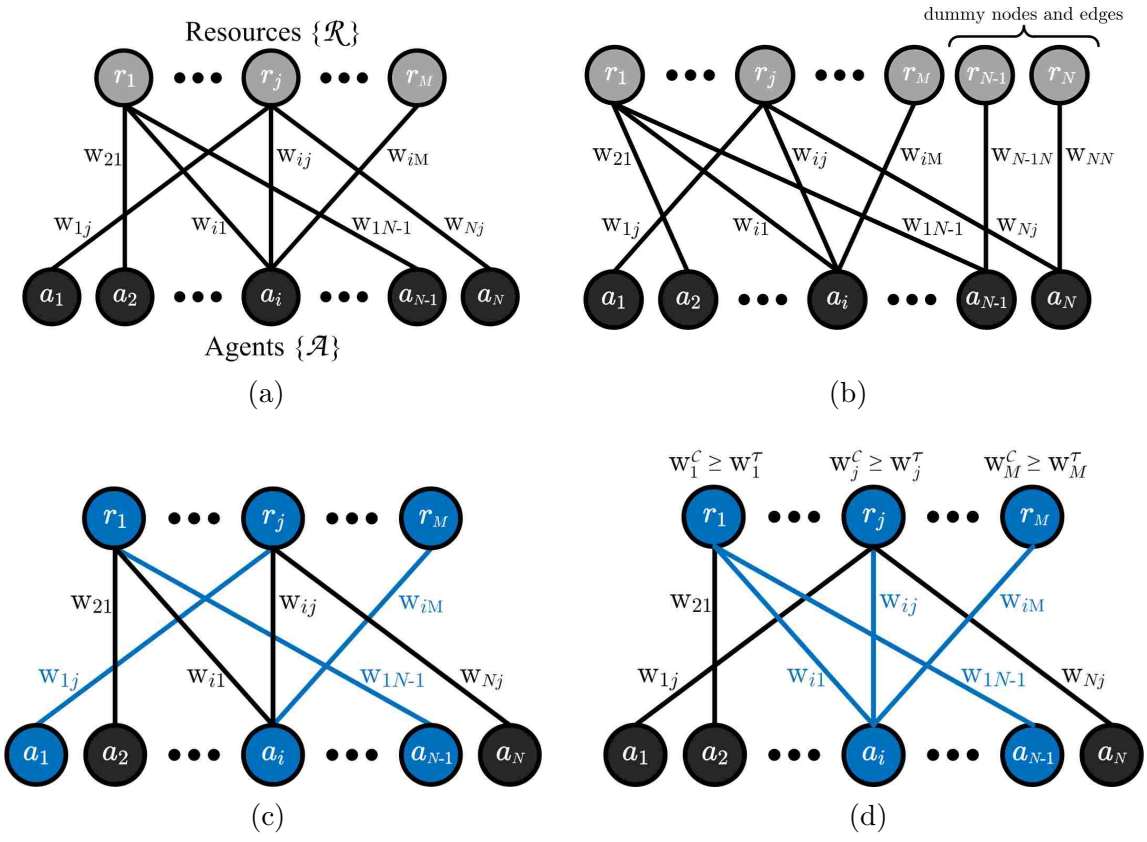


Figure 5.3: Sketch of the steps followed to compute the coalition using the Hungarian Algorithm. (a) The bipartite graph $\mathcal{G} = (\mathcal{V} = \mathcal{R} \cup \mathcal{A}, \mathcal{E}, \mathcal{W})$. (b) Adding dummy nodes and edges in order to get the same number of nodes for $\mathcal{R}$ and $\mathcal{A}$. (c) Matching obtained by applying the Hungarian Algorithm (the blue edges are those that are part of the matching). (d) Coalition formed based on the matching such that condition (5.13) holds.

this method and the use of bipartite graphs in solving assignment problems, the reader is referred to [125–127].

Since the Hungarian Algorithm cannot work with node sets of different size, a set of dummy nodes is created at the beginning to guarantee node sets of equal size, see line 4. Then, an initial labeling that satisfies (5.1) is implemented where all the nodes in $\mathcal{R}$ are labeled to zero and the label for each node in $\mathcal{A}$ is set to

Algorithm 5.1 Hungarian Method

Require: The bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$.

$N \leftarrow$ number of nodes in $\mathcal{A}$
$M \leftarrow$ number of nodes in $\mathcal{R}$
$n \leftarrow \max(N, M)$
if $|N - M| > 0$ **then**
    Fill graph $\mathcal{G}$ with $D = |N - M|$ dummy vertices with dummy weights
for $i = 1, \ldots, n$ **do**
    $\ell(a_i) \leftarrow \max_{j=1,\ldots,n} \mathrm{w}_{r_j}^{a_i}$
    $\ell(r_i) \leftarrow 0$
$\mathcal{M} \leftarrow \emptyset$ ▷ Initial matching is assumed empty
flagInitSets $\leftarrow$ true
repeat
    if flagInitSets = true **then**
        Find a_i that is free
        $u \leftarrow a_i$
        $\mathcal{S} \leftarrow \{a_i\}$
        $\mathcal{T} \leftarrow \emptyset$
        flagInitSets $\leftarrow$ false
    Compute $\mathcal{N}_{\mathcal{E}}(\mathcal{S})$ ▷ The set of all neighbors of $\mathcal{S}$
    if $\mathcal{N}_{\mathcal{E}}(\mathcal{S}) = \mathcal{T}$ **then**
        ▷ Update labeling
        $\Delta \leftarrow \min_{a_i \in \mathcal{S}, r_j \in \mathcal{T}} \{\ell(a_i) + \ell(r_j) - \mathrm{w}_{r_j}^{a_i}\}$
        if $a_i \in \mathcal{S}$ **then**
            $\ell(a_i) \leftarrow \ell(a_i) - \Delta$
        if $r_j \in \mathcal{T}$ **then**
            $\ell(r_j) \leftarrow \ell(r_j) + \Delta$
    else
        Pick a node $y \in \mathcal{N}_{\mathcal{E}}(\mathcal{S}) - \mathcal{T}$
        if y is free **then**
            Do the augmentation procedure ▷ Definition 5.7
            flagInitSets $\leftarrow$ true
        else
            Find the node z which y is matched to
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{z\}$
            $\mathcal{T} \leftarrow \mathcal{T} \cup \{y\}$
until $\mathcal{M}$ is perfect matching

