University of Modena and Reggio Emilia

DOCTORAL SCHOOL
INDUSTRIAL INNOVATION ENGINEERING

XXXIV Period

Multi-Robot Systems Control and Human-Multi-Robot Interaction

Beatrice Capelli

PhD Course Coordinator:
Prof. Franco Zambonelli

Supervisor:
Prof. Lorenzo Sabattini

To my family

Abstract

Multi-robot systems are rapidly improving and thanks to technological innovations they will become part of our everyday life. This near future will need new control strategies for these systems to enable them to efficiently complete their tasks, and to interact with human operators. In this thesis we proposed new control strategies to take multi-robot systems one step forward towards their applications in real world and their common usage.

First of all, we introduced a novel strategy for connectivity maintenance that can efficiently mediate between the task of the system and the need of communication among the agents of the system. In fact, most of the time a multi-robot system needs communication among its agents in order to work properly, hence a proper controller should consider this aspect. The proposed methodology is flexible, both in terms of type of tasks that can be achieved and of communication network, which can change during the task, always ensuring connectivity.

One of the most useful features of multi-robot systems is their intrinsic distributed nature, which enables them with the ability of simultaneously monitor different areas, or measure different quantities of interest. This ability is usually used in coverage tasks, where the robots are deployed in a large area that could not be monitored, or measured, efficiently by a single robot or by human operators. Coverage is a well-known and well studied problem for multi-robot systems, but it is usually studied under mild assumptions. Hence, we proposed a distributed coverage control law that considers the limited sensing range of robots and does not rely on a communication network.

Another challenging aspect of deploying multi-robot systems in real world is to provide them stable control laws. In this aspect, we introduced a novel methodology that enables a multi-robot systems to achieve its task in a stable manner, minimally changing the primary task. This method allows us to implement control law that may lead to instability in a safe manner, and eventually to deal with delays and interaction with unknown environments.

Finally, the topic of human-multi-robot interaction is faced. In fact, in order to deploy a multitude of robots along with operators, we need to investigate how robots can exchange information with humans. We focused on the ability of a multi-robot system to communicate its intention to a human operator. This ability is defined as legibility. We studied if these systems can communicate in an implicit manner their objective, both in terms of spatial goal and coordination objective. Furthermore, we characterized how the different characteristic of control laws can affect legibility.

Sommario

I sistemi multi-robot stanno migliorando rapidamente e grazie alle innovazioni tecnologiche diventeranno parte integrante della vita di tutti i giorni. Questo futuro è ormai vicino ed ha bisogno di nuove strategie di controllo per permettere a questi sistemi di svolgere in maniera efficiente le attività a loro assegnate e di interagire con gli operatori. In questa tesi proponiamo nuove metodologie di controllo per portare i sistemi multi-robot un passo avanti verso il loro utilizzo in applicazioni nel mondo reale e affinché diventino di uso comune.

Prima di tutto abbiamo introdotto una nuova strategia per il mantenimento della connettività che può efficacemente bilanciare l'obiettivo del sistema con la necessità di avere comunicazione tra gli agenti del sistema stesso. Infatti, la maggior parte dei sistemi multi-robot ha bisogno di comunicazione tra gli agenti per poter funzionare correttamente, quindi un controllore adeguato deve tenere conto di questo aspetto. La metodologia proposta è flessibile sia in termini di tipologie di compiti che possono essere ottenuti, sia di rete di comunicazione, che può cambiare durante l'attività, ma sempre garantendo un sistema connesso.

Una delle caratteristiche più utili dei sistemi multi-robot è la loro natura intrinsecamente distribuita, che permette a questi sistemi di monitorare contemporaneamente aree diverse o di misurare molteplici quantità di interesse allo stesso tempo. Questa abilità è spesso impiegata nell'ambito di compiti di coverage, in cui i robot sono dispiegati in ampie aree che non potrebbero essere monitorate, o misurate, efficacemente da un robot singolo o da operatori. Il problema del coverage è ben noto e studiato nell'ambito dei sistemi multi-robot, ma è solitamente affrontato introducendo delle ipotesi importanti. Proponiamo quindi una legge di controllo distribuita che considera le capacità di percezione limitate dei robot e che non si affida ad una rete di comunicazione.

Un altro aspetto impegnativo da tenere in considerazione nel dispiegamento dei sistemi multi-robot nel mondo reale è il fatto di poter implementare leggi di controllo stabili. Da questo punto di vista, introduciamo una nuova metodologia che permette ad un sistema multi-robot di compiere il proprio obiettivo in una maniera stabile, andando ad intaccare il meno possibile l'obiettivo stesso. Questo metodo permette di implementare in maniera stabile anche una legge di controllo che potrebbe introdurre instabilità, ed eventualmente anche di poter far fronte alla presenza di ritardi e all'interazione con un ambiente sconosciuto.

Infine, è trattato l'argomento dell'interazione tra uomo e sistemi multi-robot. Infatti, per poter utilizzare una moltitudine di robot al fianco degli operatori è necessario investigare come i robot possono comunicare con l'uomo. Ci siamo focalizzati

sull'abilità dei sistemi multi-robot di comunicare il proprio intento ad un operatore. Questa abilità è definita come leggibilità. Abbiamo investigato se esiste la possibilità per questi sistemi di comunicare in maniera implicita il loro obiettivo, sia in termini di obiettivi spaziali, sia di obiettivi di coordinazione. Abbiamo inoltre caratterizzato come le diverse caratteristiche delle leggi di controllo possono influenzare la leggibilità.

Acknowledgement

I would like to thank first of all my family, for always believing in me and supporting me, I will always appreciate this and I will be forever grateful for all the advice and lessons you gave me.

I would like to thank my advisor, Prof. Lorenzo Sabattini, for the great help during these years, for the long talks, and the great support he always gave me. I also thank all the people from the ARS Control lab, the lab mates, the other professors, the students, and the researchers for the support and the help during this long journey.

I would like to thank this age of progress for enabling me to reach this goal, to travel, and to meet amazing people around the world. The journey has been complicated for the health situation, but even in this difficult period the opportunity of growth and research has not ended. I am very grateful for this journey that made me improve and gave me the opportunity to improve the research by a very small fraction.

Contents

1 Introduction **1**
1.1 Contributions and Outline 2

2 Connectivity Maintenance **7**
2.1 Introduction to Connectivity Maintenance 8
2.1.1 Literature Review 8
2.1.2 Background on Graph Theory 9
2.1.3 Introduction to Control Barrier Functions 11
2.2 Connectivity Maintenance: a Control Barrier Functions approach . . 13
2.2.1 System Definition and Problem statement 13
2.2.2 CBF for Connectivity Maintenance 15
2.2.3 Evaluation 19
2.2.4 Conclusions 21
2.3 Connectivity Maintenance in Presence of Delay 23
2.3.1 Literature Review 23
2.3.2 Problem Statement 23
2.3.3 Heuristic for CBF Implementation in the Presence of Delays . 24
2.3.4 Experimental Validation 25
2.3.5 Conclusions 31
2.4 Higher Algebraic Multiplicity of Algebraic Connectivity 32
2.4.1 Literature Review 32
2.4.2 Introduction to Nonsmooth Control Barrier Functions 32
2.4.3 Problem Statement 33
2.4.4 Optimization-based Controller for Connectivity Maintenance . 35
2.4.5 Experimental Validation 37
2.4.6 Conclusions 41

3 Coverage Control: Limited Sensing Range Issue **45**
3.1 Literature review 45
3.2 Problem Statement 47
3.3 Background on Coverage Control 48
3.4 Limited Voronoi 49
3.5 Proposed Control Strategy 51
3.6 Varying-Variance Control 54
3.7 Experimental Validation 55

3.7.1 Simulations . . . 56
3.7.2 Implementation on the Robots . . . 57
3.8 Conclusions . . . 60

4 Robust Stability: a Control Barrier Functions Approach 61
4.1 Introduction . . . 61
4.2 Passivity and Energy Tanks . . . 62
4.3 Control Barrier Function for Passivity Preservation . . . 64
4.4 Case studies . . . 68
4.4.1 Case Study 1: Time-Varying Stiffness . . . 68
4.4.2 Case Study 2: Multi-Robot Control . . . 69
4.5 Experimental Validation . . . 70
4.5.1 Case Study 1: Time-Varying Stiffness . . . 70
4.5.2 Case Study 2: Multi-Robot Control . . . 76
4.6 Conclusions . . . 78

5 Human-Multi-Robot Interaction: Legibility of Multi-Robot Systems 81
5.1 Introduction to Legibility . . . 82
5.1.1 Literature Review . . . 82
5.2 Definition of Legibility in Multi-Robot Systems . . . 85
5.3 Single Group Legibility . . . 86
5.3.1 Methods . . . 86
5.3.2 Results . . . 95
5.3.3 Conclusions . . . 99
5.4 Multiple Groups Legibility . . . 100
5.4.1 Methods . . . 100
5.4.2 Results . . . 104
5.4.3 Conclusions . . . 105
5.5 Coordination Behaviors and Legibility . . . 107
5.5.1 Multi-Robot Coordination Objectives . . . 107
5.5.2 Methods . . . 113
5.5.3 Results . . . 117
5.5.4 Conclusions . . . 124

6 Main Findings and Future Works 125
6.1 Main Findings . . . 125
6.2 Future Works . . . 126

Chapter 1

Introduction

A fleet of drones is taking care of a vineyard, they alleviate the work of the farmer and they decrease the use of pesticides and chemical products, being very precise in the actions they take. In a warehouse the automatic guided vehicles are managing the shipping, while a group of drones is updating the inventory, and in the same environment the workers are lightened by the repetitive tasks. In the hospital the cleaning and the sanitation is entrusted to a group of robots that share the same halls with patients and medical staff.

This may be a futuristic vision, but it is not so far away from the real potential of multi-robot system. In fact, in recent years robotic systems have become more affordable, reliable, effective, diversified, and overall more present in our everyday life. Recent reports [1, 2, 3, 4] state that robotics industry will exponentially increase its development and income in the coming years. According to estimates, it will become one of the trending sector of industry: *"According to GlobalData forecasts, it was worth $45bn in 2020, and by 2030, it will have grown at a compound annual growth rate (CAGR) of 29% to $568bn"* [5]. This increase will be especially driven by mobile robots, such as service robots, drones, and mobile robots for logistics. This trend will be even increased by the impact of the pandemic. In fact, the need of introducing automation in logistics, distribution, inspection, and disinfection is highly increased. Focusing on mobile robots, some of the future and present applications are: automatic materials handling, automatic inventory in warehouse, inspection for oil rigs, chemical plants, wind turbines, infrastructure, electric transmission lines, and detection of mines. This boost in innovation will lead to better robots in terms of autonomy, price, sensing, actuation, ans so on.

These technological advancements have allowed the growth of a particular branch: multi-robot systems [6, Chap. 53], which uses the concept that having more robots, possibly simpler and cheaper, is better than having a big and expensive robot. The main positive aspects of multi-robot systems are [7]:

- Robustness: due to their intrinsic redundant nature they are more fault tolerant than a single robot, which is instead a single point of failure;

- Cost: the possibility of using simpler robots, with bounded resources, allows these systems to be economically competitive. In fact, the possibility of de-

ploying multiple robots for the same task does not require to mount all the sensor or all the tools on one expensive robot;

- Effectiveness: multi-robot systems can outperform single robot's performance thanks to possibility of parallelizing work. In addition, they are suitable for those tasks that are distributed by nature, such as monitoring multiple events at the same time, or measuring quantities in different spots simultaneously. In addition, the possibility of deploying heterogeneous groups of robots introduce also a high flexibility into the system.

On the other side, the drawbacks are: a more complex system, communication requirements, as the intrinsic distributed nature of multi-robot systems imposes the presence of an infrastructure that allows the robots to communicate, coordinate, and ultimately to function together, and overall the need of new control strategies, which may start from the ones developed for single-robot system. In particular, mobile multi-robot systems are the ones that can perform tasks moving in an environment [8]. These systems differ from multi-agent systems, in which usually the robotic component is stationary, and from distributed artificial intelligence system, where the focus is more on software agents.

Multi-robot systems are used in multiple domains, such as: exploration [9, 10], monitoring [11, 12], patrolling [13, 14], warehousing [15, 16], search and rescue [17, 18], agriculture [19, 20], military tasks [21, 22], and health care [23, 24]. These are just the dawn of the possibilities of multi-robot systems, in fact with the improving of technology they will enter more and more in our everyday life. From this, the need of improving the present control strategies, or building new ones, and to overall prepare these kind of systems to be deployed in real world applications.

The detailed literature review of the topics addressed in the thesis is reported in the respective sections.

1.1 Contributions and Outline

As multi-robot systems need to overcome the threshold of the laboratories' doors, we must address some of the limitations that are now blocking their use in real world. The main challenges to address in order to deploy multi-robot systems to real world are:

- Communication and connectivity: the robots must be able to communicate among them to coordinate and to achieve the common objective. The communication must interfere as little as possible on the main task of the system, namely the connectivity constraint should not prevent the accomplishment of the task, for example over-constraining the movement of the robots;

- Consider less restrictive hypotheses in the control strategies: most of the common control strategies that are used in multi-robot systems are based on some important assumptions that can not always be met in real world applications. In order to perform some real tasks, the control strategies should be able to work even when some of these assumptions are no longer valid;

- Robustness: the real world applications involve a real environment, which is not necessarily known a priori from the multi-robot system and with which it may interact. Enabling the control laws of the system to be robust with respect to possible sources of instability is mandatory;

- Interaction with human operators: as multi-robot systems will become more and more frequent, their interaction with humans must be studied and investigated. In fact, the efficacy of this interaction may increase even more the development of this kind of systems.

This thesis aims at investigating all these aspects, understanding the actual state of the art and analyzing the possible gaps present, proposing an effective control strategies and validating it in simulations, or possibly in real experiments.

Throughout the sections we will use both the single integrator and the double integrator dynamics to model the single robot of the system. We will use alternatively the two models as they allow us to consider more effects, in the case of the double integrator, or to consider just the kinematics, in the case of the single integrator, as the focus may be on other aspects and considering an over complicated model would not be beneficial. It is worth noting that, although the single integrator dynamics is a very simplified model, it can still effectively be used to control real mobile robots. In fact, using a sufficiently good trajectory tracking controller, the single integrator model can be employed to generate velocity references for widely used mobile robotic platforms, such as wheeled mobile robots [25], and unmanned aerial vehicles [26].

The dissertation is organized as follows.

In Chapter 2 we present a novel connectivity maintenance method. Connectivity maintenance is a crucial aspect of the deployment of multi-robot systems in real world applications. In fact, in order to collaborate and to achieve a common objective the robots need to coordinate, communicate, take decisions, and so on. These actions may happen among the robots themselves and also with a central unit, which may include the presence of a human operator. Connectivity maintenance is usually addressed with two approaches: local and global connectivity. We will focus in the latter one, which allows the system to reconfigure the communication network as long as connectivity is ensured. The presented method takes into account the fact that excessively constraining the movement of the robots may lead to a worsening of the performance of the system. Hence, the method balances the performance of the system with the need of keeping the system connected. The control law will be adapted for some particular conditions, and it will be validated both in simulations and in real experiments. The study of this topic has produced the following publications:

- **Connectivity Maintenance: Global and Optimized approach through Control Barrier Functions**
  Beatrice Capelli, Lorenzo Sabattini
  IEEE International Conference on Robotics and Automation (ICRA), 2020.

- **Decentralized Connectivity Maintenance with Time Delays using Control Barrier Functions**

Beatrice Capelli, Hassan Fouad, Giovanni Beltrame, Lorenzo Sabattini
IEEE International Conference on Robotics and Automation (ICRA), 2021.

- **Network Connectivity Maintenance via Nonsmooth Control Barrier Functions**
Pio Ong, Beatrice Capelli, Lorenzo Sabattini, Jorge Cortés
IEEE International Conference on Decision and Control (CDC), 2021.

- **Nonsmooth Control Barrier Function Design of Continuous Constraints for Network Connectivity Maintenance**
Pio Ong, Beatrice Capelli, Lorenzo Sabattini, Jorge Cortés
submitted to SAGE International Journal of Robotics Research (IJRR).