Output: the matching $\mathcal{M}$

the maximum weight of all the edges which are connected to that node, line 5. In this fashion, we guarantee that each node in $\mathcal{A}$ is connected to at least one tight edge. After initializing the matching $\mathcal{M}$ as empty, the main loop of the method starts. The algorithm works with two sets $\mathcal{S}$ and $\mathcal{T}$. Informally speaking, $\mathcal{S}$ contains vertices in $\mathcal{A}$ which are endpoints of potential matching extensions and $\mathcal{T}$ contains vertices in $\mathcal{R}$ which are already part of the matching. At each iteration $\mathcal{S}$ and $\mathcal{T}$ are initialized if it is required. For this initialization, $\mathcal{T} = \emptyset$ while a free node $u = a_i \in \mathcal{A}$ is picked to add it to $\mathcal{S}$, *i.e.*, $\mathcal{S} = \{u\}$. Then, $\mathcal{N}_{\mathcal{E}}(\mathcal{S})$ is computed employing (5.2) in order to compare it with $\mathcal{T}$. If $\mathcal{N}_{\mathcal{E}}(\mathcal{S}) = \mathcal{T}$, the labels need to be updated since we could not find potential candidates to add to the matching. This update is done by following the rule given in lines 20 to 25. After this update, the method recomputes $\mathcal{N}_{\mathcal{E}}(\mathcal{S})$. In contrast, if $\mathcal{N}_{\mathcal{E}}(\mathcal{S}) \neq \mathcal{T}$, a node $y \in \mathcal{N}_{\mathcal{E}}(\mathcal{S}) - \mathcal{T}$ is picked. If y is free then the matching is extended using the augmenting path from u to y and it is required to initialize again $\mathcal{S}$ and $\mathcal{T}$. Otherwise, y is already matched to some node $z \in \mathcal{A}$. Thus, the sets $\mathcal{S}$ and $\mathcal{T}$ are enlarged, see lines 32 and 33. The algorithm stops if $\mathcal{M}$ is a perfect matching. Using the matching $\mathcal{M}$, we can obtain a coalition considering the agents in $\mathcal{A}$ associated with all the edges in $\mathcal{M}$. We denote this coalition as $\mathcal{C}_{\mathcal{M}}$. However, $\mathcal{C}_{\mathcal{M}}$ does not necessarily satisfy condition (5.13). This means that the coalition $\mathcal{C}_{\mathcal{M}}$ cannot necessary execute the task. Also, a proper subset of $\mathcal{C}_{\mathcal{M}}$ can satisfy condition (5.13). Thus, it can be possible to have a coalition that accomplish the task with less agents than $\mathcal{C}_{\mathcal{M}}$. Next, we present an algorithm to build a coalition based on the matching obtained from the Hungarian Algorithm overcoming these issues.

5.3.2 Building a Coalition from a Matching

The method for building a coalition $\mathcal{C}$ from the matching, computed using the Hungarian Algorithm, is summarized in Algorithm 5.2. Before finding $\mathcal{C}$, we check first if it is possible to form $\mathcal{C}$ such that it accomplishes the given task τ (see line 1). The

following proposition states the necessary and sufficient condition to guarantee that a coalition which can execute the task exists.

Proposition 5.2. *If and only if*

$$\forall j \in I_{\mathcal{R}},\ \mathrm{w}_{r_j}^{\mathcal{A}} \geq \mathrm{w}_{r_j}^{\tau}, \tag{5.14}$$

then a coalition $\mathcal{C}$ that can execute the task τ exists.

Proof. ($\Rightarrow$) Assume that $\forall j \in I_{\mathcal{R}}\ \mathrm{w}_{r_j}^{\mathcal{A}} \geq \mathrm{w}_{r_j}^{\tau}$. Then, pick as coalition the whole set of agents, *i.e.*, set $\mathcal{C} = \mathcal{A}$. Therefore, $\mathcal{C} = \mathcal{A}$ can execute the task τ according to Definition 5.12.

($\Leftarrow$) Let $\mathcal{C} \subseteq \mathcal{A}$ be a coalition that can execute the task τ. From Definition 5.12, we know that $\forall j \in I_{\mathcal{R}},\ \mathrm{w}_{r_j}^{\mathcal{C}} \geq \mathrm{w}_{r_j}^{\tau}$. Also, we know from Remark 5.1 that for any coalition $\forall j \in I_{\mathcal{R}},\ \mathrm{w}_{r_j}^{\mathcal{A}} \geq \mathrm{w}_{r_j}^{\mathcal{A}}$. Consequently, we get that $\forall j \in I_{\mathcal{R}},\ \mathrm{w}_{r_j}^{\mathcal{A}} \geq \mathrm{w}_{r_j}^{\tau}$. □

Remark 5.3. *From this proposition, even having as a coalition the whole set of agents, it is not possible to satisfy condition (5.13) if $\exists j \in I_{\mathcal{R}},\ \mathrm{w}_{r_j}^{\tau} > \mathrm{w}_{r_j}^{\mathcal{A}}$.*

After checking condition (5.14), see line 1 in Algorithm 5.2, we initialize the coalition $\mathcal{C}$ as empty. Also, we initialize a vector that helps us to know when the algorithm should stop since we found a coalition. This vector is the insufficient capability resource vector $\mathbf{w}^{\tau,\mathcal{C}}$ and it is defined by

$$\mathbf{w}^{\tau,\mathcal{C}} = \begin{bmatrix} \mathrm{w}_{r_1}^{\tau,\mathcal{C}} & \mathrm{w}_{r_2}^{\tau,\mathcal{C}} & \dots & \mathrm{w}_{r_M}^{\tau,\mathcal{C}} \end{bmatrix}, \tag{5.15}$$

where

$$\forall j \in I_{\mathcal{R}}, \mathrm{w}_{r_j}^{\tau,\mathcal{C}} = \begin{cases} \mathrm{w}_{r_j}^{\tau} - \mathrm{w}_{r_j}^{\mathcal{C}} & \text{if } \mathrm{w}_{r_j}^{\tau} \geq \mathrm{w}_{r_j}^{\mathcal{C}}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.16}$$

Notice that $\mathbf{w}^{\tau,\mathcal{C}} = \mathbf{w}^{\tau}$ when $\mathcal{C} = \emptyset$. Conversely, $\mathbf{w}^{\tau,\mathcal{C}} = \mathbf{0}$ when $\mathcal{C}$ can execute the task, *i.e.*, when $\mathcal{C}$ satisfies condition (5.13). The bold zero "**0**", denotes the zero

vector of appropriate dimensions. Also, notice that if $\exists j \in I_{\mathcal{R}}, \mathrm{w}_{r_j}^{\tau,\mathcal{C}} = 0$ then the coalition already satisfies the resource r_j required by task τ.

If $\mathbf{w}^{\tau,\mathcal{C}} \neq \mathbf{0}$ (see lines 4 to 8), we form the bipartite graph and use the Hungarian Method to find a matching $\mathcal{M}$. Once $\mathcal{M}$ is obtained, we select iteratively a_i that is assigned to the resource with the maximum requirement value in the vector $\mathbf{w}^{\tau}$ according to the matching $\mathcal{M}$. Next, we check if a_i is already in the coalition $\mathcal{C}$. If it is not, we add it to $\mathcal{C}$ and update the values of $\mathbf{w}^{\mathcal{C}}$ and $\mathbf{w}^{\tau,\mathcal{C}}$ (lines 14 to 15). If the update allows that $\mathbf{w}^{\tau,\mathcal{C}} = \mathbf{0}$, the algorithm stops since we find a coalition $\mathcal{C}$. Otherwise, we continue checking the matching $\mathcal{M}$ if it is required, *i.e.*, if $\mathbf{w}^{\tau,\mathcal{C}}$ is still not equal to the zero vector. We continue generating a new matching $\mathcal{M}$ by updating the bipartite graph without taking into account any more of the agents already in $\mathcal{C}$, neither the resources that $\mathcal{C}$ already has (lines 5 and 6). The steps to find the coalition $\mathcal{C}$ are depicted in Figure 5.3.

5.3.3 Heterogeneity of the Formed Coalition

Once we found a coalition, we are interested in measure its heterogeneity according to the definitions given in Section 5.1.2. Thus, we use (5.5) for the case of the coalition $\mathcal{C}$ obtained as a result of applying Algorithm 5.2.

We assume that the agents in set $\mathcal{A}$ have equal probabilities to be chosen. Then, $p_i = 1/N \ \forall i = 1, \ldots, N$. Now, assume that the coalition $\mathcal{C}$ which is given as output of Algorithm 5.2 is of the form

$$\mathcal{C} = \{c_1, ..., c_n\} \text{ with } c_i \in \mathcal{A} \ \ \forall i = 1, ..., n.$$

By Definition 5.8, the entropy of $\mathcal{C}$ is equal to

$$E(\mathcal{C}) = -\frac{n}{N} \log \left(\frac{1}{N} \right) = \frac{n}{N} \log N. \tag{5.17}$$

The Rao's quadratic entropy of $\mathcal{C}$ given in Definition 5.9 requires the metric d_{ij} which

Algorithm 5.2 Building Coalition

Require: $\mathbf{w}^{\tau}$ resource requirement vector, $\mathbf{W}^{\mathcal{A}}$ capability resource matrix. $\mathcal{A} = \{a_1, \ldots, a_M\}$ set of agents. $\mathcal{R} = \{r_1, \ldots, r_M\}$ set of resources.

if $\mathrm{w}^{\mathcal{A}}_{r_j} \geq \mathrm{w}^{\tau}_{r_j} \forall j = 1, \ldots, M$ **then**
  $\mathcal{C} \leftarrow \emptyset$ ▷ Initial coalition is empty
  $\mathbf{w}^{\tau,\mathcal{C}} \leftarrow \mathbf{w}^{\tau}$ ▷ Initial vector of insufficient capability resources
  while $\mathbf{w}^{\tau,\mathcal{C}} \neq \mathbf{0}$ **do**
    $\mathbf{W}^{\mathcal{A}}(i,:) \leftarrow 0 \; \forall i \in I_{\mathcal{A}}, a_i \in \mathcal{C}$ ▷ so the Hungarian Method does not considers the agents already in $\mathcal{C}$
    $\mathbf{W}^{\mathcal{A}}(:,j) \leftarrow 0 \; \forall j \in I_{\mathcal{R}}, \mathrm{w}^{\tau,\mathcal{C}}_{r_j} = 0$ ▷ so the Hungarian Method does not considers the resources already present in $\mathcal{C}$
    Using $\mathbf{W}^{\mathcal{A}}$, $\mathcal{R}$ and $\mathcal{A}$, generate the bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$
    $\mathcal{M} \leftarrow$ Hungarian Method($\mathcal{G}$)
    while $\mathcal{M} \neq \emptyset$ **do**
      $j \leftarrow$ argmax $\mathbf{w}^{\tau}$
      find a_i assigned to r_j from the matching $\mathcal{M}$
      if $a_i \notin \mathcal{C}$ **then**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_i\}$
        Compute $\mathbf{w}^{\mathcal{C}}$ ▷ see (5.11) and (5.12)
        Compute $\mathbf{w}^{\tau,\mathcal{C}}$ ▷ see (5.15) and (5.16)
        if $\mathbf{w}^{\tau,\mathcal{C}} = \mathbf{0}$ **then break**
      $\mathcal{M} \leftarrow \mathcal{M} - \{a_i\}$
else
  $\mathcal{C} \leftarrow \emptyset$

Output: the coalition $\mathcal{C}$

we define as

$$d^2_{ij} = \frac{1}{M} \sum_{k=1}^{M} \left(\mathrm{w}^{c_i}_{r_k} - \mathrm{w}^{c_j}_{r_k} \right)^2 . \tag{5.18}$$

This metric gives us an idea of how distinct the resources available at each agent of $\mathcal{C}$ are. Notice that (5.18) can be simply expressed as

$$d^2_{ij} = \frac{1}{M} \left\| \mathbf{w}^{c_i} - \mathbf{w}^{c_j} \right\| . \tag{5.19}$$

From (5.4), the Rao's quadratic entropy of $\mathcal{C}$ is then

$$Q(\mathcal{C}) = \frac{1}{N^2 M} \sum_{i=1}^{n} \sum_{j=1}^{n} \left\| \mathbf{w}^{c_i} - \mathbf{w}^{c_j} \right\| . \tag{5.20}$$

Substituting (5.17) and (5.20) in the definition of heterogeneity (Definition 5.10), we find that

$$H(\mathcal{C}) = \frac{n \log N}{N^3 M} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\mathbf{w}^{c_i} - \mathbf{w}^{c_j}\| \,. \tag{5.21}$$

Once the coalition $\mathcal{C}$ is obtained from Algorithm 5.2, we use (5.21) to compute its heterogeneity. In the next section, we present and discuss the results of applying the proposed methodology to solve Problem 5.1 for different cases.

5.4 Simulation Results

We implement both Algorithms 5.1 and 5.2, and run two experiments. For all the tests, the dimension of the set of resources $\mathcal{R}$ is set to 5, *i.e.*, $M = 5$, and the dimension of the set of agents $\mathcal{A}$ to 20, *i.e.*, $N = 20$. Also, the entries of the resource requirement vector $\mathbf{w}^{\tau}$ are randomly generated such that

$$\forall j \in I_{\mathcal{R}}, \;\; 0 < \gamma \leq \mathrm{w}^{\tau}_{r_j} \leq \Gamma,$$

where γ and Γ are lower and upper bounds, respectively.

Based on the assumptions in Section 5.2, we need to fill out the capability resource matrix $\mathbf{W}^{\mathcal{A}}$ so that its entries for each agent satisfy conditions (5.9) and (5.10). Therefore, we generate its elements randomly according to

$$\mathbf{W}^{\mathcal{A}}(i,j) = \begin{cases} 0 & \text{with probability } 1-\epsilon, \\ \text{rand. num.} \;\in [1 \;\; \mathrm{w}^{\tau}_{r_j}] & \text{with probability } \epsilon, \end{cases} \tag{5.22}$$

for all $i \in I_{\mathcal{A}}$ and for all $j \in I_{\mathcal{R}}$ with $\epsilon \in [0 \quad 1]$. Therefore, we can guarantee conditions (5.9) and (5.10). Notice that a zero in $\mathbf{W}^{\mathcal{A}}(i,j)$ means that the robotic agent a_i does not have the resource r_j. Thus, using (5.22), we are able to simulate the effect of having a sparse matrix and then check how it affects to the coalition.