In applications such as exploration, monitoring, and measuring the type of controller that is usually used is coverage control. This type of controller allows us to deal with different problems using the same strategy. In particular, in Chapter 3 we present a method to start closing the gap between the use of robots in controlled environment, e.g., research laboratories, and in real ones. The main issue we address is considering the limited sensing range of robots, which is a realistic limit of the actual technological development. Along with the theoretical explanation, we will report the validation of the presented method. The results of this research has been submitted in the following paper:

- **On Coverage Control for Limited Range Multi-Robot Systems**
Federico Pratissoli, Beatrice Capelli, Lorenzo Sabattini
submitted to IEEE International Conference on Robotics and Automation (ICRA), 2022.

Chapter 4 faces the problem of robust stability of multi-robot systems considering always the trade off between the constraints that must be respected and the overall performance of the system. Robust stability is an important aspect to consider in carrying multi-robot systems into real applications as it allows the system to safely interact with the environment, to deal with the presence of delay in the control loop, and generally to ensure stability. The problem will be faced from a theoretical point of view, and then it will be evaluated in simulations and experiments. The method is presented in a paper currently submitted:

- **Passivity and Control Barrier Functions: Optimizing the Use of Energy**
Beatrice Capelli, Cristian Secchi, Lorenzo Sabattini
IEEE Robotics and Automation Letters (RA-L), 2021.

Lastly, in Chapter 5 we address the aspect of enabling multi-robot systems to coexist with human operator. In particular, we studied the legibility of this type of robotic systems. Legibility is the ability of robots of communicating their intent to human users. This aspect was almost unstudied in terms of multi-robot systems. We think that the topic needed deepening in view of applications in which robots will share the environment with operators, which may not be expert users of the

field. From this the need of understanding and studying the communication abilities of multi-robot systems. We will report the user studies we conducted, together with their analysis, and the subsequent results. The research of this topic has carried to the following publications:

- **Communication Through Motion: Legibility of Multi-Robot Systems**
  Beatrice Capelli, Cristian Secchi, Lorenzo Sabattini
  IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS), 2019.

- **Human-Friendly Multi-Robot Systems: Legibility Analysis**
  Beatrice Capelli and Lorenzo Sabattini
  Human-Friendly Robotics (HFR), 2019.

- **Understanding Multi-Robot Systems: on the Concept of Legibility**
  Beatrice Capelli, Valeria Villani, Cristian Secchi, Lorenzo Sabattini
  IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019.

- **Towards the Legibility of Multi-Robot Systems**
  Beatrice Capelli, María Santos, and Lorenzo Sabattini
  submitted to IEEE Transaction on Robotics (T-RO).

Chapter 2

Connectivity Maintenance

Connectivity maintenance is a fundamental aspect to consider when deploying a multi-robot systems. In fact, the robots need to communicate in order to coordinate, share information, and overall complete their task. Communication is needed as the small and simple robots that constitute the system do not have full knowledge of the state of the system, hence, they need to acquire this information by means of communication with the other agents. Like this also an agent with limited capabilities can obtain information by the sum of all the small pieces of knowledge that the other robots have.

Mobile robots transfer information by means of three main methods:

1. *Networked systems*: the robots communicate explicitly through an active means, such as radio, wireless network, infrared, etc.;

2. *Stigmergy*: a robot achieves information through sensing the effects of the other robots on the environment, thus mimic the communication principle used by ants and other animals. A particular version of stigmergy is *virtual stigmergy*, where each robot stores its partial knowledge of the environment and shares it through the network, i.e., it mixes stigmergy with networked systems;

3. *Passive communication*: robots measure directly the actions of their teammates.

In this chapter we will mainly consider networked systems, which have the ability to communicate directly and explicitly, but have the drawbacks of depending on a network that may be noisy, with limited bandwidth, and on the existence of the network itself. Namely, the robots must be connected all together in order to create a networked system.

We will not consider the possibility of intermittent connectivity, even though some interesting works have been proposed. Intermittent connectivity allows the group of robots to temporarily disconnect in order to perform some autonomous tasks and then to reconnect. Usually these control strategies, i.e., see [27, 28], focus on planning in advance the motions of the robots in such a way to ensure the connection of the group after a given time. Another possibility is to introduce a

human operator whose task is to reconnect the robots at the correct position and within the fixed time [29]. However, in this chapter we will focus on defining a control law that is able to maintain continuously the connectivity of the multi-robot system. The proposed method will allow the system to be deployed in unknown areas, which is forbidden in the case of predetermined trajectory, and it does not require the introduction of a human operator.

The contributions of this chapter are:

- Section 2.2: we propose a novel method that can optimize the achievement of the system's task, and at the same time, to ensure connectivity maintenance. We report the mathematical proofs and the simulations carried out to verify the effectiveness of the proposed method;

- Section 2.3: we deployed a distributed implementation of the proposed method on real robots. During the experiments we encountered an issue with delay and we solved it proposing a heuristic approach;

- Section 2.4: we propose an alternative solution that can deal with the issue of higher multiplicity of algebraic connectivity. We report the simulations and experiments we did in order to verify the method.

2.1 Introduction to Connectivity Maintenance

2.1.1 Literature Review

Communication among the robots is crucial for most cooperative applications, such as search and rescue [17], patrolling [13, 14], or exploration [10, 9]. Hence, the controller for a multi-robot system should take into account communication, which is usually represented by a communication graph, and should be able to maintain connectivity, i.e., the possibility for the robots to exchange information among each other. However, in several applications of interest, the desired input, namely the input defined to achieve the desired objective, can lead to a disconnection of the group. For example, if we want to perform measurements with sensors and maximize the area we analyze, we should control the robots in such a way that they spread as much as possible all over the area. But, as they usually have a limited range of communication, we may cause the disconnection of the group.

In general, to maintain connectivity there are two approaches: local and global. The local approach consists in keeping the connection between two robots if it exists when the task begins, so if the group starts connected it will be connected during the overall task. The main disadvantage of this method is the low flexibility due to the inability to change the communication graph. However, it can be easily implemented in a decentralized manner, thanks to the fact that it only needs local information, and in literature there exist a lot of examples [30, 31, 32]. On the other side, global connectivity consists in keeping the group connected allowing the robots to disconnect, if such disconnection does not lead to a loss of communication

in the group. This method provides a higher flexibility, allowing the robots to rearrange their interconnections to accommodate external needs, such as the presence of obstacles that obstruct their motion, e.g., a narrow passage. In [33, 34, 35] a decentralized connectivity maintenance is built through a gradient descent approach, while a recent comparison between the two approaches can be found in [36].

In order to increase the flexibility and the efficiency of the multi-robot system, we aim at considering global connectivity preservation together with some desired control input, and combine them in such a way to optimize the task while maintaining communication constraints. Control Barrier Functions (CBFs) are a suitable method for solving this problem. In fact, they allow us to consider, at the same time, different objectives and constraints. CBFs are commonly used for safety critical applications, such as in the automotive field where they are used for adaptive cruise control [37] and line keeping [38]. Another field of application is human-robot interaction, where CBFs allow to respect some safety constraints and, at the same time, optimize the task [39]. In the multi-robot domain, CBFs are used in [40] to guarantee collision-free movement of a group of mobile robots. In [41] they are exploited to combine different aspects: safety, connectivity, coverage and energy management. Moreover, in [42] CBFs are used to optimize connectivity and coverage. In [41, 42] the approach towards connectivity is local because the CBFs focus on maintaining all the initial links between the robots.

In this chapter we propose to integrate global connectivity into a desired control, e.g., coverage, formation control, patrolling, by means of a CBF. This method allows us to optimize the desired behavior of the system and, at the same time, to maintain connectivity. The optimization process allows to avoid the tuning procedure, which is usually present when two or more objectives or constraints are considered in the same controller [43]. Like in other global approaches, the initial configuration should generate a connected graph. But, differently from others, e.g., [34], where the initial value of connectivity should be above a certain desired value, we are able to increase connectivity to the desired value, or above. This property is due to the fact that the chosen CBF is also a Control Lyapunov Function (CLF).

2.1.2 Background on Graph Theory

In this Section we will report briefly some notation and properties of graph theory, for further details refer to [44, 45, 46]. Graph theory is fundamental for multi-robot system control as it allows us to model the communication topology in a mathematical formulation that can be used to formulate control laws, to verify and maintain connectivity.

We denote by $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}_{\geq 0}$, and $\mathbb{R}_{>0}$ the set of natural, real, real non-negative, and real positive numbers, respectively. Given $x \in \mathbb{R}^n$, let $\|x\|$ be the Euclidean norm.

The symbol $\mathbb{L}$ will represent the set of locally Lipschitz functions. A continuous function $\Omega : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class $\mathcal{K}$ function if it is strictly increasing and $\Omega(0) = 0$. It is an extended class $\mathcal{K}$ if it is a class $\mathcal{K}$ function, and it is defined on the entire real line, e.g., $\Omega : \mathbb{R} \to \mathbb{R}$.

A *graph* (*undirected graph*) is a pair $\mathcal{G} = (V, E)$, where the set V is constituted by

N elements, which are called *vertices* or *nodes*, and E is a set of unorderded pairs of vertices, i.e., $E \subseteq V \times V$. Elements of E are called *edges*. As the graph is undirected, the edge $e_{i,j}$ exists if and only if $e_{j,i}$ exists. The edge $e_{i,j}$ is the edge between the i-th and the j-th element of V, i.e., $i, j \in V$ and $i \neq j$. If the graph is *directed* the set of edges E is a set of ordered pairs. In the present work we will always consider undirected graphs. A graph is *connected* if there exists a path between any two vertices. A *path* is an ordered sequence of vertices that are consecutively linked with edges of the graph. We define the set of neighbors of the i-th element as $\mathcal{N}_i = \{j \in V | e_{i,j} \in E\}$.

A *weighted* graph is a triplet $\mathcal{G} = (V, E, A)$, where to the definition of graph (V, E) we add the *(weighted) adjacency matrix* $A \in \mathbb{R}^{N \times N}$, which is a nonnegative matrix defined as:

$$A = \begin{cases} a_{i,j} > 0 & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}, \tag{2.1}$$

where $a_{i,j}$ is the edge weight of $e_{i,j}$. Then we define the *(weighted) degree matrix* $D \in \mathbb{R}^{N \times N}$, which is a diagonal matrix and each element of the diagonal is equal to $\psi_{i,i} = \sum_{j=1}^{N} a_{i,j}$. Hence, $D = diag\{\psi_{i,i}\}$. Combining the adjacency and the degree matrix, we obtain the *(weighted) Laplacian matrix* $L = D - A$, with $L \in \mathbb{R}^{N \times N}$. The Laplacian matrix has some important properties [47]:

- $L\mathbf{1} = 0$, which means that the first eigenvalue of the Laplacian is zero and the associated eigenvector is $\mathbf{1}$, namely the column vector of all ones;
- Considering the set of eigenvalues of the Laplacian matrix $(\lambda_1, \ldots, \lambda_N)$ we know that:
  - They can be ordered as: $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$;
  - The second eigenvalue λ_2 is the algebraic connectivity of the graph (or Fiedler eigenvalue [48]), and we have $\lambda_2 > 0$ if and only if $\mathcal{G}$ is connected;
  - The multiplicity of 0 as an eigenvalue of L represents the number of connected components of $\mathcal{G}$;
  - Considering a weighted graph, we know that λ_2 is a non-decreasing function of each edge weight.

A *proximity graph* is a as a graph $\mathcal{G}$ in which the vertex set V is a set of distinct point in the space and the edge set E depends on the relative positions of the points. A multi-robot system's communication network can be represented with a proximity graph where the vertex set contains the robots themselves and the edges set represents the communication links among the robots, i.e., an edge exists between two robots if they can communicate. The proximity graphs are state dependent, and consequently time dependent. We will hereafter omit the time dependence, when not strictly necessary, for ease of notation. The proximity graph function associates to a set of vertices the proximity graph itself. We consider a set of points (robots) $\mathcal{P} = \{p_1, \ldots, p_N\}$, where $p_i \in \mathbb{R}^d$, and d is the dimension in which the robots move. We denote, for $p \in \mathbb{R}^d$ and $r \in \mathbb{R}_{>0}$, the closed and open ball in $\mathbb{R}^d$ centered at p with

radius r with $\overline{B}(p,r) = \left\{q \in \mathbb{R}^d | \|q-p\| \leq r\right\}$ and $B(p,r) = \left\{q \in \mathbb{R}^d | \|q-p\| < r\right\}$, respectively. We report some interesting examples of proximity graphs:

1. the *r-disk graph* $\mathcal{P} \mapsto \mathcal{G}_{disk}(\mathcal{P}, r) = (\mathcal{P}, E_{\mathcal{G}_{disk}}(\mathcal{P}, r))$, where the edges are defined as:
$$E_{\mathcal{G}_{disk}}(\mathcal{P}, r) = \{(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \text{diag}(\mathcal{P} \times \mathcal{P}) \mid \|p_i - p_j\| \leq r\}$$

2. the *Delaunay graph* $\mathcal{P} \mapsto \mathcal{G}_D(\mathcal{P}) = (\mathcal{P}, E_{\mathcal{G}_D}(\mathcal{P}))$, where the edges are defined as:
$$E_{\mathcal{G}_D}(\mathcal{P}) = \{(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \text{diag}(\mathcal{P} \times \mathcal{P}) \mid V_i(\mathcal{P}) \cap V_j(\mathcal{P}) \neq \emptyset\}$$

3. the *r-limited Delaunay graph* $\mathcal{P} \mapsto \mathcal{G}_{LD}(\mathcal{P}, r) = (\mathcal{P}, E_{\mathcal{G}_{LD}}(\mathcal{P}, r))$, where edges $(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \text{diag}(\mathcal{P} \times \mathcal{P})$ are defined as:
$$(V_i(\mathcal{P}) \cap \overline{B}(p_i, r/2)) \cap (V_j(\mathcal{P}) \cap \overline{B}(p_j, r/2)) \neq \emptyset$$

In particular, the r-disk graph will be used for in Chapter 2 to represent the underlying communication network of a system in which the communication model can be approximated to the fact that the robots can communicate to their neighbors if these are within their sensing range r. The Delaunay and r-limited Delaunay will be used in Chapter 3 for the implementation of the coverage control with limited sensing capabilities.

In order to understand if we are able to compute a proximity graph from the information of another proximity graph we can use the definition of *spatially distributed proximity graphs*. A proximity graph $\mathcal{G}_1$ is *spatially distributed over* another proximity graph $\mathcal{G}_2$ if, for all $p \in \mathcal{P}$,

$$\mathcal{N}_{\mathcal{G}_{1,p}}(\mathcal{P}) = \mathcal{N}_{\mathcal{G}_{1,p}}(\mathcal{N}_{\mathcal{G}_{2,p}}(\mathcal{P})),$$

which means that each node of $\mathcal{G}_2$ can compute its set of neighbors of $\mathcal{G}_1$ just with the information of its neighbors in $\mathcal{G}_2$.

2.1.3 Introduction to Control Barrier Functions

In this section we introduce the concept of Control Barrier Functions (CBFs), which are an extension of Barrier Functions for systems that do not present a forward invariant set for the desired state of the system. For more details the reader is referred to [49, 50].

We consider an affine control system:

$$\dot{\chi} = f(\chi) + g(\chi)\mu \tag{2.2}$$

where $\chi \in \mathbb{R}^p$ represents the state of the system, $\mu \in U \subseteq \mathbb{R}^q$ is the control input, $f(\chi), g(\chi) \in \mathbb{L}$.