In the first experiment, we generate coalitions assuming that the value of ϵ goes from 0.1 to 0.9 in steps of 0.05. We conducted ten trials for each value of ϵ. For each trial, we calculate the following metrics for the generated coalition $\mathcal{C}$,

1. the cardinality of the coalition given by $c = |\mathcal{C}|$,
2. the weight of the coalition given by

$$\mathrm{w} = \sum_{j=1}^{M} \frac{\mathrm{w}_{r_j}^{\mathcal{C}} - \mathrm{w}_{r_j}^{\tau}}{\mathrm{w}_{r_j}^{\tau}}, \text{ and}$$

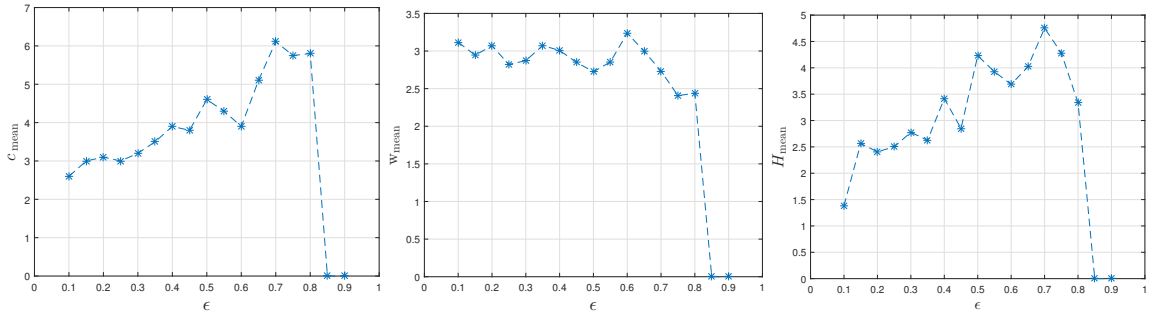
3. the heterogeneity of the coalition given by (5.21).

The mean values of these metrics with respect to the value of ϵ are shown in Figure 5.4.

In the second experiment, the value of ϵ is fixed at 0.5 and we generate coalitions varying the diameter of the interval $\begin{bmatrix}\gamma & \Gamma\end{bmatrix}$. The diameter of this interval is given by $|\gamma - \Gamma|$. By performing this experiment, we consider the effect of increasing the resource requirements associated with a given task on the coalition. Once again, we conduct ten trials and compute the same metrics described before. The results for this experiment are given in Figure 5.5.

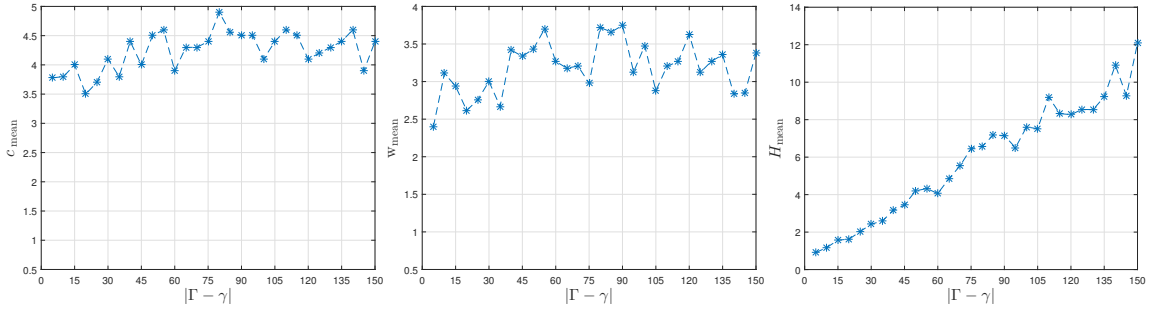
Notice that as ϵ approaches to 1 in (5.22), the matrix $\mathbf{W}^{\mathcal{A}}$ becomes more and more sparse, *i.e.*, more and more elements of $\mathbf{W}^{\mathcal{A}}$ are zero. This implies that the agents have less and less resources available to execute the task. Thus, it is possible that condition (5.14) does not hold and then there is no coalition that can execute τ.

From the results of our first experiment given in Figure 5.4, it is clear that no coalition exists when $\epsilon > 0.8$ because the mean of the coalition cardinality c_{mean} is zero. In addition, the mean of the coalition weight w_{mean} decreases as $\mathbf{W}^{\mathcal{A}}$ becomes more sparse, *i.e.*, as ϵ goes to one. We expected this result since $\mathbf{w}^{a_i}$ has more and more zero elements for more and more agents $a_i \in \mathcal{A}$ as the probability ϵ goes



(a) (b) (c)

Figure 5.4: Simulation results for the experiment where ϵ goes from 0.1 to 0.9 in steps of 0.05. We run 10 iterations for each trial. The parameters for all the trials were $M = 5, N = 20, \gamma = 5$, and $\Gamma = 50$.



(a) (b) (c)

Figure 5.5: Simulation results for the experiment where the diameter of the interval $[\gamma \quad \Gamma]$ goes from 5 to 150 in steps of 5. We run 10 iterations for each trial. The parameters for all the trials were $M = 5, N = 20, \gamma = 5$, and $\epsilon = 0.5$.

to 1. As for the mean heterogeneity H_{mean}, notice that it grows with ϵ and it is zero for the cases that a coalition cannot be found. This growth happens because the cardinality of the coalition increases because there are more robots with fewer resources capabilities as ϵ goes to one.

For our second experiment (see Figure 5.5), c_{mean} and w_{mean} are not drastically affected if we increase the range of the values that the resources required by a task can

have. However, the heterogeneity grows again as this range increases. This means that we compute coalitions with the same number of robots on average, but their capability resource vector is increasing to compensate higher resource requirements.

By carrying out these experiments, it is clear that heterogeneity defined in (5.5), is directly related to the capability that a multi-agent system has to execute a given task. We would need more robots with less resources capabilities as in the case of the first experiment or fewer robots with higher resource capabilities as in the second experiment.

5.5 Conclusions

In this chapter, we reviewed and formulated the multi-agent coalition formation problem. We proposed a solution to this problem by first representing the relation agents-resources as a weighted bipartite graph and then obtaining a matching using the Hungarian Algorithm. Based on this matching, we generated a coalition such that it has the resource capabilities required for a given task. Also, we measured the heterogeneity of the resultant coalition. We ran numerical simulations to validate the proposed algorithms. We showed a direct relationship between the task requirements and the heterogeneity of the multi-agent system selected to execute the task. Depending on the task requirements, a team made of a large number of agents with limited capabilities might be needed or a team with fewer robots but with higher resources.