What CBFs aim to do is keeping, or eventually carrying, the system (2.2) in a certain closed set $\mathcal{C} \subset \mathbb{R}^p$ defined as:

$$\begin{aligned} \mathcal{C} &= \{\chi \in \mathbb{R}^p | h(\chi) \geq 0\} \\ \partial\mathcal{C} &= \{\chi \in \mathbb{R}^p | h(\chi) = 0\} \\ \text{Int}(\mathcal{C}) &= \{\chi \in \mathbb{R}^p | h(\chi) > 0\} \end{aligned} \tag{2.3}$$

where $h(\chi) : \mathbb{R}^p \to \mathbb{R}$ is a continuously differentiable function. At this point there are two different approaches to define the control strategy: Reciprocal Control Barrier Functions (RCBFs) and Zeroing Control Barrier Functions (ZCBFs). For RCBFs we must define a function $B(\chi)$ that tends to infinity while the state approaches the boundary of the set $\mathcal{C}$, while for ZCBFs the function $h(\chi)$ must tends to zero on the boundary. We decided to use ZCBFs because they do not present the drawbacks of a function that assumes unbounded values on the boundary and they allow also to consider some perturbations or errors in the system (defining the function $h(\chi)$ on a set $\mathcal{D}$ larger than $\mathcal{C}$, $\mathcal{C} \subseteq \mathcal{D} \subset \mathbb{R}^p$). In addition, under certain conditions, they can be straightforwardly transformed in Control Lyapunov Functions, because they are well-defined also outside of the set $\mathcal{C}$. In the following we will use the term CBF meaning ZCBF.

The CBF should keep the state of the system inside[1] the set $\mathcal{C}$, and this is guaranteed choosing the input μ in (2.2) in such a way that:

$$\frac{d}{dt}h(\chi) \geq -\alpha\left(h(\chi)\right) \tag{2.4}$$

where $\alpha(\cdot) : \mathbb{R}^p \to \mathbb{R}^p$ is an extended class $\mathcal{K}$ function[2] and $\alpha(\cdot) \in \mathbb{L}$. Using (2.2), we can rewrite (2.4) as an explicit function of μ, as follows:

$$\frac{d}{dt}h(\chi) = \frac{\partial h(\chi)}{\partial \chi}\dot{\chi} = L_f h(\chi) + L_g h(\chi)\mu \geq -\alpha\left(h(\chi)\right)$$

where L_f and L_g are the Lie derivatives of $h(\chi)$:

$$L_f h(\chi) = \frac{\partial h(\chi)}{\partial \chi} f(\chi),\ L_g h(\chi) = \frac{\partial h(\chi)}{\partial \chi} g(\chi).$$

Then, the function $h(\chi)$ is a CBF if the following properties are satisfied [49]:

Property 2.1.1. *$h(\chi)$ is continuously differentiable.*

Property 2.1.2. *$h(\chi)$ is of relative degree one, i.e., its first order time derivative depends explicitly on the control input.*

Property 2.1.3. *It is possible to find an extended class $\mathcal{K}$ function $\alpha\left(h(\chi)\right)$ such that:*

$$\sup_{\mu \in U} \left[L_f h(\chi) + L_g h(\chi)\mu + \alpha\left(h(\chi)\right)\right] \geq 0,\ \forall \chi \in \mathcal{D} \tag{2.5}$$

[1] Formally, to make the set $\mathcal{C}$ *forward invariant*, i.e., if the system starts in whatever state $\chi(0) \in \mathcal{C}$, then $\chi(t) \in \mathcal{C},\ \forall t > 0$.

[2] Typically, $\alpha(h)$ is chosen equal to a linear function $\alpha(h(\chi)) = kh(\chi)$ or to the cube $\alpha(h(\chi)) = kh(\chi)^3$, with $k \in \mathbb{R}_{>0}$.

In order to make the set $\mathcal{C}$ forward invariant, we can apply any controller $\mu(\chi) : \mathcal{D} \to U$, $\mu(\chi) \in \mathbb{L}$ such that $\mu(\chi) \in K_{cbf}(\chi)$. The set $K_{cbf}(\chi)$ is defined as:

$$K_{cbf}(\chi) = \{\mu \in U | L_f h(\chi) + L_g h(\chi)\mu + \alpha\left(h(\chi)\right) \geq 0\}$$

It is important to notice that the closed loop controller $\mu(\chi) \in K_{cbf}(\chi)$ does not ensure that the system will converge to the given set $\mathcal{C}$ if it starts outside of it.

For this purpose, we can introduce a Control Lyapunov Function (CLF), which is a continuously differentiable, positive definite function $\Phi(\chi) : \mathcal{W} \subset \mathbb{R}^p \to \mathbb{R}_0^+$ such that:

$$\inf_{\mu \in U} \left[L_f \Phi(\chi) + L_g \Phi(\chi)\mu\right] \leq -\zeta\left(\Phi(\chi)\right), \ \forall \chi \in \mathcal{W}$$

where ζ is a class $\mathcal{K}$ function. Hence, it is possible to find an input $\mu \in U$ that stabilizes a point $\chi* \in \mathbb{R}^p$, or a set. More details can be found in [51].

In general, if we do not consider limitation on the input, we usually have a desired controller $\mu_{des} \in \mathbb{R}^q$ and we want to find $\mu \in K_{cbf}(\chi)$ that is minimally modified with respect to μ_{des}. For this purpose, it is useful to solve the following Quadratic Program (QP):

$$\begin{aligned} \mu(\chi) = \underset{\mu \in \mathbb{R}^q}{\text{argmin}} \quad & \frac{1}{2}\|\mu - \mu_{des}\|^2 \\ \text{s.t.} \quad & L_f h(\chi) + L_g h(\chi)\mu \geq -\alpha\left(h(\chi)\right) \end{aligned} \tag{2.6}$$

The optimization problem (2.6) has a Lipschitz continuous solution μ^* if μ_{des} is Lipschitz continuous, as well as the functions $f(\chi)$ and $g(\chi)$ in (2.2). Furthermore, $h(\chi)$ should have a Lipschitz continuous gradient and verify the relative degree one condition, i.e., $L_g h(\chi) \neq 0$, for all χ inside the safe set (see [49, Thm. 3]).

2.2 Connectivity Maintenance: a Control Barrier Functions approach

In this section we propose a novel method for connectivity maintenance that can optimally consider the task of the multi robot system and to ensure connectivity, based on Control Barrier Functions. We provide a mathematical proof of the method, and we validate it in simulations, demonstrating also its versatility in different applications. The research reported in this section has been published in [52].

2.2.1 System Definition and Problem statement

We consider a system composed by N robots moving in a n-dimensional space, and we define the position of the i-th robot as $x_i \in \mathbb{R}^n$. Then we can stack all the positions in a vector, which will represent the state of the system: $x = \left[x_1^T, \ldots, x_N^T\right]^T \in \mathbb{R}^{nN}$. We consider single-integrator dynamics:

$$\dot{x} = u \tag{2.7}$$

where $u \in U \subseteq \mathbb{R}^{nN}$ is the control input of the system. Hence, (2.7) can be represented by the general model of an affine control system, reported in (2.2), with $f(x) = \mathbb{O} \in \mathbb{R}^{nN \times nN}$, $g(x) = \mathbb{I} \in \mathbb{R}^{nN \times nN}$, where $\mathbb{O}$ and $\mathbb{I}$ are, respectively, the null and the identity matrix.

As in [53], we define the edge weights of the graph $\mathcal{G}$ as:

$$a_{i,j} = \begin{cases} e^{\left(R^2 - d_{ij}^2\right)^2/\sigma} - 1 & \text{if } d_{i,j} \leq R \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

where $d_{i,j} = \|x_i - x_j\|$ is the Euclidean distance between the i-th and the j-th robot, $R > 0$ is the communication distance, and $\sigma \in \mathbb{R}_{>0}$ is a positive constant to set the edge weight $a_{i,j} \leq 1$, for normalization purpose[3]. The above definition of the weight respects the following constraints:

- $a_{i,j} \geq 0$ to have similar properties to the unweighted Laplacian [47], and $a_{i,j} = 0$ if $d_{i,j} > R$;
- continuously differentiable with respect to the distance between the robots;
- decreasing as the distance increments.

We will hereafter assume that the Laplacian matrix has simple eigenvalues, i.e., $\lambda_i \neq \lambda_{i+1}, \forall i \in [2, \ldots, N-1]$. We will deal with the issue of multiple eigenvalues in Section 2.4.

This assumption is almost always verified in the considered scenario, in which edge weights are defined, in (2.8), as a function of the time-varying inter-robot distance. In fact, as shown in [54], in the pathological case in which the Laplacian has multiple eigenvalues, a small perturbation (introduced, in our case, by the motion of the robots) in the edge weights is sufficient to recover the simple eigenvalues situation.

In this chapter, we aim at solving the following problem: a multi-robot system, starting from an initial configuration where the group is connected ($\lambda_2(x(0)) > 0$), remains connected, while being controlled to achieve some desired objective.

From a mathematical point of view, we want to minimize the difference between the desired control input $u_{des} \in U \subseteq \mathbb{R}^{nN}$ and the actual input $u \in U \subseteq \mathbb{R}^{nN}$, which should ensure the connectivity of the system, encoded by $h(x)$. This problem will be solved, exploiting the CBF method, by means of a Quadratic Program (QP) subject to constraints:

$$\begin{aligned} u(x) = \underset{u \in \mathbb{R}^{nN}}{\operatorname{argmin}} \quad & \frac{1}{2}\|u - u_{des}(x)\|^2 \\ \text{s.t.} \quad & L_f h(x) + L_g h(x) u \geq -\alpha\left(h(x)\right) \\ & u(x) \in \mathbb{L} \end{aligned} \tag{2.9}$$

[3]One possible choice for σ is $\frac{R^4}{log(2)}$.

2.2.2 CBF for Connectivity Maintenance

In order to solve the problem stated in Section 2.2.1, we propose the following candidate CBF:

$$h(x) = \lambda_2(x) - \varepsilon \tag{2.10}$$

where $\lambda_2(x)$ is the second eigenvalue of the Laplacian matrix and $\varepsilon \in \mathbb{R}_{>0}$ is introduced in order to keep connectivity above a desired value. In the following, we will omit the dependence of λ_2 from the state x, for ease of notation.

The CBF $h(x)$ given in (2.10) defines the set $\mathcal{C}$, described as in (2.3), in which we want to keep our system:

$$\mathcal{C} = \{x \in \mathbb{R}^{nN} | \lambda_2 \geq \varepsilon\}$$

In addition, the CBF (2.10) is defined on a set $\mathcal{D}$, with $\mathcal{C} \subset \mathcal{D}$:

$$\mathcal{D} = \{x \in \mathbb{R}^{nN} | \lambda_2 > 0\}$$

This means that, in the following, we assume that the system always starts in a connected configuration.

We will now demonstrate the existence of a solution to the QP problem (2.9), choosing $\alpha(h(x)) = \varphi \cdot h(x)$, with $\varphi \in \mathbb{R}_{>0}$. Without loss of generality, in the following we will consider $\varphi = 1$, for ease of notation. Then we obtain:

$$\frac{\partial \lambda_2}{\partial x} u + (\lambda_2 - \varepsilon) \geq 0 \tag{2.11}$$

We rename the partial derivative with β:

$$\beta = [\beta_1, \ldots, \beta_N] = \frac{\partial \lambda_2}{\partial x} = \left[\frac{\partial \lambda_2}{\partial x_1}, \ldots, \frac{\partial \lambda_2}{\partial x_N}\right] \tag{2.12}$$

As shown in [33], the i-th component of β can be written as:

$$\beta_i = \frac{\partial \lambda_2}{\partial x_i} = \sum_{j \in \mathcal{N}_i} \frac{\partial a_{i,j}}{\partial x_i} \left(v_2^i - v_2^j\right)^2 \tag{2.13}$$

where v_2^i and v_2^j are the i-th and j-th component of the eigenvector associated to λ_2, respectively.

We will now show that at least one component of β is different from zero. This result is instrumental for the main results, that will be subsequently presented. We make the following assumption:

Assumption 2.2.1. *A collision avoidance mechanism is implemented, that prevents robots from colliding among each other.*

Lemma 2.2.2. *Consider a multi-robot system with communication topology defined according to the edge weights given in* (2.8)*, and consider the definition of β given in* (2.12)*. Let the communication graph be connected. Then, a value $k \in [1, \ldots, N]$ always exists such that $\beta_k \neq 0$.*

Proof. For ease of notation the demonstration is carried on component-wise, considering $x_i \in \mathbb{R}$. All the results can be trivially extended to the multi-dimensional case.

To prove the statement we first divide the communication graph $\mathcal{G}$ in z subgraphs, where $z \geq 2$. The rule to divide the nodes in the subgraphs is their corresponding component inside the second eigenvector of the Laplacian matrix, namely v_2, which is the eigenvector associated to λ_2. As discussed in Section 2.1.2, the eigenvector associated to the null eigenvalue of the Laplacian matrix is the column vector of all ones, i.e., $\mathbf{1}$. Since the eigenvectors of a matrix are orthogonal, we know in particular that v_2 is orthogonal to $\mathbf{1}$, that is: $v_2 \neq \rho\mathbf{1}$, where $\rho \in \mathbb{R} \setminus \{0\}$. Then, defining $\gamma_1, \ldots, \gamma_N \in \mathbb{R}$ as the components of v_2 (e.g., $\gamma_1 = v_2^k$ and $\gamma_2 = v_2^j$), we can always find at least two values such that $\gamma_1 \neq \gamma_2$.

We divide the communication graph $\mathcal{G}$ in z subgraphs $\mathcal{G}_i$, where $i = 1, \ldots, z$. Let:

$$\begin{array}{llll} \mathcal{G}_1 = \{V_1, E_1\} & \ldots & \mathcal{G}_z = \{V_z, E_z\} \\ V_1 = \{\omega \in V | v_2^\omega = \gamma_1\} & \ldots & V_z = \{\omega \in V | v_2^\omega = \gamma_z\} \\ E_1 = \{e_{i,j} \in E | i, j \in V_1\} & \ldots & E_z = \{e_{i,j} \in E | i, j \in V_z\} \end{array}$$

Hence, $\mathcal{G} = \bigcup_{i=1}^{z} \mathcal{G}_i \cup \overline{\mathcal{G}}$, where $\overline{\mathcal{G}} = \{\overline{V}, \overline{E}\}$, $V = \bigcup_{i=1}^{z} V_i$ and $E = \bigcup_{i=1}^{z} E_i \cup \overline{E}$. $\overline{E}$ represents the set of edges that link the subgraphs, namely the edges whose adjacent vertices belong to different subgraphs (the set of these vertices is $\overline{V}$):

$$\overline{E} = \{e_{i,j} \in E | (i \in V_s \wedge j \in V_t) \vee (i \in V_t \wedge j \in V_s)\}$$

with $s, t \in Z$, $Z = [1, \ldots, z]$.

We define $i \in V$ an extreme node if:

$$x_i < x_j \vee x_i > x_j, \forall j \neq i \in V$$

We know that two nodes always exist that verify this definition for Assumption 2.2.1.

Consider now the subgraph $\overline{\mathcal{G}}$ and in particular one extreme node of it, namely the k-th vertex. For this vertex we demonstrate that $\beta_k \neq 0$. The term β_k consists of a summation of terms, defined as the multiplication of two terms: $(v_2^k - v_2^j)^2$, and $\frac{\partial a_{k,j}}{\partial x_k}$. Considering the first term, namely $(v_2^k - v_2^j)^2$, we know that it is different from zero for all $j \neq k \in \overline{V}$, for the definition of the subgraphs. The second term $\frac{\partial a_{k,j}}{\partial x_k}$ can be divided in two additional terms $\left(\frac{\partial a_{k,j}}{\partial x_k} = \frac{\partial a_{k,j}}{\partial d_{k,j}} \frac{\partial d_{k,j}}{\partial x_k}\right)$. For $j \neq k \in \overline{V} \wedge j \in \mathcal{N}_k$, we can analyze the additional terms as follows, considering the definition of $a_{k,j}$ in (2.8):

- the term $\frac{\partial a_{k,j}}{\partial d_{k,j}}$ is different from zero, since $j \in \mathcal{N}_k$;
- the term $\frac{\partial d_{k,j}}{\partial x_k}$ is different from zero if $d_{k,j}$ is different from zero, namely if the positions of robots i and j do not coincide. This is ensured according to Assumption 2.2.1.

If the k-th vertex has just one neighbor in $\overline{\mathcal{G}}$, then the proof is completed.

If instead it has multiple neighbors in $\overline{\mathcal{G}}$ we need to consider the summation of multiple terms. In particular, we know that, if the summation is composed by terms of equal sign, then the summation will be different from zero. Considering the elements that build each addend, we know that the sign is defined by the difference between the position of the two vertices. As x_k is an extreme vertex then it will have all these terms with the same sign, which ends the demonstration. □

If we can ensure that the optimization problem (2.9) admits always a solution, then we can guarantee that the multi-robot system stays connected during the desired task.

Theorem 2.2.3. *Given the CBF described in* (2.10)*, consider the QP problem described in* (2.9)*. Then, we can always find $u \in U$ that verifies the constraint* (2.11) *of the QP problem.*

Proof. Substituting β, defined in (2.12), in the constraint of the QP problem (2.11), we obtain $\beta u \geq \varepsilon - \lambda_2$. We note that, if we can find an input u such that:

$$\beta u \geq \varepsilon \tag{2.14}$$

then this input will also satisfy the main constraint (2.11) because, from the characteristics of the Laplacian matrix, we know that $\lambda_2 \geq 0$ (Section 2.1.2).
Inequality (2.14) can be rewritten in an extended version: $\beta_1 u_1 + \beta_2 u_2 + \cdots + \beta_n u_n \geq \varepsilon$. We focus on the k-th element:

$$\beta_k u_k \geq \varepsilon - \sum_{i=1, i\neq k}^{n} \beta_i u_i \tag{2.15}$$

Define $\sigma_k = \sum_{i=1, i\neq k}^{n} \beta_i u_i$ and rewrite (2.15) as:

$$\beta_k u_k \geq \varepsilon - \sigma_k \tag{2.16}$$

If we consider an unbounded input, namely $U = \mathbb{R}^m$, we can always find a component u_k such that:

$$u_k \geq \frac{\varepsilon - \sigma}{\beta_k} \text{ if } \beta_k > 0, \quad u_k \leq \frac{\varepsilon - \sigma}{\beta_k} \text{ if } \beta_k < 0 \tag{2.17}$$

The definition in (2.17) is not well-defined for $\beta_k = 0$. However, according to Lemma 2.2.2, it is always possible to find at least one value k for which $\beta_k \neq 0$.

Concluding, we can always find a value $k \in [1, \ldots, N]$ for which u_k verifies (2.16) and hence (2.11).

□

We will now show that the candidate CBF given in (2.10) is a valid CBF.

Lemma 2.2.4. *Function $h(x)$ introduced in* (2.10) *is a valid CBF.*

Proof. In order to prove the statement, we need to show that $h(x)$ in (2.10) satisfies Properties 2.1.1, 2.1.2 and 2.1.3 (Section 2.1.3).

Property 2.1.1 requires $h(x)$ to be continuously differentiable. Considering the definition given in (2.10), we can write

$$\frac{\partial h(x)}{\partial x} = \frac{\partial \lambda_2}{\partial x}$$

As shown in [55, Thm. 1], simple eigenvalues of a real symmetric matrix are infinite times differentiable with respect to the variation of the element of the matrix. Considering the Laplacian matrix defined in Section 2.1.2, such elements are the edge weights defined in (2.8), or their summation. The edge weights are by construction infinite times differentiable with respect to the state x of the system. Hence, Property 2.1.1 is verified.

Property 2.1.2 requires $h(x)$ to be of relative degree one. This can be easily seen computing the first order time derivative of $h(x)$ in (2.10), as follows:

$$\frac{d}{dt}h(x) = \frac{\partial \lambda_2}{\partial x}\frac{d}{dt}x = \frac{\partial \lambda_2}{\partial x}u$$

Property 2.1.3 can be demonstrated according to Theorem 2.2.3. □

The chosen CBF (2.10) can also ensure an increase of λ_2, till at least the threshold value ε, if the system starts connected. This means that, if the system starts inside $\mathcal{D}$, then the input $u \in U$ that results from the QP problem will carry the system inside the set $\mathcal{C}$.

Theorem 2.2.5. *Let $\Phi_{\mathcal{C}}(x) : \mathcal{D} \to \mathbb{R}$ be:*

$$\Phi_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ -h(x) & \text{if } x \in \mathcal{D} \setminus \mathcal{C} \end{cases} \tag{2.18}$$

Consider the CBF $h(x)$ defined in (2.10). Then, $\Phi_{\mathcal{C}}(x)$ is a CLF for the system.

Proof. The candidate CLF (2.18) is continuously differentiable and positive definite by construction. We should demonstrate that the orbital derivative $\dot{\Phi}_{\mathcal{C}}(x)$ is negative definite or semidefinite. We evaluate $\dot{\Phi}_{\mathcal{C}}(x)$:

$$\dot{\Phi}_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ -\dot{h}(x) & \text{if } x \in \mathcal{D} \setminus \mathcal{C} \end{cases}$$

We focus on the case $x \in \mathcal{D} \setminus \mathcal{C}$:

$$\dot{\Phi}_{\mathcal{C}}(x) = -\frac{\partial h(x)}{\partial x}\dot{x} = -\dot{h}(x) \leq \alpha\left(h(x)\right)$$

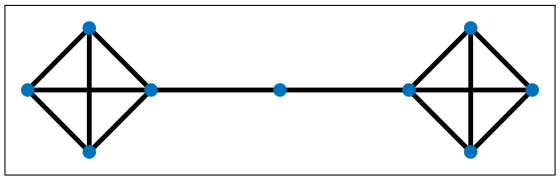
where $\alpha(\cdot)$ is a $\mathcal{K}$ function, which is guaranteed by (2.5).
Given the chosen CBF (2.10), we know that:

$$\alpha\left(h(x)\right) < 0 \text{ if } x \in \mathcal{D} \setminus \mathcal{C}$$

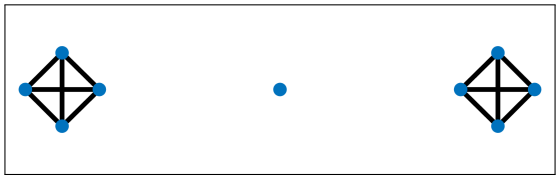
Combining the set of inequalities above, we get:

$$\dot{\Phi}_{\mathcal{C}}(x) < 0$$

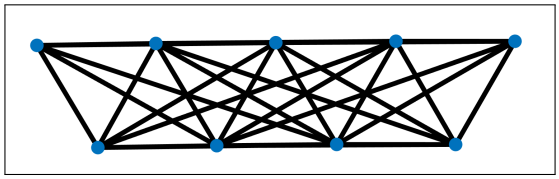
which ensures that the $\Phi_{\mathcal{C}}(x)$ is a CLF and asymptotically stabilizes the set $\mathcal{C}$. □



(a) $t = 0$s.



(b) $t = 0.5$s, $h(x)$ disabled.



(c) $t = 20$s, $h(x)$ enabled.

Figure 2.1: Some snapshots of the consensus simulation.

2.2.3 Evaluation

In this section we evaluate the proposed method considering a few different applications. In all the simulations, performed in MATLAB® , we imposed the threshold value $\varepsilon = 0.1$.

In order to avoid collisions among the robots, thus verifying Assumption 2.2.1, along the lines of [41], we included an additional CBF h_{safety}:

$$h_{safety}(x_i, x_j) = d_{i,j}^2 - d_{min}^2 \tag{2.19}$$

where $d_{min} \in \mathbb{R}_{>0}$ represents the distance between the robots that we consider safe, based on their the moving and sensing capabilities. This additional CBF can be considered together with $h(x)$, defined in (2.10), in a cumulative CBF:

$$h_{tot} = h \wedge h_{safety} = \min\{h, h_{safety}\} \tag{2.20}$$

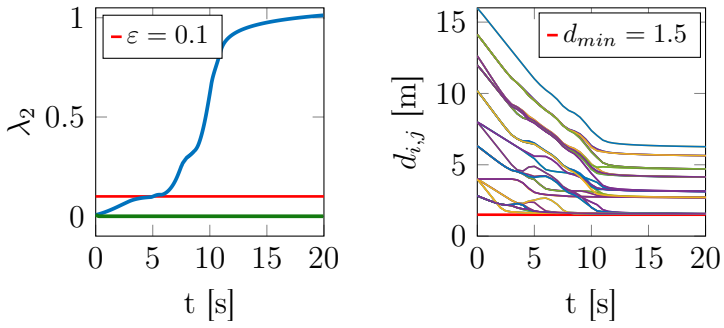
The reader is referred to [56, 41] for the composition of CBFs.

Consensus

The first example is based on consensus, which is a standard control strategy for multi-robot system. It is usually referred to as an aggregative behavior, but in [30] it has been found that, in some configurations (Fig. 2.1a), it can carry to a disconnection of the group. Fig. 2.2a reports the behavior of λ_2 with and without the CBF: it is easy to see how, without the constraint on the global connectivity, the consensus in this particular configuration leads to a disconnection of the group (Fig. 2.1b). Fig. 2.2b reports the distance between the robots and it demonstrates that the collision avoidance, together with the connectivity maintenance, can be achieved with the proposed method. The final configuration is reported in Fig. 2.1c.

Disconnecting Control Action

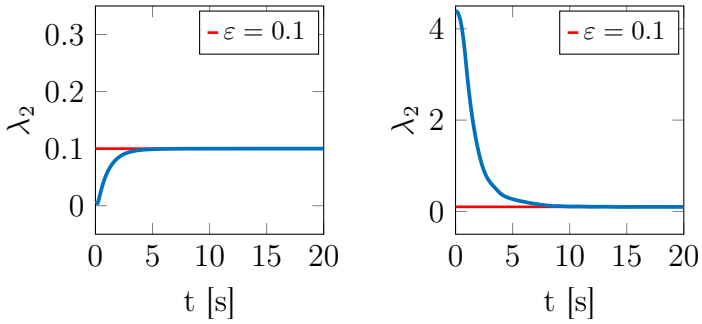
In this section we want to demonstrate that the proposed method can maintain connectivity against a desired controller that tends to break down the group. The



(a) Value of λ_2 for: $h(x)$ enabled, $h(x)$ disabled.

(b) Distance between the neighboring robots.

Figure 2.2: Graphs of the consensus simulation.



(a) $N = 4$, $\lambda_2(x(0)) < \varepsilon$

(b) $N = 40$, $\lambda_2(x(0)) > \varepsilon$.

Figure 2.3: Value of λ_2 in simulations with disconnecting u_{des} (2.21).

desired controller is defined for the i-th robot, where $i \in [1, \dots, N]$, as:

$$u_{des}^i = \begin{bmatrix} k\cos\left(\frac{2\pi}{N+1}i\right) & k\sin\left(\frac{2\pi}{N+1}i\right) \end{bmatrix}^T, \tag{2.21}$$

where $k \in \mathbb{R}_{>0}$ is a tuning parameter.

We report two main trials of this series of simulations. The first has 4 robots and the system starts with $0 < \lambda_2 < \varepsilon$, namely $x(0) \in \mathcal{D} \setminus \mathcal{C}$. Fig 2.3a reports the trend of λ_2 during the simulation and, as stated in Theorem 2.2.5, the CLF defined as (2.18) makes the set $\mathcal{C}$ asymptotically stable and the chosen CBF can actually increase the connectivity value up to the desired threshold.

In the second example, the system, composed of 40 robots, starts inside the set $\mathcal{C}$. Fig. 2.3b reports the behavior of λ_2.

Coverage

In many typical applications of multi-robot system, the main objective is searching or monitoring a certain quantity or data with ad-hoc sensors. Usually the area $\Upsilon \in \mathbb{R}^n$ in which the robots are deployed has some points of interest, e.g., a higher probability

to find victims in a search and rescue scenario or to measure the quantity that should be monitored. This different importance of the zones is usually described by means of a density function associated to the domain [57, 10], namely $\phi(\cdot) : \Upsilon \to \mathbb{R}_0^+$. These density functions can be defined for robot with different sensors [58], exploiting heterogeneity in multi-robot systems, or time-varying [59], to consider the evolution of the area during the task. However, spreading robots over an area can lead to loss of connectivity.

In this simulation, we solved the QP problem (2.9) using, as desired input, the one coming from a Voronoi-based coverage algorithm. In particular, we chose to implement the standard Lloyd approach [60, 57], which defines the input for the i-th robot as

$$u_{des}^i = -k\left(x_i - c_i\right),$$

where $k \in \mathbb{R}_{>0}$ is a tuning parameter, and c_i is the center of mass of the i-th Voronoi cell $\mathcal{V}_i$:

$$\mathcal{V}_i(x) = \{q \in \Upsilon | \|q - x_i\| \leq \|q - x_j\|, \forall j \in [1, \ldots, N]\}.$$

In addition, we report a comparison with a local connectivity approach, used in [41, 42]. The approach proposed in these works is based on finding a CBF h_{local} that defines the desired set as the set that maintains all the initial links:

$$h_{local}(x_i, x_j) = R^2 - d_{i,j}^2. \tag{2.22}$$

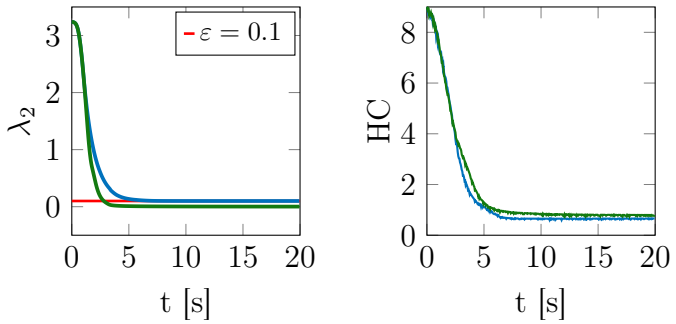
This CBF, as h_{safety} (2.19), is defined for each pair of connected robots and not for the whole system as, vice-versa, our $h(x)$ (2.10). The comparison is based on the locational cost [57], which is usually used to measure the efficiency of the coverage algorithms (lower value indicates better performance):

$$\mathcal{H}(x) = \sum_{i=1}^{N} \int_{\mathcal{V}_i(x)} \|q - x_i\|^2 \phi(q) dq. \tag{2.23}$$

We report, in the following, a significant example that highlights the differences between the proposed method and the existing one. The simulation was conducted with 4 robots, starting from a connected configuration above the given threshold ($x(0) \in \mathcal{C}$). Fig. 2.4a reports the behavior of the algebraic connectivity and, as might be expected, the proposed method keeps the value above the threshold, while the trial with h_{local} keeps the group connected, but with a lower value of λ_2. In addition, the proposed method allows to obtain a lower value of locational cost (2.23), which is compared to the one obtained with h_{local} in Fig. 2.4b.

2.2.4 Conclusions

In this section we have proposed a method to incorporate, in an optimal way, connectivity maintenance and a desired controller. The method is based on the definition of a Control Barrier Function that ensures the maintenance of the algebraic connectivity above a given value. The simulations show the feasibility and the flexibility



(a) Value of λ_2. (b) Value of $H(x)$.

Figure 2.4: Coverage simulation with comparison between: $h(x)$ and h_{local}.

of the method over different typical tasks. It is important to highlight that the proposed method was developed considering a centralized implementation. In fact, the calculation of the QP problem (2.9) and of all its components, i.e., λ_2 and its gradient, was carried out on a central unit. The central unit knew the position of all the agents and was able to communicate to every one its control input, obtained from the QP problem. This centralized implementation can be overcome using alternative methods to calculate algebraic connectivity, such as a decentralized estimation as presented in [34] or building progressively a full knowledge of the network on each robot as will be presented in Section 2.3. In addition, the QP problem (2.9) can be solved independently from each robot if it has the correct information, as will be shown in Section 2.3.

2.3 Connectivity Maintenance in Presence of Delay

In order to further validate the proposed method for connectivity maintenance we report in this section its implementation on a real robotics platform. During the validation we came across the problem of delay in the network, which can cause disconnection and other undesired effects. In order to counterbalance the presence of delay we propose a heuristic approach and we validate it both in simulations and real experiments. The research reported in this section has been published in [61].