Future work involves exploiting the heterogeneity metric as a design parameter of multi-robotic teams such that the formed teams are heterogeneous enough to carry out a given mission. Varying heterogeneity as a design parameter can potentially have a critical implication on the whole network capabilities. For example, it can improve the robustness and adaptability of a robotic team.

Chapter 6

Concluding Remarks and Future Work

In this dissertation, we focused on developing coordinated control algorithms for taking advantage of the unique dynamics, capabilities and other resources that are present in a heterogeneous network made of ground and aerial robotic platforms. Transportation of a micro-sized mobile ground sensor as a cable-suspended load is, for example, a relevant capability of a flying robot. A critical step of this useful application is the lift of the payload from the ground. Therefore, we defined a hybrid system that represents particular operational modes of the quadrotor-load system. Furthermore, we showed that the hybrid system is differentially-flat. Exploiting this property, we generated a minimum jerk trajectory based on a series of waypoints which are associated with the modes of the hybrid system. We designed a geometric controller to track the generated trajectory. In addition, we combine this controller with least-squares parameter estimation to perform the lift maneuver even in the case of uncertainty knowing the total load mass. We modified the proposed control scheme to validate our method in a real world application.

Assuming the mobile ground sensors are already deployed in a cluttered envi-

ronment and the aerial vehicles are better equipped for communications purposes, we developed coordinated control techniques for the heterogeneous robotic network where a target area can be sensed while the network connectivity is maintained. We designed decentralized control laws to the mobile sensors, so a predefined region is explored while avoiding collisions with other sensors and obstacles. In the case of the aerial communication relays, we first solved the case of one relay that has to maintain an adequate relative position with a group of ground sensors to guarantee their connectivity with a base station. Then, we extended our approach to the case of multiple aerial relays by using learning techniques to enhance the received signal strength between the mobile sensors and the relays which in turn improves the network connectivity.

Even though RF is the standard technology for wireless operations of robotic platforms, we studied designs for OW communications between aerial and ground robots. Our intention is not to replace RF communication systems with OW technology. On the contrary, we seek to combine them to achieve a robust connectivity. By considering a realistic model for the optical link, we defined a connectivity cone on top of the receiver where a minimum transmission rate can be guaranteed. Then, we presented a target tracking controller for an aerial optical transmitter so that it remains inside of this cone even having noisy measurements of the position of a ground optical receiver. In addition, we considered the case when both the aerial and the ground platforms are equipped with a hybrid RF/OW communication system. We exploited this bi-modal communication scheme and antenna diversity to solve the problem of autonomously localizing the ground sensor relative to the aerial vehicle. Numerical simulations which included physics based optical channel modeling validated the performance of the proposed control strategy for all these cases.

In order to fully exploit the benefits of using a heterogeneous network it is also required to study how the members of the network can be selected. Thus, we studied the formation of such a multi-agent system. We represented the relation between

agent's capabilities and task requirements as a graph, specifically as a bipartite graph. By using methods for this type of graph, we were able to form a group of agents capable of accomplishing the requirements imposed by a given task. Also, we analyzed and measured the heterogeneity of the formed group. From our results, it is clear that there is a direct relation between the amount of the resources required to execute a task and the heterogeneity of the multi-agent system selected to execute such task.

Future work will be centered on extending the proposed methods described in this dissertation. For example, the work on cable-suspended load lift has the intuitive extension of using multiple quadrotors to lift heavy loads. An important challenge to consider on this application is the optimal location of the anchor or attachment point on the load such that the aerial vehicles apply a maximum tension force while avoiding collisions with other quadrotors. In reference to the coordinated control under connectivity constrains, one immediate problem that needs to be solved is how to add the information about the mobile sensors to the learning algorithm to achieve a better result for the connectivity index. We have used only the received signal strength to guide the aerial relays, but the number of sensors inside the cover regions of each relay can be employed, for example, as a new state input of the learning strategy. Also, an important avenue for future work is the experimental validation of the methods developed in this dissertation. The optical wireless communication link is of particular interest for our lab since, from the best of our knowledge, this communication system has not been tested yet for micro-sized ground and aerial platforms and it has the potential to dramatically expand communication rates when it is combined with RF-based systems. Another important expansion of this work is about how heterogeneity can be used as design parameter to pick up a multi-agent team. Possibly with a better understanding of how heterogenity can be analyzed for the case of a multi-robotic network, one can determine the range of tasks that the given network can or cannot accomplish.

References

[1] R. Cortez, R. Fierro, and J. Wood, "Connectivity maintenance of a heterogeneous sensor network," in *Distributed Autonomous Robotic Systems*, ser. Springer Tracts in Advanced Robotics, A. Martinoli, F. Mondada, N. Correll, G. Mermoud, M. Egerstedt, M. A. Hsieh, L. E. Parker, and K. Sty, Eds. Springer Berlin Heidelberg, 2013, vol. 83, pp. 33–46.

[2] L. Iocchi, D. Nardi, M. Piaggio, and A. Sgorbissa, "Distributed coordination in heterogeneous multi-robot systems," *Autonomous Robots*, vol. 15, no. 2, pp. 155–168, 2003.

[3] N. Bezzo, M. Anderson, R. Fierro, and J. Wood, "A real world coordination framework for connected heterogeneous robotic systems," in *International Symposium on Distributed Autonomous Robotic Systems*, Baltimore, MD, November 2012.

[4] S. Gil, M. Schwager, B. J. Julian, and D. Rus, "Optimizing communication in air-ground robot networks using decentralized control," in *IEEE International Conference on Robotics and Automation (ICRA), 2010*, 2010, pp. 1964–1971.

[5] T. Service and J. Adams, "Coalition formation for task allocation: theory and algorithms," *Autonomous Agents and Multi-Agent Systems*, vol. 22, no. 2, pp. 225–248, 2011.

[6] A. O. Pullin, N. J. Kohut, D. Zarrouk, and R. S. Fearing, "Dynamic turning of 13 cm robot comparing tail and differential drive," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 5086–5093.

[7] A. Wallar, E. Plaku, and D. A. Sofge, "Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 969–980, July 2015.

[8] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, "Valve turning using a dual-arm aerial manipulator," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2014, pp. 836–841.

[9] V. Ghadiok, J. Goldin, and W. Ren, "Autonomous indoor aerial gripping using a quadrotor," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2011, pp. 4645–4651.

[10] R. Spica, A. Franchi, G. Oriolo, H. Bulthoff, and P. Giordano, "Aerial grasping of a moving target with a quadrotor UAV," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2012, pp. 4985–4992.

[11] J. Thomas, G. Loianno, J. Polin, K. Sreenath, and V. Kumar, "Toward autonomous avian-inspired grasping for micro aerial vehicles," *Bioinspiration & Biomimetics*, vol. 9, no. 2, p. 025010, 2014.