2.3.1 Literature Review

When addressing a real deployment, one must take into account time delays. It is well known that delays usually lead to instability in controlled systems, if they are not properly considered [62]. Some recent works [63, 64] addressed the delay problem in the CBF approach. In particular, in [63] time delays were approximated and the proposed solution consisted in using the predicted value of the state for the CBF, instead of the current one. In [64] the time delays were not approximated and the existence of safety functionals was investigated, but the CBF was not explicitly derived due to the complexity of the problem.

2.3.2 Problem Statement

We consider the problem of implementing, in a fully distributed setting, the CBF-based control strategy proposed in Section 2.2 on a team of robots. In particular, we consider the following issues:

1. *Decentralized implementation*: each robot computes its control input independently, based on the data available on board;

2. *Imperfect state knowledge*: each robot can measure its own state, and can communicate it to the rest of the robots in the group. Each robot has then access to the full state of the multi-robot system, but the information is subject to communication delays.

It is worth noting that the simulation results reported in Section 2.2 were carried out in an ideal case, while the experiments in this Section consider a more realistic situation. Table 2.1 summarizes the main differences.

In particular, the QP problem (2.9) has been reformulated as:

$$\begin{aligned} u(x) = \underset{u\in\mathbb{R}^{nN}}{\operatorname{argmin}} \quad & \frac{1}{2}\|u - u_{des}(x)\|^2 \\ \text{s.t.} \quad & \frac{\partial \lambda_2}{\partial x} u \geq -(\lambda_2 - \varepsilon) \\ & u \in U \end{aligned} \tag{2.24}$$

in order to consider the limitations on the input of the real robots.

Table 2.1: Main differences between previous simulations, reported in Section 2.2.3, and the experiments of this section.

	Previous simulations Section 2.2.3	Experiments
Implementation	Centralized	Decentralized
Positions	Absolute	Relative
Kinematics	Omni-directional	Differential-drive
Input	$u \in \mathbb{R}^{nN}$	$u \in U$

2.3.3 Heuristic for CBF Implementation in the Presence of Delays

As reported in Section 2.3.4, we performed initial experiments to evaluate the behavior of the control strategy proposed in Section 2.2. When using a decentralized multi-robot setup, the presence of communication delays causes a degradation of the system performance.

Considering a constant delay $\tau \in \mathbb{R}^+$, the general model of the system given in (2.2) can be rewritten as

$$\dot{x}(t) = f\left(x(t), x(t-\tau)\right) + g(x(t), x(t-\tau))u(t) \tag{2.25}$$

to explicitly take into account the delayed state information used by each robot.

As reported in [64], the presence of such delay modifies the safety set of the CBF. However, the explicit computation of this modified safety set is challenging, and not practical in the general case. To avoid the need to explicitly compute the modified safety set, we propose a heuristic.

Each robot aggregates the state x from all other robots in the network through multi-hop communication, a process that introduces time delay due to communication latency. From the state, each robot can calculate the value of λ_2, which, in turn, is affected by the delay. More specifically, the computation taken at time t is based on the state $x\left(t-\tau\right)$. Hence, the computed value of λ_2 does not represent the *real* connectivity condition, which should use the state $x(t)$.

It is worth remarking that λ_2 is a non-decreasing function of each edge weight. Considering the definition given in (2.8), the value of each edge weight $a_{i,j}$ decreases, as the inter-robot distance $d_{i,j}$ increases. Hence, we propose a heuristic based on the worst-case scenario that takes place when, during the time τ, the increase of the distance $d_{i,j}$ is maximum. Namely, when robots i and j move in opposite directions at the maximum of their velocity. Fig. 2.5 shows a schematic representation of the idea behind the proposed heuristic.

Hence, the proposed heuristic consists of redefining the edge weights in (2.8), replacing the measured distance $d_{i,j}$ with a modified distance $\delta_{i,j}$, defined as:

$$\delta_{i,j} = d_{i,j} + 2v_{max}\left(\tau + \kappa\right), \tag{2.26}$$

where v_{max} is the maximum velocity of the robots. We also introduce a correction factor $\kappa \in \mathbb{R}^+$ to take into account additional disturbing factors that can affect the

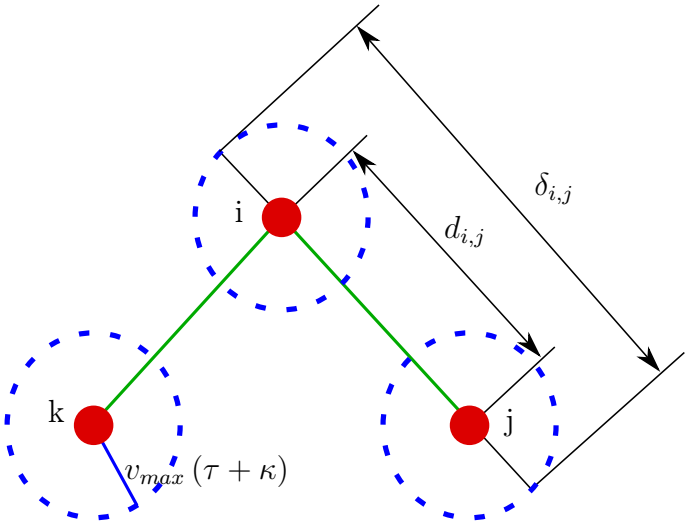


Figure 2.5: Schematic for the proposed heuristic. In *red* the robots, in *blue* the possible area in which each robot can be after the time τ, and in *green* the current edges.

computation of λ_2 from the communicated state. These disturbing factors include all the unmodelled elements that characterize the real system with respect to the ideal one, such as real robot kinematics with limited inputs, model errors, etc.

2.3.4 Experimental Validation

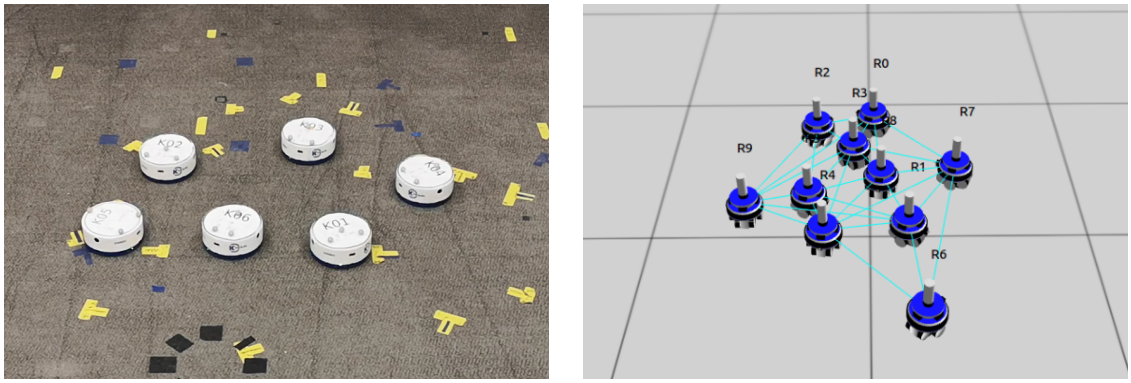
In this section we describe the experiments we carried out to demonstrate the effectiveness of the proposed control strategy, in the presence of time delays.

Hardware and Software Implementation

The experimental validation of the proposed connectivity maintenance strategy had been carried out on a group of K-Team Khepera IV (KH4)[4] robots deployed inside a 2×2 m arena equipped with an OptiTrack tracking system. The Khepera robot is equipped a with 800MHz ARM Cortex-A8 processor, mounting the Yocto operating system. The robots communicate through Wi-Fi and a software hub, `blabbermouth`[5] which emulates range and bearing sensors, namely each robot has the (simulated) ability to measure the distance and orientation between itself and its neighbors. In addition, `blabbermouth` can emulate limited communication range, packet drops, etc.

The control algorithm is implemented in Buzz [65], which is a programming language specific for swarm behaviors. In particular, during the experiments, we use the virtual stigmergy [66], a mechanism to share information through the multi-robot system. This mechanism allowed us to distribute the relative positions among all the robots. From this information, each robot could compute the value of λ_2

[4]https://www.k-team.com/khepera-iv
[5]https://github.com/MISTLab/blabbermouth



(a) Experimental setup. (b) ARGoS simulator.

Figure 2.6: Experimental validation.

and of the corresponding eigenvector of the Laplacian matrix (v_2). These values were used to define the QP problem (2.24) solved on board of each robot, with the `alglib` library[6].

For our simulations, we used ARGoS [67], a physics-based multi-robot simulator. ARGoS supports Buzz and allowed us to use the same scripts in simulation and on the real robots.

We did not use single-integrator dynamics (2.7) directly: the robots have differential-drive kinematics that need an additional transformation of the velocity inputs. We used input-output state feedback linearization [68] to transform the velocity input u into the suitable velocity commands for the robots, simulated and real.

We tested the CBF for connectivity maintenance with different desired behaviors. For both the simulations and the real implementation we set: $R = 1\text{m}$, $d_{min} = 0.25\text{m}$, and $u_{max} = 0.2\text{m/s}$.

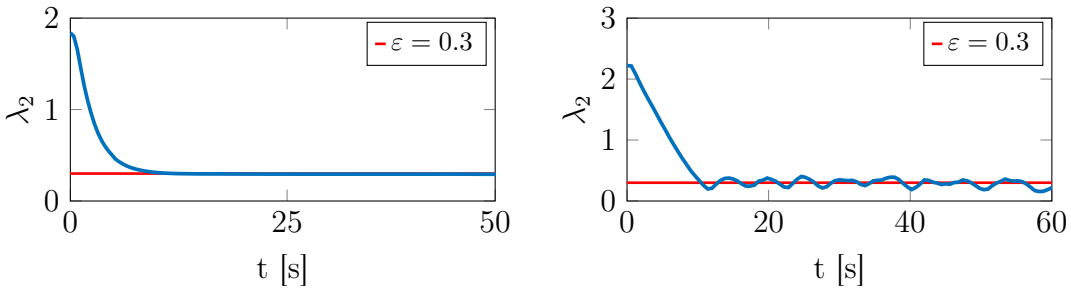
Initial Experiments

In the first experiments we implemented the control strategy proposed in Section 2.2 on the robots. The first desired behavior was the most challenging for the connectivity issue: the robots were controlled to disconnect (disconnecting behavior). Namely, the robots run away from each other, with a desired controller defined for the i-th robot, where $i \in [1, \dots, N]$, as in (2.21).

Fig. 2.7a illustrates the behavior of λ_2 in an ideal simulation, similar to the ones presented in Section 2.2.3.

This control would have led to disconnection, but, as can be seen from Fig. 2.7b, the proposed method reacts and prevents disconnections also in the real experiments. However, the threshold value is exceeded, namely $\lambda_2 < \varepsilon$. It is worth noting that the oscillating behavior is caused by the non-holonomic kinematics of the robots. To reduce this issue we can add an artificial damping to the dynamics of the robots.

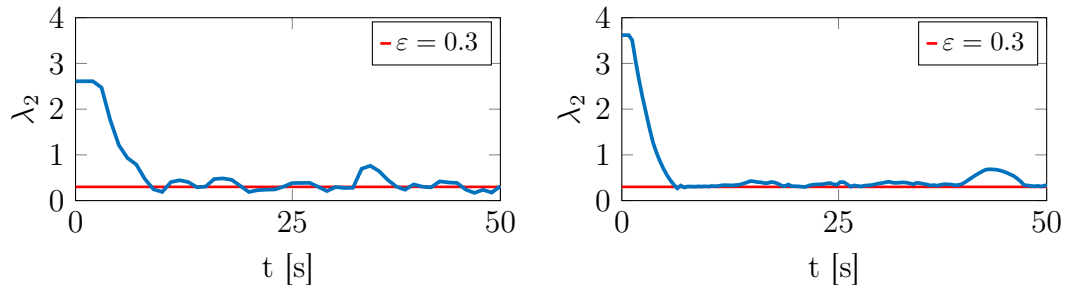
[6]https://www.alglib.net



(a) Ideal conditions, as in Section 2.2 ($N = 10$, $\varepsilon = 0.3$).

(b) Real conditions in initial experiments on real robots ($N = 3$, $\varepsilon = 0.3$).

Figure 2.7: Algebraic connectivity in ideal and real conditions with disconnecting behavior. In the real conditions, the threshold ε is crossed due to the presence of delay.



(a) With delay addition (delay = 0.1s, variable delay = 0).

(b) Without delay addition (delay = 0s, variable delay = 0).

Figure 2.8: Algebraic connectivity in the simulations for delay analysis with disconnecting behavior ($N = 10$, $\varepsilon = 0.3$).

Investigative Simulations

To understand which disturbing factor caused the decreased performance of the control law, we reproduced the same experimental conditions in simulation with ARGoS. In particular, we add a delay in the calculation of λ_2 and of v_2. Fig. 2.8a reports the trend of λ_2 in a simulation with the disconnecting behavior as the desired behavior for the robots, and the similarity is clear with the trend in the physical robots experiment (Fig, 2.7b). Instead, Fig. 2.8b reports the trends of λ_2 without the addition of the delay, and the behavior is approaching the ideal case reported in Fig. 2.7a. The small differences are probably caused by the different kinematics of the robots and the limited input.

From these simulations we deduce that the main issue is the delay, which corroborates the introduction of the heuristic proposed in Section 2.3.3.

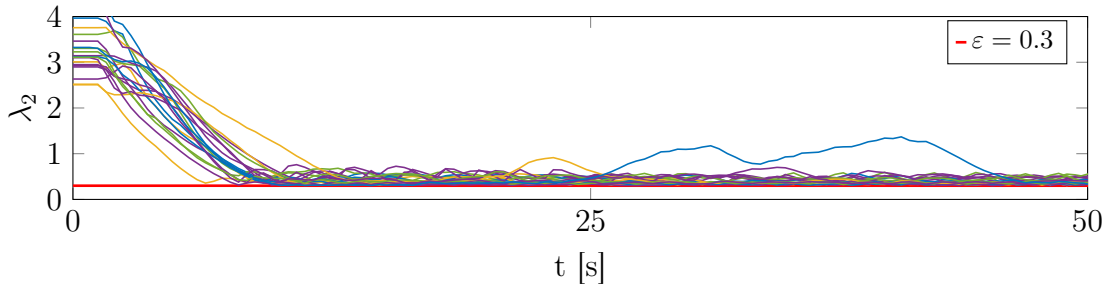


Figure 2.9: Algebraic connectivity in 20 simulations with the proposed heuristic and disconnecting behavior ($N = 10$, $\varepsilon = 0.3$, delay $= 0.05$ s, variable delay $= 1$).

Experiments with the Heuristic for Delay

In this section, we test the efficacy of the heuristic proposed in (2.26) for the disconnecting behavior.

Simulations

We investigated different combinations of variables: number of robots, threshold, and value of delay (possibly different for each robot). The diversification is aimed at verifying the effectiveness in a wide range of cases. For each combination of the variables we performed 20 experiments with random initial positions of the robots. Fig. 2.9 reports the behavior of λ_2 in every trial for one combination of the variables.

Table 2.2 summarizes the results of the simulations in an aggregate form. In particular, for each trial we recorded the minimum value of λ_2: the last three columns of Table 2.2 report the mean value (*Mean*), the standard deviation (*Std*), and the minimum value (*Min*), for each combination of the parameters, reported in the first four columns of the table. In particular, the third column (*Max delay*) was the maximum delay injected in the system. In addition, if the value in the fourth column (*Delay variable*) is equal to zero, then all the robots had the same delay, if the value is one, the robots worked with random delay, uniformly chosen between the maximum value and the null value (at least one robot had delay equal to the maximum value).

It is worth noting that in just few cases (highlighted in grey in Table 2.2) the minimum value of λ_2 fell below the given threshold, and this happened when using variable delays. However, even in these particular cases, the proposed heuristic compensates the effect introduced by the delay, and the minimum value of λ_2 is comparable to the one obtained in absence of delay (highlighted in light blue in Table 2.2).