[12] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. Mueller, J. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *Control Systems, IEEE*, vol. 34, no. 4, pp. 46–64, Aug 2014.

[13] I. Beloti Pizetta, A. Santos Brandão, and M. Sarcinelli-Filho, "Modelling and control of a PVTOL quadrotor carrying a suspended load," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2015, pp. 444–450.

[14] I. Palunko, P. Cruz, and R. Fierro, "Agile load transportation : safe and efficient load manipulation with aerial robots," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 69–79, 2012.

[15] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Automated aerial suspended cargo delivery through reinforcement learning," *Artificial Intelligence*, 2014, in press - available online. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0004370214001416

[16] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2216–2222.

[17] K. Sreenath, N. Michael, and V. Kumar, "Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 4888–4895.

[18] F. Goodarzi and T. Lee, "Dynamics and control of quadrotor UAVs transporting a rigid body connected via flexible cables," in *American Control Conference (ACC), 2015*, July 2015, pp. 4677–4682.

[19] J. Fink, A. Ribeiro, and V. Kumar, "Robust control of mobility and communications in autonomous robot teams," *IEEE Access*, vol. 1, pp. 290–309, 2013.

[20] P. Cruz, R. Fierro, W. Lu, and S. Ferrari, "Maintaining robust connectivity in heterogeneous robotic networks," in *Proc. SPIE 8741, Unmanned Systems Technology XV*, vol. 8741, 2013, pp. 87 410N–87 410N–15.

[21] H. M. La, R. Lim, and W. Sheng, "Multirobot cooperative learning for predator avoidance," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 52–63, Jan 2015.

[22] G. A. S. Pereira, A. K. Das, V. Kumar, and M. F. M. Campos, "Decentralized motion planning for multiple robots subject to sensing and communication constraints," in *Proceedings of the Second MultiRobot Systems Workshop*. Kluwer Academic Press, 2003, pp. 267–278.

[23] J. Esposito and T. Dunbar, "Maintaining wireless connectivity constraints for swarms in the presence of obstacles," in *IEEE International Conference on Robotics and Automation (ICRA), 2006*, May 2006, pp. 946 – 951.

[24] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, 2007.

[25] M. A. Hsieh, A. Cowley, V. Kumar, and C. J. Taylor, "Maintaining network connectivity and performance in robot teams," *Journal of Field Robotics*, vol. 25, no. 1-2, pp. 111–131, 2008.

[26] P. Robuffo Giordano, A. Franchi, C. Secchi, and H. H. Bülthoff, "A passivity-based decentralized strategy for generalized connectivity maintenance," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 299–323, 2013.

[27] H. Elgala, R. Mesleh, and H. Haas, "Indoor optical wireless communication: potential and state-of-the-art," *IEEE Communications Magazine*, vol. 49, no. 9, pp. 56–62, September 2011.

[28] D. Borah, A. Boucouvalas, C. Davis, S. Hranilovic, and K. Yiannopoulos, "A review of communication-oriented optical wireless systems," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, 2012.

[29] Z. Ghassemlooy, W. Popoola, and S. Rajbhandari, *Optical Wireless Communications: System and Channel Modelling with MATLAB®*. Taylor & Francis, 2012.

[30] Z. C. Bagley, D. H. Hughes, J. C. Juarez, P. Kolodzy, T. Martin, M. Northcott, H. A. Pike, N. D. Plasson, B. Stadler, L. B. Stotts, and D. W. Young, "Hybrid optical radio frequency airborne communications," *Optical Engineering*, vol. 51, no. 5, pp. 055 006–1–055 006–25, 2012.

[31] D. O'brien, L. Zeng, H. Le-Minh, G. Faulkner, J. Walewski, and S. Randel, "Visible light communications: Challenges and possibilities," in *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sept 2008, pp. 1–5.

[32] H. Chowdhury and M. Katz, "Data download on move in indoor hybrid (radio-optical) WLAN-VLC hotspot coverages," in *IEEE Vehicular Technology Conference (VTC Spring)*, June 2013, pp. 1–5.

[33] Z. Xu and B. Sadler, "Ultraviolet communications: Potential and state-of-the-art," *IEEE Communications Magazine*, vol. 46, no. 5, pp. 67–73, May 2008.

[34] R. J. Drost, T. J. Moore, and B. M. Sadler, "Ultraviolet scattering propagation modeling: analysis of path loss versus range," *J. Opt. Soc. Am. A*, vol. 30, no. 11, pp. 2259–2265, Nov 2013. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-30-11-2259

[35] (2015, July) New milestones in connectivity lab's aircraft and laser programs. Newsroom, Facebook. [Online]. Available: http://newsroom.fb.com/news/2015/07/new-milestones-in-connectivity-labs-aircraft-and-laser-programs/

[36] I. Vasilescu, C. Detweiler, M. Doniec, D. Gurdan, S. Sosnowski, J. Stumpf, and D. Rus, "AMOUR V: A hovering energy efficient underwater robot capable of dynamic payloads," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 547–570, 2010.

[37] M. Doniec, I. Topor, M. Chitre, and D. Rus, "Autonomous, localization-free underwater data muling using acoustic and optical communication," in *Experimental Robotics*, ser. Springer Tracts in Advanced Robotics, J. P. Desai, G. Dudek, O. Khatib, and V. Kumar, Eds., vol. 88. Springer International Publishing, 2013, pp. 841–857.

[38] M. Doniec, A. Xu, and D. Rus, "Robust real-time underwater digital video streaming using optical communication," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 5117–5124.

[39] I. Rust and H. Asada, "A dual-use visible light approach to integrated communication and localization of underwater robots with application to non-destructive nuclear reactor inspection," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 2445–2450.

[40] B. Tian, F. Zhang, and X. Tan, "Design and development of an LED-based optical communication system for autonomous underwater robots," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2013, pp. 1558–1563.

[41] J. M. Kahn, R. H. Katz, and K. S. Pister, "Emerging challenges: Mobile networking for "Smart Dust"," *Journal of Communications and Networks*, vol. 2, no. 3, pp. 188–196, Sept 2000.

[42] W. S. Rabinovich, J. L. Murphy, M. Suite, M. Ferraro, R. Mahon, P. Goetz, K. Hacker, W. Freeman, E. Saint Georges, S. Uecke, and J. Sender, "Free-space optical data link to a small robot using modulating retroreflectors," in *Proc. SPIE*, vol. 7464, 2009, pp. 746 408–746 408–9.

[43] J. Derenick, C. Thorne, and J. Spletzer, "On the deployment of a hybrid free-space optic/radio frequency (FSO/RF) mobile ad-hoc network," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Aug 2005, pp. 3990–3996.

[44] L. Vig and J. Adams, "Multi-robot coalition formation," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 637–649, Aug 2006.

[45] A. Campbell and A. Wu, "Multi-agent role allocation: issues, approaches, and multiple perspectives," *Autonomous Agents and Multi-Agent Systems*, vol. 22, no. 2, pp. 317–355, 2011.