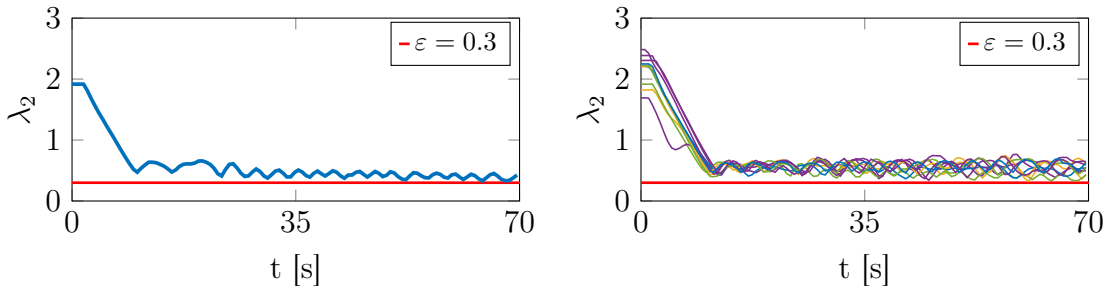
In the simulations without delay, the mean value is guaranteed to be above the threshold by the presence of the parameter κ, which allows to compensate the aforementioned other disturbing factors. During the simulations, the parameter κ was empirically tuned for the actual environment: we used the value $\kappa = 0.04$, which provided good results with all parameter combinations.

Table 2.2: Data of the simulations with the heuristic correction ($\kappa = 0.04$).

N	ε	Delay max [s]	Delay variable	Mean	Std	Min
5	0.1	0	0	0.104	0.003	0.099
5	0.1	0.1	0	0.177	0.023	0.134
5	0.1	0.1	1	0.126	0.027	0.081
5	0.1	0.05	0	0.126	0.013	0.109
5	0.1	0.05	1	0.123	0.079	0.082
5	0.3	0	0	0.310	0.008	0.295
5	0.3	0.1	0	0.458	0.046	0.400
5	0.3	0.1	1	0.370	0.083	0.265
5	0.3	0.05	0	0.359	0.025	0.315
5	0.3	0.05	1	0.321	0.026	0.277
10	0.1	0	0	0.103	0.005	0.088
10	0.1	0.1	0	0.213	0.030	0.149
10	0.1	0.1	1	0.129	0.041	0.075
10	0.1	0.05	0	0.136	0.015	0.114
10	0.1	0.05	1	0.105	0.016	0.078
10	0.3	0	0	0.317	0.009	0.299
10	0.3	0.1	0	0.603	0.059	0.492
10	0.3	0.1	1	0.433	0.100	0.272
10	0.3	0.05	0	0.340	0.031	0.343
10	0.3	0.05	1	0.326	0.042	0.262

Real Experiments

For testing the effectiveness of the proposed heuristic, we replicated the experiments of the disconnecting behavior in the real setup (described in Section 2.3.4). We empirically tuned κ at the value of 0.05, and we measured that $\tau \approx 0.3$ s. We performed several experiments with 3 robots starting in random initial positions. Fig. 2.10a shows the value of λ_2 for a representative trial, with starting positions of the robots similar to the ones of the experiment reported in Fig. 2.7b. With the introduction of the heuristic, the threshold value is never exceeded, differently from what happens without the correction for the presence of delay (Fig. 2.7b). To further validate the efficacy of the heuristic, we replicated the experiment with different starting positions. Fig. 2.10b reports the behavior of λ_2 in the ten different trials. The effect of the proposed heuristic on the calculation of λ_2 on the robots



(a) Initial positions similar to experiment reported in Fig. 2.7b.

(b) Random initial positions (10 replications).

Figure 2.10: Algebraic connectivity in the experiments on real robots with the proposed heuristic and disconnecting behavior ($N = 3$, $\varepsilon = 0.3$).

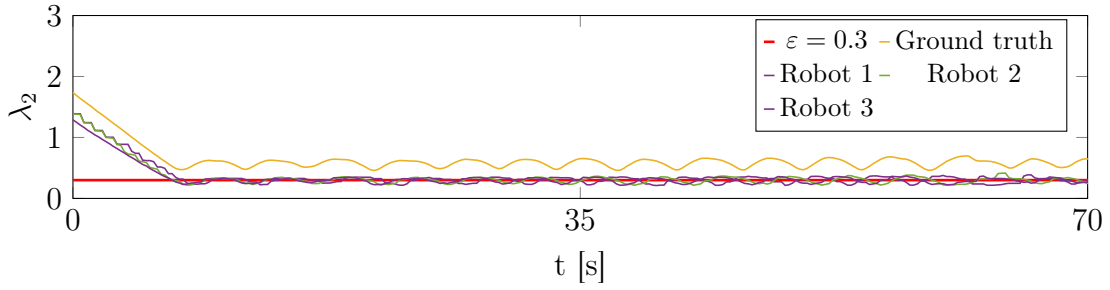


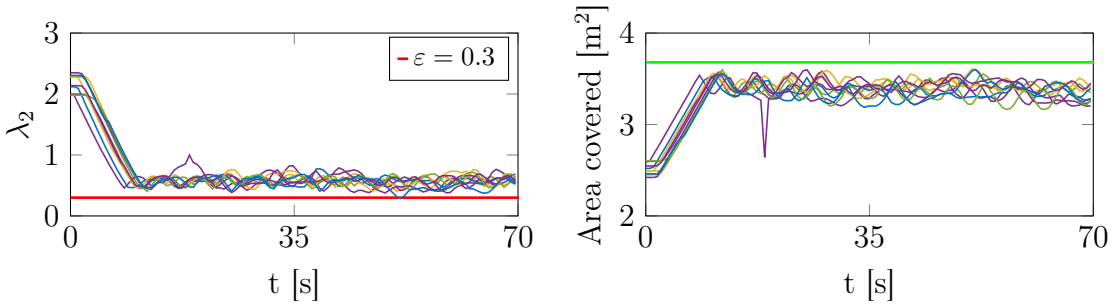
Figure 2.11: Difference between the actual algebraic connectivity and the values used by the robots. ($N = 3$, $\varepsilon = 0.3$).

is highlighted in Fig. 2.11. The actual value of λ_2 is calculated with the ground truth data, namely the data extrapolated from the Optitrack system, which report the exact distance among the robots. Instead, the value of λ_2 calculated on the robots is lower as a consequence of considering a worst-case distance. Finally, these experiments confirm the delay-compensation capacity of the proposed heuristic.

Performance Analysis

The second desired behavior is a coverage task [57], a standard problem of multi-robot systems, that we tested to investigate the effect on performance of choosing the worst-case scenario in (2.26). We performed a set of ten experiments, in which three Khepera IV robots used the available neighbours' positions to calculate a Voronoi tessellation using Fortune's algorithm [69], and then each robot carried out a Lloyd relaxation by chasing the center of its cell, in a manner similar to [57]. The area to be covered was 5.29 m^2 and the maximum covered area achievable was 3.68 m^2, with the given threshold $\varepsilon = 0.3$, and with a sensing range $R_{sensing} = 0.75$ m. We need to refer to a particular ε because a different choice of ε causes different performance of the system. However, the analysis of the influence of ε on the performances of the system goes beyond the scope of this section.

Hence, to analyse the impact of the proposed heuristic, we report, in Fig. 2.12b,



(a) Algebraic connectivity during the experiments.

(b) Area covered with respect to the maximum area achievable with the chosen ε (reported in *green*).

Figure 2.12: Coverage experiments with $N = 3$, $\varepsilon = 0.3$, $\kappa = 0.05$ (10 replications).

the area covered during the experiments with respect to the maximum achievable. It is clear how the maximum value is never reached, but the performance is not drastically decreased. Considering all the experiments, we obtain a mean deviation of 12.71% from the maximum area, with standard deviation equal to 3.03%. In addition, it is important to remark that, without the proposed heuristic, algebraic connectivity goes below the desired value, and the robots usually disconnect.

2.3.5 Conclusions

In this section we have presented a first implementation on real robots of the proposed Control Barrier Function for connectivity maintenance. The experiments show that the CBF is effective in a more complex, non-ideal, decentralized scenario, but with decreased performance. We found out that the main issue is the presence of delay in the system, in particular the fact that each robot works with a delayed state of the system. To solve this problem we have proposed a heuristic approach and we have demonstrated its effectiveness in simulation and on real robots.

The proposed heuristic was tuned for the system and for the experiments considered in Section 2.3.4, but it may give an idea on how to deal with the presence of delay in a system when using CBFs. As reported in [64], computing explicitly the safety domain in the presence of delays is challenging. Hence, we propose a heuristic approach that maintains comparable performance to the ideal case, as stated in Section 2.3.4. The proposed heuristic was developed explicitly considering the system detailed in Section 2.3.4, but the concept can be easily extended to more general scenarios, in which the control law is computed based on relative positions among the robots. In addition, the proposed heuristic does not increase the computation time with respect to the nominal case, since the modified inter-robot distance is trivially computed with constant factors.

2.4 Higher Algebraic Multiplicity of Algebraic Connectivity

In this section we improve the solution to the problem of keeping connectivity while optimizing the task of a multi-robot system. This method can be used also in case of multiple eigenvalues of the Laplacian matrix, therefore, we can get rid off of the hypothesis of simple eigenvalues that we have introduced in Section 2.2. In fact, when λ_2 has algebraic multiplicity higher than one its gradient is not well-defined. The research presented in this Section has been published in [70, 71].

2.4.1 Literature Review

A lot of works on global connectivity maintenance do not consider the issue of the eigenvalue having multiplicity higher than one. This issue is known to cause problems and requires special care in the literature. In particular, the issue has been addressed in the estimation of algebraic connectivity. For example, in [72] the proposed method for estimation cannot retrieve the multiplicity of the eigenvalues, and it is well known that the power iteration algorithm can not converge to any value if the algebraic connectivity has multiplicity higher than one [36]. To the best of our knowledge, in the context of feedback control for connectivity maintenance, [73] is the only work that has investigated this issue. In this section we improve the method proposed in Section 2.2 by leveraging Nonsmooth Control Barrier Functions (NCBFs) [56] to deal with the fact that the multiplicity of the eigenvalues might change with the network state. We use set-valued theory to study the continuity of the proposed controller.

The paper [74] studies the smoothness properties of optimization-based controllers with CBFs using perturbation theory. However, the result is limited to continuously differentiable CBF constraints, which is not applicable to our problem. Interpreting control constraint sets as set-valued maps, optimization-based controllers can be viewed as a parametric optimization problem, with the system state as the parameter. An important and well known result in this regard is [75, Thm. 17.31]. The work [76] studies the continuity of optimization-based controllers using set-valued theory, but the results are restricted to affine constraints.

2.4.2 Introduction to Nonsmooth Control Barrier Functions

As reported in Section 2.1.3 one of the main properties of a CBF is that it should be continuously differentiable. The CBF proposed in Section 2.2 is Lipschitz, but not continuously differentiable everywhere. In particular, when λ_2 has multiplicity higher than one the gradient (2.13) is not well-defined. Hence, to overcome this problem we can use Nonsmooth Control Barrier Functions (NCBFs) [56], which follow a similar concept of CBFs, but they allow us to deal with not continuously differentiable function. In fact, for a NCBF the sufficient condition is to be locally Lipschitz.

NCBFs are used to establish forward invariance of a desired set, as CBFs. Consider a set $\mathcal{C} = \{\chi \in \mathbb{R}^p \mid h(\chi) \geq 0\}$ with a locally Lipschitz continuous function $h : \mathbb{R}^p \to \mathbb{R}$, referred to as a Nonsmooth Control Barrier Function. We can ensure h remains positive during the evolution of the system (2.2) by limiting the choice of control μ, at each state χ, to the set

$$\mathcal{U}(\chi) = \left\{\mu \in \mathbb{R}^q \mid \min \mathcal{L}_f h(\chi) \geq -\alpha(h(\chi))\right\}$$

where α is a locally Lipschitz extended class $\mathcal{K}$ function, and

$$\mathcal{L}_f h(\chi) = \partial h(\chi)^T \cdot f(\chi, \mu),$$

is the weak set-valued Lie derivative [56, 77], $\partial h(\chi)$ is the generalized gradient of f, which is the extension of the gradient for functions that are locally Lipschitz, but not necessarily continuously differentiable. Given two matrices $A, B \in \mathbb{R}^{n\times n}$, the Frobenius inner product is defined as $A \cdot B = \sum_{i,j} A_{ij}B_{ij}$. It is worth noting that CBFs are a particular case of NCBFs, as $\frac{d}{dt}h(\chi(t)) \in \mathcal{L}_f h(\chi(t))$.

For $n \in \mathbb{N}$, we denote $[n] = \{1, \ldots, n\}$. Furthermore, let $\mathbb{S}^n = \left\{v \in \mathbb{R}^n \mid \|v\| = 1\right\}$ denote the set of normalized vectors in $\mathbb{R}^n$. Given a set $\mathcal{S}$, we denote with $\text{co}(\mathcal{S})$ its convex hull (or convex closure) and it represents the intersection of all the convex sets that contain the set $\mathcal{S}$.

2.4.3 Problem Statement

As λ_2 is only Lipschitz, but not continuously differentiable everywhere, there are some conditions where (2.27) is not well-defined. In fact, when the multiplicity of λ_2 is higher than one, it is no longer differentiable. Hence, we should find a way to deal with the conditions in which the communication network carries to such an increase of multiplicity.

We report an example of simulation in MATLAB® where λ_2 gains multiplicity higher than one. In this section k_{QP} is the solution of the QP problem:

$$\begin{aligned} k_{QP}(x) = \underset{u\in\mathbb{R}^{nN}}{\text{argmin}} \ & \|u - u_{des}(x)\|^2 \\ \text{s.t.} \quad & \frac{\partial \lambda_2}{\partial x}(x)u \geq -\alpha(\lambda_2(x) - \varepsilon), \end{aligned} \tag{2.27}$$

which is the equivalent of (2.9). We consider $N = 4$ robots moving in a two-dimensional space ($n = 2$), with a weighted r-disk proximity graph and the edge weights defined as in (2.8). Let α be the identity function, the threshold value $\varepsilon = 0.1$, and the communication radius $R = 3$m. The desired controller u_{des} constantly disperses the robots in different directions, as defined in (2.21).

Fig. 2.13 shows the trajectories of the robots during the simulation. The robot positions at $t = t^*$, when the multiplicity of λ_2 increases, are represented with asterisks. We note that these positions form a symmetric shape, and it is known that λ_2 increases its multiplicity when agents are in symmetric positions. Fig. 2.14 shows the evolution of λ_2 and λ_3 during the simulation, and it is clear that the two converge

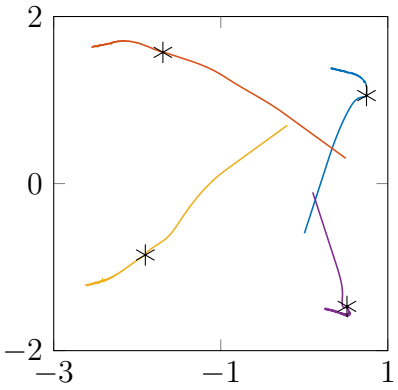


Figure 2.13: Trajectories of the robots during the simulation. The positions at $t = t^*$ are pointed out with asterisks.

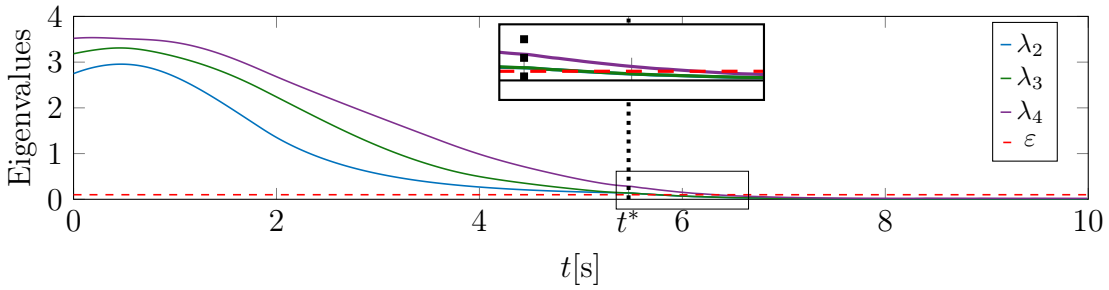


Figure 2.14: Eigenvalues ($\lambda_2, \lambda_3, \lambda_4$) during the simulation. The instant t^* when λ_2 and λ_3 converge to the same value is highlighted with the black dotted vertical line.

to the same value at $t = t^*$. Note that when this happens, the controller (2.27) is not well-defined and the following evolution depends on the implementation choice. Here, to continue the simulation, we use the formula (2.13), which assumes the eigenvalue always remains simple. Fig. 2.15 shows the applied inputs, in which we observe a chattering behavior. We attribute this to the combination of the digital implementation with the fact that the controller neglects higher eigenvalues (e.g., λ_3). By focusing on λ_2, the controller does not make any effort to limit how fast λ_3 may approach the threshold ε. As such, only when λ_3 becomes as small as λ_2, the controller jumps in value to prevent it from going lower. We suspect that the chattering is due to λ_2 and λ_3 interchanging their role. Fig. 2.14 shows that the controller (2.27) proposed in Section 2.2 allows λ_2 to reach zero and fails to maintain connectivity in this particular condition.