[46] L. Parker and F. Tang, "Building multirobot coalitions through automated task solution synthesis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1289–1305, July 2006.

[47] J. Chen and D. Sun, "Coalition-based approach to task allocation of multiple robots with resource constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 3, pp. 516–528, July 2012.

[48] S. Sen and J. Adams, "sA-ANT: A hybrid optimization algorithm for multirobot coalition formation," in *IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 2, Nov 2013, pp. 337–344.

[49] X. Liang and Y. Xiao, "Studying bio-inspired coalition formation of robots for detecting intrusions using game theory," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 3, pp. 683–693, June 2010.

[50] Construction Safety Association, *Helicopter Lifting, Safety Guidelines for Construction*, Ontario - Canada, 2000.

[51] UK Civil Aviation Authority, *Helicopter External Load Operations*, 4th ed., Safety Regulation Group, 2006.

[52] Y. Mostofi, M. Malmirchegini, and A. Ghaffarkhah, "Estimation of communication signal strength in robotic networks," in *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 1946–1951.

[53] J. Twigg, J. Fink, P. Yu, and B. Sadler, "Rss gradient-assisted frontier exploration and radio source localization," in *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, May 2012, pp. 889–895.

[54] E. Ackerman. (2014, July) Robocopters Haul Tons of Stuff in Afghanistan, Return Home Victorious. [Online]. Available: http://spectrum.ieee.org/automaton/robotics/military-robots/kmax-robocopters-haul-tons-of-stuff-afghanistan

[55] D. Mellinger, Q. Lindsey, M. Shomin, and V. Kumar, "Design, modeling, estimation and control for aerial grasping and manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2011, pp. 2668–2673.

[56] N. Michael, J. Fink, and V. Kumar, "Cooperative manipulation and transportation with aerial robots," *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[57] A. Faust, I. Palunko, P. Cruz, R. Fierro, and L. Tapia, "Learning swing-free trajectories for UAVs with a suspended load," in *IEEE International Conference on Robotics and Automation*, Germany, May 2013, pp. 4887–4894.

[58] T. Lee, K. Sreenath, and V. Kumar, "Geometric control of cooperating multiple quadrotor uavs with a suspended payload," in *IEEE 52nd Annual Conference on Decision and Control (CDC)*, December 2013, pp. 5510–5515.

[59] M. Manubens, D. Devaurs, L. Ros, and J. Cortés, "Motion planning for 6-D manipulation with aerial towed-cable systems," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, June 2013.

[60] W. Guofan and K. Sreenath, "Geometric control of multiple quadrotors transporting a rigid-body load," in *IEEE Annual Conference on Decision and Control (CDC)*, Dec 2014, pp. 6141–6148.

[61] S. Dai, T. Lee, and D. Bernstein, "Adaptive control of a quadrotor uav transporting a cable-suspended load with unknown mass," in *IEEE Annual Conference on Decision and Control (CDC)*, Dec 2014, pp. 6149–6154.

[62] M. Bisgaard, J. D. Bendtsen, and A. L. Cour-Harbo, "Modeling of generic slung load system," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 573–585, 2009.

[63] J. H. Gillula, G. M. Hoffmann, H. Haomiao, M. P. Vitus, and C. J. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 335–354, 2011.

[64] E. Frazzoli, M. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1077–1091, Dec 2005.

[65] P. J. Cruz and R. Fierro, "Autonomous lift of a cable-suspended load by an unmanned aerial robot," in *IEEE Conference on Control Applications (CCA)*, Oct 2014, pp. 802–807.

[66] P. J. Cruz, M. Oishi, and R. Fierro, "Lift of a cable-suspended load by a quadrotor: A hybrid system approach," in *American Control Conference (ACC)*, July 2015, pp. 1887–1892.

[67] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, September 2012.

[68] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *IEEE Conference on Decision and Control (CDC)*, December 2010, pp. 5420–5425.

[69] J. Lygeros, S. Sastry, and C. Tomlin, *Hybrid Systems: foundations, advanced topics and applications*, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 2012, (*In preparation*). [Online]. Available: http://inst.eecs.berkeley.edu/$\sim$ee291e/sp14/handouts/book.pdf

[70] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 2520–2525.

[71] M. Hehn and R. D'Andrea, "Quadrocopter trajectory generation and control," in *Proceedings of the IFAC world congress*, 2011, pp. 1485–1491.

[72] R. Stengel, *Optimal Control and Estimation*, ser. Dover Books on Advanced Mathematics. Dover Publications, 1994.

[73] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. John Wiley & Sons, Inc., 2012.

[74] J. Slotine and W. LI, *Applied Nonlinear Control*. Prentice-Hall Inc., 1991.

[75] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. Society for Industrial and Applied Mathematics, 2006.

[76] C. Lee, K. Hedrick, and K. Yi, "Real-time slip-based estimation of maximum tire-road friction coefficient," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 2, pp. 454–458, June 2004.

[77] Ascending Technologies. Product overview, AscTec Hummingbird. Last modified Jan, 2015. [Online]. Available: http://wiki.asctec.de/display/AR/AscTec+Hummingbird

[78] ——, *AscTec Hummingbird with AutoPilot User's Manual*, AscTec, 2010. [Online]. Available: http://robotics.caltech.edu/~ndutoit/wiki/images/7/70/AscTec_AutoPilot_manual_v1.0_small.pdf

[79] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz," in *IEEE International Conference on Robotics and Automation (ICRA)*, April 2007, pp. 361–366.

[80] M. Achtelik, M. Achtelik, S. Weiss, and R. Siegwart, "Onboard IMU and monocular vision based control for MAVs in unknown in- and outdoor environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 3056–3063.

[81] A. Isidori, *Nonlinear Control Systems*, ser. Communications and Control Engineering. Springer London, 1995.

[82] J. Wang, T. Bierling, L. Hcht, F. Holzapfel, S. Klose, and A. Knoll, "Novel dynamic inversion architecture design for quadrocopter control," in *Advances in Aerospace Guidance, Navigation and Control*, F. Holzapfel and S. Theil, Eds. Springer Berlin Heidelberg, 2011, pp. 261–272.

[83] National Instruments. (2016) NI CompactRIO. [Online]. Available: http://www.ni.com/compactrio/

[84] M. A. Al-Alaoui, "Al-Alaoui operator and the new transformation polynomials for discretization of analogue systems," *Electrical Engineering*, vol. 90, no. 6, pp. 455–467, 2008.

[85] National Instruments. (2016) LabVIEW Real-Time Module. [Online]. Available: http://www.ni.com/labview/realtime/

[86] N. Bezzo, B. Griffin, P. Cruz, J. Donahue, R. Fierro, and J. Wood, "A cooperative heterogeneous mobile wireless mechatronic system," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 1, pp. 20–31, Feb 2014.