Remark 2.4.1. *It is worth noting that the constraint $u(x) \in \mathbb{L}$ (in Section 2.2) and the constraint $u(x) \in U$ (in Section 2.3) balanced the issue of jittering limiting the abrupt changes of the input if the condition of higher multiplicity was encountered. In addition, the assumption of having λ_2 simple is legitimate for most cases.*

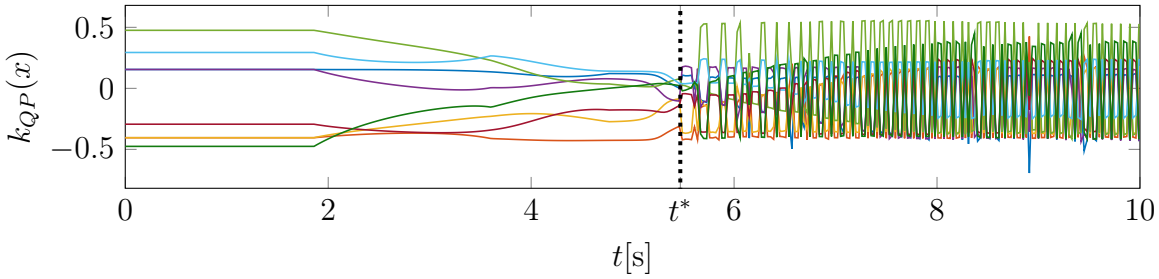


Figure 2.15: The applied inputs to the robots. When the multiplicity of λ_2 increases at t^*, the controller $k_{QP}(x)$ is no longer continuous and begins to chatter. We report all the components of $k_{QP}(x)$.

2.4.4 Optimization-based Controller for Connectivity Maintenance

In this section, we address the design of an optimization-based controller's constraint that accounts for the possibility of the algebraic connectivity not being simple. The idea is to consider all the possible rating changes of the eigenvalue and to constrain them, in such a way to obtain a controller that is able to face with the possible swap of eigenvalues, and consequently of gradient, when the multiplicity of λ_2 increases. In fact, as the constraint of the QP problem (2.27) depends on the gradient of λ_2 when this is not well-defined or when it changes abruptly we consequently have impact on k_{QP}.

First of all to deal with the condition in which the gradient of λ_2 is not well-defined, we need to introduce the generalized gradient of the eigenvalue λ_m, with $m \in [N]$, with respect to the matrix L (see e.g., [73]), which is given by:

$$\partial_L \lambda_m(L) = \mathrm{co}\{v_m v_m^\top \mid v_m \in \mathcal{V}_m\}, \tag{2.28}$$

where $\mathcal{V}_m = \{v_m \in \mathbb{S}^n \mid Lv_m = \lambda_m v_m\}$ is the set representing the set of normalized eigenvectors of the Laplacian matrix L associated to λ_m. Considering λ_m as a composite function, and using the nonsmooth chain rule [78, Thm. 2.3.10], we can find the expression for the weak set-valued Lie derivative [56, Rmk. 2.1] of λ_m:

$$\mathcal{L}\lambda_m(x,u) = \partial_L \lambda_m(L(x)) \cdot \Big(\sum_{r \in [Nn]} \frac{\partial L}{\partial x_r} u_r \Big).$$

We indicate with the subscript r the entries of the x and of the input u, instead of the index of the robot. The set-valued Lie derivative represents the possible values the eigenvalue's rate of change can take, i.e., $\frac{d\lambda_m}{dt} \in \mathcal{L}\lambda_m(x(t),u(t))$, along the trajectory. When the eigenvalue's multiplicity is larger than one, the set $\mathcal{L}\lambda_m(x)$ is no longer a singleton, i.e., it contains more than one element. This can explain the chattering behavior seen in Fig. 2.15 because the formula for controller (2.9) only uses one of the many possible rate of change, and therefore, it does not work with all possible values in $\mathcal{L}\lambda_m(x,u)$. To deal with all possible values, one possible solution is to work

with the minimum value of the set-valued Lie derivative $\mathcal{L}\lambda_m(x,u)$, leading to the NCBF constraint

$$\min \mathcal{L}\lambda_m(x,u) \geq -\alpha(\lambda_m(x)-\varepsilon).$$

This is a well-defined constraint on u for all x. By using the constraint for $m=2$, we can ensure the Fiedler eigenvalue λ_2 (and consequently all nontrivial eigenvalues) is above the threshold ε. We can define $\min \mathcal{L}\lambda_m(x,u) = \delta_m(x,u)$, which is defined as:

$$\delta_m(x,u) = \min_{v\in\mathcal{V}_m(x)} v^\top \big(\sum_{r\in[Nn]} \frac{\partial L}{\partial x_r} u_r \big) v. \tag{2.29}$$

Hence, we can propose the following NCBF:

$$\delta_2(x,k(x)) \geq -\alpha(\lambda_2(x)-\varepsilon). \tag{2.30}$$

The optimization problem (2.27) is updated as:

$$\begin{aligned} k_{dis}(x) = \underset{u\in\mathbb{R}^{nN}}{\operatorname{argmin}} \ & \|u-u_{des}(x)\|^2 \\ \text{s.t.} \quad & \delta_2(x,u) \geq -\alpha(\lambda_2(x)-\varepsilon). \end{aligned} \tag{2.31}$$

Where the constraint is well-defined, but it has the problem of being discontinuous every time the multiplicity of λ_2 increases. The consequence of this can be seen in the simulations reported in Section 2.4.5. In fact, the controller that results from solving the optimization problem is discontinuous and it can carry to the disconnection of the multi-robot system. This is due to the fact that we are still not considering the fact that the eigenspace corresponding to λ_2 changes dimension. Hence, we should take into account the eigenvectors of those eigenvalues that contribute to the increase of multiplicity.

For solving this problem, we can consider the worst case rate of change of λ_2, i.e., $\delta_{[n]}$, which considers all the possible direction of change of the eigenvalue, considering the span constituted by all the eigenvectors. We define

$$\mathcal{V}_{[m]}(x) = \operatorname{span}\{\cup_{2\leq p\leq m}\mathcal{V}_p\} \cap \mathbb{S}^n,$$

which is the collection of all normalized vectors spanned by the eigenvectors associated to eigenvalues smaller than or equal to λ_m. Then the worst case rate of change possible among all nontrivial eigenvalues up to λ_m results

$$\delta_{[m]}(x,u) = \min_{v\in\mathcal{V}_{[m]}(x)} v^\top \big(\sum_{r\in[nd]} \frac{\partial L}{\partial x_r} u_r \big) v. \tag{2.32}$$

Note that $\delta_{[m]}(x,u) \leq \delta_m(x,u)$. The special case $m=N$ allows us to consider an eigenspace that can not gain dimension as it is a constant set. Hence, we propose the following NCBF

$$\delta_{[N]}(x,u) \geq -\alpha(\lambda_2(x)-\varepsilon), \tag{2.33}$$

which results in the following optimization problem:

$$\begin{aligned} k_{str}(x) = \underset{u\in\mathbb{R}^{nN}}{\operatorname{argmin}} \ & \|u-u_{des}(x)\|^2 \\ \text{s.t.} \quad & \delta_{[N]}(x,u) \geq -\alpha(\lambda_2(x)-\varepsilon). \end{aligned} \tag{2.34}$$

which replaces δ_2, in the constraint of (2.31), with $\delta_{[N]}$. The constraint is continuous, but it is too conservative as it bounds the rate of change of λ_2 as it always has the highest possible multiplicity, i.e., $N-1$. This cause an over constrained motion of the robots when λ_2 is simple, as will be shown in Section 2.4.5.

To obtain a less conservative constraint we propose multiple NCBF constraints, one for each eigenvalue. The constraint for λ_m will consider the rate of change for all eigenvalues lower than or equal to λ_m, which can potentially affect the rate of change in the case of higher multiplicity. Finally, we propose the following optimization problem

$$\begin{aligned} k_{con}(x) = \underset{u\in\mathbb{R}^{nN}}{\operatorname{argmin}} \ & \|u - u_{des}(x)\|^2 \\ \text{s.t.} \quad & \delta_{[m]}(x,u) \geq -\alpha(\lambda_m(x) - \varepsilon), \ \forall m \geq 2. \end{aligned} \tag{2.35}$$

We can observe that $\delta_{[m]} \leq \delta_{[N]}$ and $\lambda_2 \leq \lambda_m$ for all $m \geq 2$. Consequently, (2.33) is stricter than each individual constraint here. In addition, the design also utilizes the CBF concept, and hence the eigenspace for λ_m only becomes relevant when the value of λ_m is close to ε.

2.4.5 Experimental Validation

In this section we report simulations and experiments we carried out to validate the proposed optimization problem. It is worth noting that the complexity of the constraints for calculating both $k_{str}(x)$ and $k_{con}(x)$ is too high for the common techniques used in the standard QP problem. In fact, each function $\delta_{[N]}(x,u)$ defining the constraints in (2.35) corresponds to an optimization problem that evaluates to a different value depending on the decision variable u, i.e., it is a nested optimization problem. In addition, these functions are non-smooth. Hence, the solution of the optimization problem was obtained using a saddle-point dynamics algorithm [79, 80]. The saddle-point algorithm involves performing a gradient descent in the primal variable (in this case u), and a gradient ascent in the dual variable that represents the constraints of the problem.

Simulations for Disconnecting Action

To verify the effectiveness of the proposed controller (2.35) we replicated the same simulation reported in Section 2.4.3. We used the same initial positions, the same parameters, and the same desired input u_{des}. Fig. 2.16 reports the behavior of the three nontrivial eigenvalues of L with the proposed controller. In particular, λ_2 never drops below the threshold ε, ensuring connectivity of the multi-robot system. Regarding the continuity of the controller, we refer to Fig. 2.17, which depicts the inputs applied to the robots. Comparing with Fig. 2.15, it is clear how the proposed method provides a controller that does not present jitter or abrupt jumps.

Simulations for Resource Gathering problem

We verified the proposed solution also with a different application, which is a gathering task in which a group of robots has to reach a set of given target as quickly

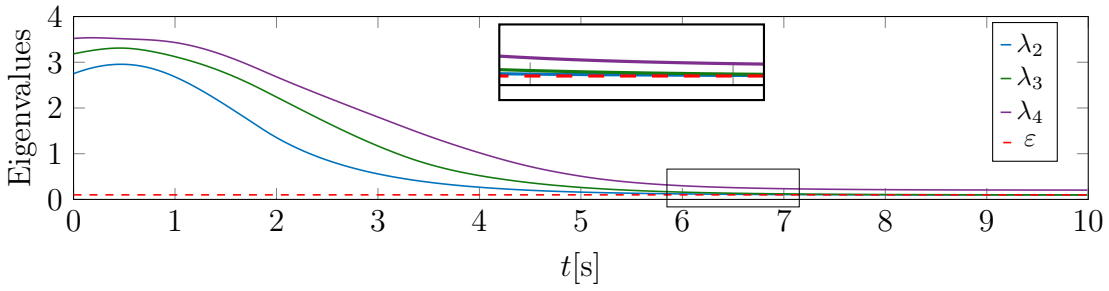


Figure 2.16: Eigenvalues (λ_2, λ_3, λ_4) with the proposed controller $u = k_{con}(x)$. All the eigenvalues are constrained above ε, and for λ_2 this means that the multi-robot system remains connected.

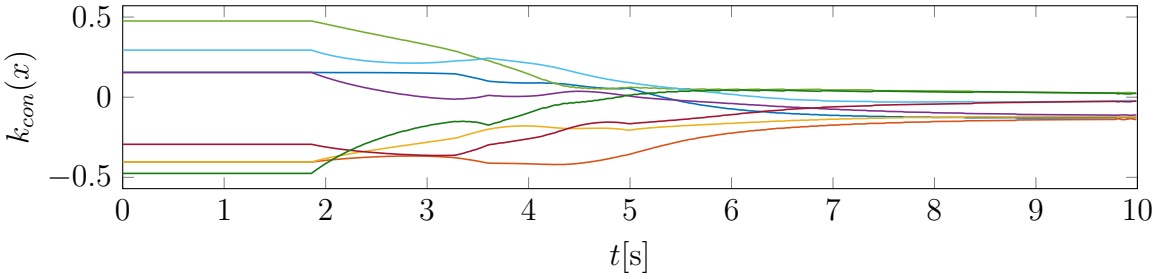


Figure 2.17: The continuous controller $k_{con}(x)$, solution of (2.35). It is clear that even in case of higher multiplicity the controller remains continuous. We report all the components of $k_{con}(x)$.

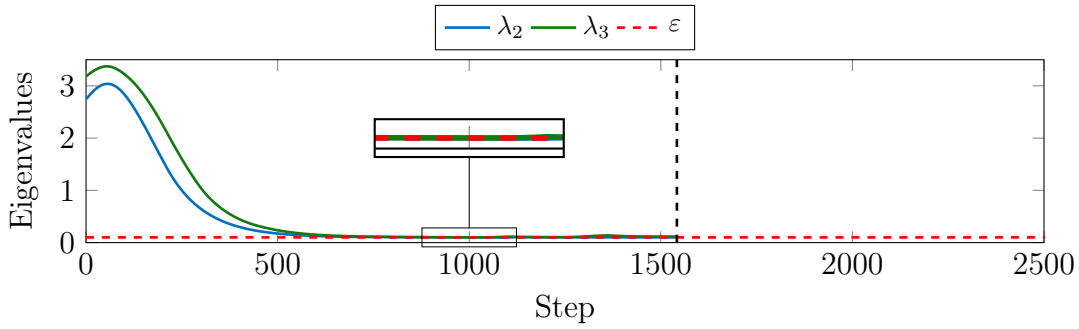
possible. We used four robots ($N = 4$), moving in a two-dimensional space ($n = 2$). Each robot had to reach the assigned target, and the targets had different priority, in such a way that the most important target was reached first. It is worth noting that the robots once reached the target may perform some other task, but this is outside the scope of this validation and we considered the mission accomplished if all the targets were reached by the corresponding robots. In a real application the robots may even gather resources at the target, or measure a given quantity.

The desired input carries the robot towards the corresponding target with a conical potential field:

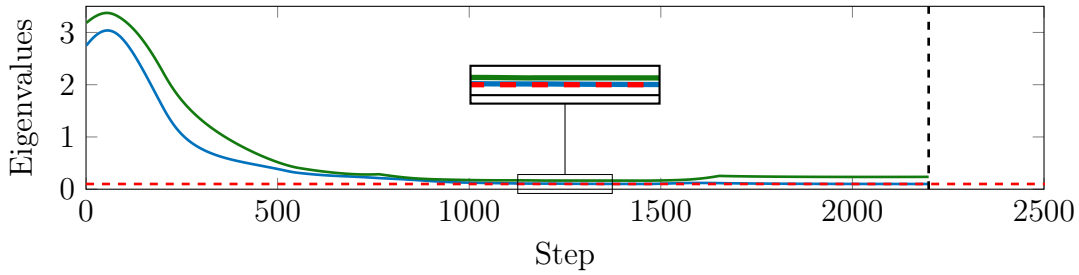
$$u_{des,i}(x_i) = k\frac{\mathbf{e}_i(x_i)}{\|\mathbf{e}_i(x_i)\|}, \tag{2.36}$$

where $\mathbf{e}_i(x_i) = x_{target,i} - x_i$ is the error between the target and the position of the robot, and $k \in \mathbb{R}_{>0}$ is a constant parameter. Once the robot gathers the assigned target, its $u_{des,i}$ is set equal to zero. This represents the fact that after having accomplished its task, the robot can collaborate at maintaining the connectivity constraint, relieving the constraint on the just achieved mission.