[87] MARHES Lab. (2016) Cable-suspended load lifting by a quadrotor UAV. [Online]. Available: https://youtu.be/rHtw80dqcGk

[88] G. Zhang and S. Ferrari, "An adaptive artificial potential function approach for geometric sensing," in *IEEE Conference on Decision and Control, held jointly with the Chinese Control Conference (CDC/CCC 2009)*, December 2009, pp. 7903 –7910.

[89] J. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, 1991.

[90] L. Chengqing, V. Ang, H. Krishnan, and S. Y. Lim, "Virtual obstacle concept for local-minimum-recovery in potential-field based navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 983–988.

[91] S. Ferrari, M. Anderson, R. Fierro, and W. Lu, "Cooperative navigation for heterogeneous autonomous vehicles via approximate dynamic programming," in *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, 2011, pp. 121–127.

[92] P. J. Cruz, C. Abdallah, and R. Fierro, "Cooperative learning for robust connectivity in multi-robot heterogeneous networks," in *Control of Complex Systems: Theory and Applications*, K. Vamvoudakis and J. Sarangapani, Eds. Elseiver, August 2016, (To appear).

[93] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.

[94] J. Nelson, J. Almodovar, M. Gupta, and W. Mortensen, "Estimating multiple transmitter locations from power measurements at multiple receivers," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. ICASSP, April 2009, pp. 2761–2764.

[95] B. Griffin, R. Fierro, and I. Palunko, "An autonomous communications relay in gps-denied environments via antenna diversity," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 9, no. 1, pp. 33–44, 2012.

[96] G. Agnarsson and R. Greenlaw, *Graph Theory: Modeling, Applications, and Algorithms*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[97] C. Godsil and G. Royle, *Algebraic Graph Theory*, ser. Graduate Texts in Mathematics. Springer New York, 2001.

[98] F. Lewis and D. Liu, Eds., *Reinforcement Learning and Approximate Dynamic Programming for Feedback Control*, ser. IEEE Press Series on Computational Intelligence. Wiley, 2012.

[99] L. Buşoniu, R. Babuška, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 2, pp. 156–172, March 2008.

[100] H. Guo and Y. Meng, "Distributed reinforcement learning for coordinate multi-robot foraging," *Journal of Intelligent & Robotic Systems*, vol. 60, no. 3-4, pp. 531–551, 2010.

[101] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[102] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I and Vol. II*, 3rd ed. Athena Scientific, 2005 (Vol. I) and 2012 (Vol. II).

[103] L. Buşoniu, R. Babuška, B. D. Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2010.

[104] P. J. Cruz and R. Fierro, "Towards optical wireless communications between micro unmanned aerial and ground systems," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2015, pp. 669–676.

[105] P. J. Cruz, B. M. Sadler, and R. Fierro, "Sensor localization using hybrid RF/optical wireless communications for an aerial data mule," in *American Control Conference (ACC)*, July 2016, (Accepted).

[106] T. C. Shen, R. J. Drost, C. C. Davis, and B. M. Sadler, "Design of dual-link (wide- and narrow-beam) LED communication systems," *Optics Express*, vol. 22, no. 9, pp. 11 107–11 118, May 2014.

[107] J. Kahn and J. Barry, "Wireless infrared communications," *Proceedings of the IEEE*, vol. 85, no. 2, pp. 265–298, Feb 1997.

[108] A. Moreira, R. Valadas, and A. de Oliveira Duarte, "Characterisation and modelling of artificial light interference in optical wireless communication systems," in *Sixth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. Wireless: Merging onto the Information Superhighway*, vol. 1, Sep 1995, pp. 326–331 vol.1.

[109] ——, "Performance of infrared transmission systems under ambient light interference," *IEE Proceedings - Optoelectronics*, vol. 143, no. 6, pp. 339–346, Dec 1996.

[110] G. Hoffmann and C. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 32–47, Jan 2010.

[111] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 425–437, Feb 2002.

[112] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[113] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[114] W. Lu, G. Zhang, S. Ferrari, M. Anderson, and R. Fierro, "A particle-filter information potential method for tracking and monitoring maneuvering targets using a mobile sensor agent," *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, vol. 11, no. 1, pp. 47–58, 2014.

[115] S. Ge and Y. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation,*, vol. 16, no. 5, pp. 615–620, 2000.

[116] P. Schneider and D. Eberly, *Geometric Tools for Computer Graphics*, ser. The Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann Publishers, 2003.

[117] G. Fischer. (2009) Distance between a point and a triangle in 3D. Retrieved November 2014. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/22857

[118] Y. Li, M. Thai, and W. Weili, *Wireless Sensor Networks and Applications*, ser. Signals and Communication Technology. Springer US, 2008.

[119] J. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks," *IEEE Network*, vol. 15, no. 5, pp. 12–19, Sept 2001.

[120] P. Rawat, K. Singh, H. Chaouchi, and J. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of Supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.

[121] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data MULEs: modeling a three-tier architecture for sparse sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, May 2003, pp. 30–41.

[122] L. Stotts, L. Andrews, P. Cherry, J. Foshee, P. Kolodzy, W. McIntire, M. Northcott, R. Phillips, H. Pike, B. Stadler, and D. Young, "Hybrid optical RF airborne communications," *Proceedings of the IEEE*, vol. 97, no. 6, pp. 1109–1127, June 2009.

[123] W. Burgard, O. Brock, and C. Stachniss, "An experimental study of exploiting multipath fading for robot communications," in *Robotics:Science and Systems III*. MIT Press, 2008, pp. 289–296.

[124] P. J. Cruz and R. Fierro, "Building coalitions of heterogeneous agents using weighted bipartite graphs," in *54th IEEE Conference on Decision and Control (CDC)*, December 2015, pp. 2822–2828.

[125] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*. Society for Industrial and Applied Mathematics, 2009.

[126] G. Agnarsson and R. Greenlaw, *Graph Theory: Modeling, Applications, and Algorithms*, ser. Featured Titles for Graph Theory Series. Pearson/Prentice Hall, 2007.

[127] A. S. Asratian, T. M. J. Denley, and R. Hggkvist, *Bipartite Graphs and their Applications*. Cambridge University Press, 1998, cambridge Books Online. [Online]. Available: http://dx.doi.org/10.1017/CBO9780511984068

[128] P. Twu, Y. Mostofi, and M. Egerstedt, "A measure of heterogeneity in multi-agent systems," in *American Control Conference (ACC)*, June 2014, pp. 3972–3977.

[129] T. Balch, "Hierarchic social entropy: An information theoretic measure of robot group diversity," *Autonomous Robots*, vol. 8, no. 3, pp. 209–238, 2000. [Online]. Available: http://dx.doi.org/10.1023/A%3A1008973424594

[130] D. A. Harrison and K. J. Klein, "What's the difference? diversity constructs as separation, variety, or disparity in organizations," *Academy of Management Review*, vol. 32, no. 4, pp. 1199–1228, 2007. [Online]. Available: http://amr.aom.org/content/32/4/1199.abstract