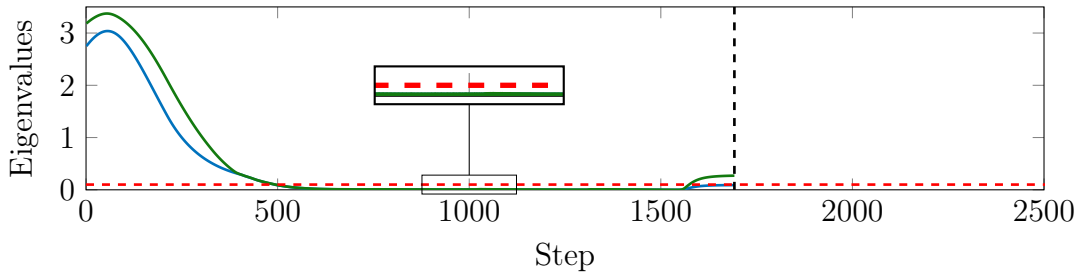
Our simulations highlight the differences among the different controllers: k_{dis}, defined in (2.31), k_{str}, defined in (2.34), and k_{con}, defined in (2.35). The initial positions and the robots' targets are the same in each simulation. Also the other



(a) Aggregate connectivity.



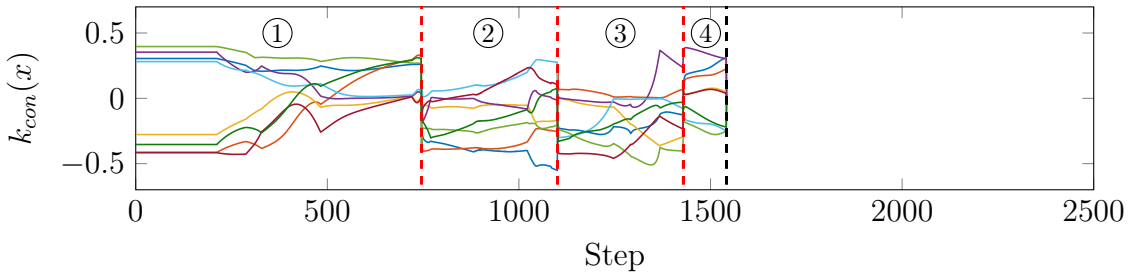
(b) Stricter connectivity.



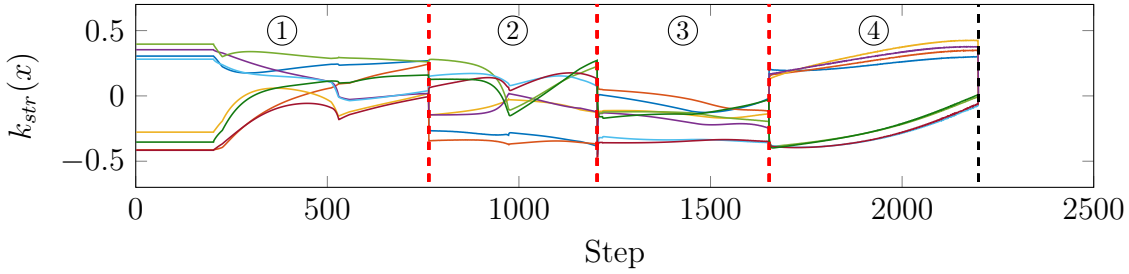
(c) Discontinuous connectivity.

Figure 2.18: Eigenvalues during the simulations with the different controllers. The *dashed black* lines represent the end of the task.

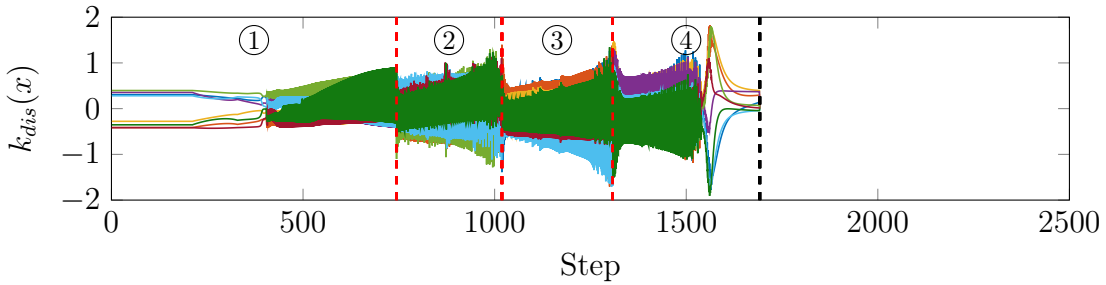
parameters are set constant in all simulations ($k = 0.5$, $\varepsilon = 0.1$). Fig. 2.18 reports the eigenvalues of the Laplacian matrix during the simulations. It is clear how both the continuous (Fig. 2.18a) and the stricter controller (Fig. 2.18b) can maintain the connectivity constraint, differently from the discontinuous one that carries to disconnection (Fig. 2.18c). Speaking of the overall time performance of the controllers, it is evident how the continuous controller outperforms the stricter one (aggregate: 1542 steps, stricter: 2199 steps). This corroborates the initial hypothesis on the fact that the stricter controller is over constraining the problem, hence resulting in a worse performance in terms of the total time it takes for the network to complete all of its goals. Instead, on the continuity of the control input, we can refer to Fig. 2.19, where it is clear how the continuous and the stricter controller produce a continuous input, and the discontinuous one reveals its discontinuous nature (Fig. 2.19c).



(a) Aggregate controller.



(b) Stricter controller.



(c) Discontinuous controller.

Figure 2.19: Control input for the different simulations. We report all the components of the control input for each robot.



Figure 2.20: Experiment setup: 4 ePucks and the corresponding targets.

Real Experiment

In the real experiment we carried out the same gathering task. We used four small wheeled robots (ePucks) that were controlled via Bluetooth from a central unit that performed the calculations. The central unit was also connected to an Optitrack system, which provided the position of the robots. In order to transform the input calculated for the single integrator dynamics to the unicycle dynamics of the robots, we used a simple input-output stated-feedback linearization [81]. We tested only the proposed controller k_{con}, as the simulations reported in the section above verified that it is the best both in terms of performance and connectivity maintenance. We set the main parameters as: $k = 0.1$, $\varepsilon = 0.3$.

Fig. 2.21 reports the eigenvalues of the performed experiment, which further confirms the effectiveness of the proposed method in maintaining connectivity. The accomplishment of the gathering task is testified by Fig. 2.23, which reports the trajectories followed by the robots. It is evident how each target had been reached by the corresponding robot, and in the final positions (reported with *triangles*) it is possible to see how the robots that have already reached their target cooperate to connectivity. The applied control input are reported in Fig. 2.22, where the jittering is due both to the non-idealities introduced while using wheeled robots, which hardly instantaneously follow an omnidirectional dynamics, and to the time needed for the calculation, which sometimes introduce a small delay. However, the jittering is far less than the one that we would have encountered applying k_{QP} or k_{dis}.

2.4.6 Conclusions

In this section we have presented a new solution to the problem of maintaining a multi-robot system connected while it is achieving some goals, and obtaining the best performance as possible. The proposed solution allows us to deal also with the situations in which λ_2 is not simple. This particular condition is not encountered frequently, but it carries to not well-defined problem for the solution proposed in Section 2.2. From this the need of finding a way to always obtain a well-defined

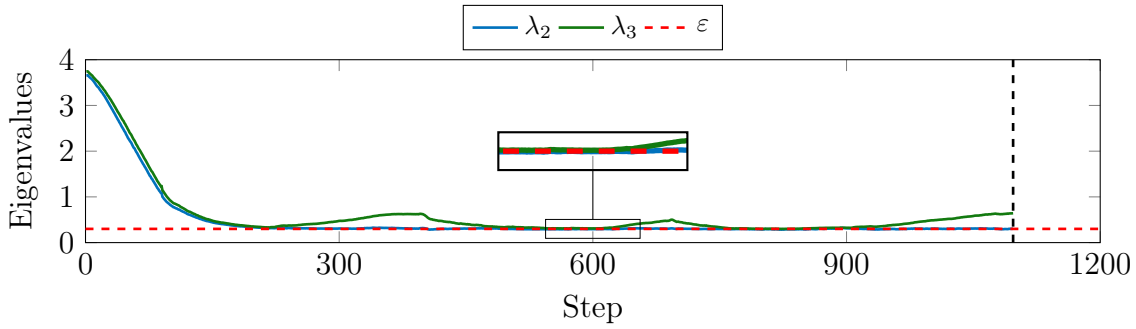


Figure 2.21: Eigenvalues during the experiment. The *dashed black* lines represent the end of the task.

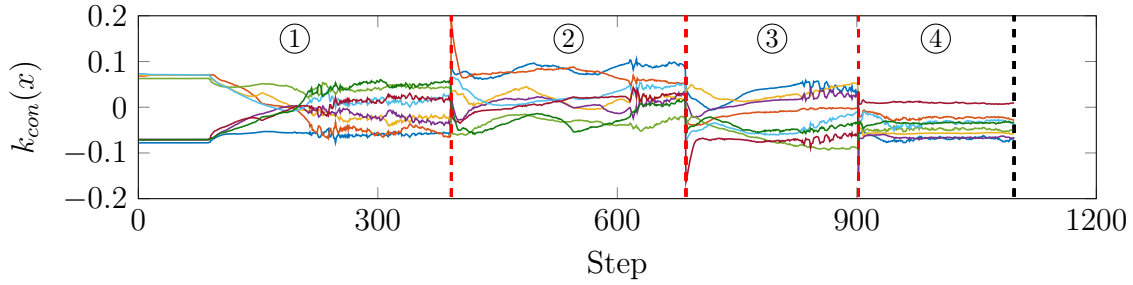


Figure 2.22: Control input applied to the robots in the experiment. We report all the components of the control input for each robot. These inputs have been transformed via input-output state-feedback linearization in order to be sent to the robots.

solution. In addition, we have taken care of the possible discontinuity that may arise from the gradient of λ_2, proposing two different NCBF. One of them is too conservative, while the other is the one that we have used in the experimental validation.

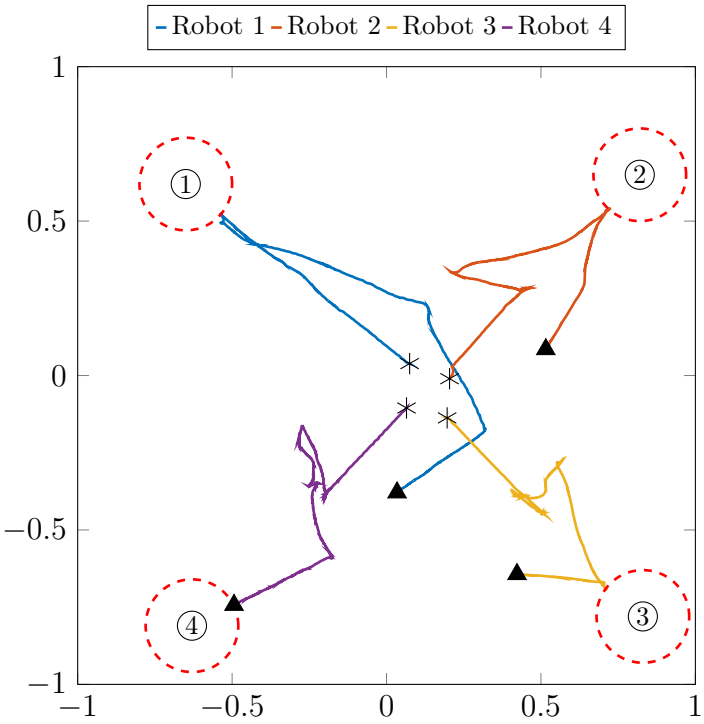


Figure 2.23: Trajectories followed by the robots during the experiment. The *dotted red* circles represent the area of tolerance in which we consider the target reached (circle of 15 cm of radius around the target). The numbers represent the order of priority of the targets. The initial and final positions are reported with *asterisks* and *triangles*, respectively.

Chapter 3

Coverage Control: Limited Sensing Range Issue

In this chapter we propose a coverage based control algorithm to coordinate a group of autonomous robots. Coverage control is generally used for deploying a group of robots in an area to detect or to measure some events or quantities, e.g., to measure pollutants, or to take photos of a perimeter to monitor. Most of the works available in literature are based on some assumptions that may prevent us to use this control strategy in real world applications. In fact, a usual assumption is that the robots have an unlimited sensing region and that they can measure all the area that is assigned to them, which is not always true, especially if the area to cover is big with respect the sensing range of the robots and their number. Alternatively, the robots are assumed to be able to communicate over a connected network. These assumptions allow the control strategy to use the *exact* Voronoi partitioning of the environment, which is the basis of most of the works present in literature. The *exact* Voronoi diagram can be computed by every robot with unlimited sensing capabilities or through a broadcasting of messages, if robots have limited sensing range. To overcome these limitations, we present a distributed control strategy to cover an unknown environment with a group of robots with limited sensing capabilities and in absence of reliable communication. The control law is based on a *limited* Voronoi partitioning of the sensing area. We propose a definition and a characterization of the *limited* Voronoi partitioning, and we demonstrate that the group of robots can optimally cover the environment using only directly detected information. Moreover, we introduce a novel methodology to deal with numerical issues in large scale experiments on hardware platforms when dealing with Gaussian-type probability density function.

The work presented in this chapter is currently under review [82].

3.1 Literature review

Coverage control has been widely studied in literature, and it is considered as one of the representative classes of control strategies for multi-robot system [83].

Various strategies have been introduced for implementing coverage control with

networked mobile robots. In particular, most relevant to this chapter are the results reported in [57, 84]. In these works, the authors presented decentralized coordination algorithms for groups of mobile agents based on Voronoi diagrams [85] and proximity graphs [44]. This approach gives a simple solution that guarantees the convergence of the networked robots to a configuration that maximizes the coverage of the environment. This strategy is based on a Voronoi partitioning of the whole environment, which is assumed to be known [86] or measurable by the robots [87]. For a successful implementation of this control law each robot should be able to compute the respective Voronoi diagram autonomously. In most of the works found in literature about the Coverage problem the Voronoi diagram is assumed to be given and is considered over the entire environment. In [87] the authors re-interpreted the problem in a non-probabilistic framework in which the robots can measure the value and the gradient of a sensory function. However, the authors assumed that a Voronoi tessellation of the environment is provided to the robots. All these works build upon the idea of *exact* Voronoi diagram, which we want to overcome with the method proposed in this chapter. There exist several algorithms to construct an *exact* Voronoi diagram by assuming that each robot can obtain the position of all the others [88] or by means of communication among sensors and broadcasting of messages [89, 90, 91].

However, in most practical scenarios, robots do not have information about the location of all other robots and can not rely on a communication network, hence, the Voronoi partitioning needs to be studied in a sensor range constrained scenario. Thus, instead of an exact Voronoi cell we consider a *limited* Voronoi cell, which is a partitioning of the area within the sensing range of the robot. A distributed algorithm was presented in [84, 57] to compute the Voronoi diagram for a limited sensing network, which is based on the algorithm presented in [92]. The exact Voronoi cell of each robot can be gradually refined by computing the partitioning within an adjustable sensing range and incrementally increasing the sensing range. The Adjust-Sensing-Radius algorithm has become the standard one in the field of distributed coverage control to solve the problem of the Voronoi partitioning. However, the objective of this algorithm is to determine the smallest distance R_i for agent i which provides enough information to compute the exact Voronoi cell V_i [57]. The algorithm requires sensors with controllable sensing range in order to determine the relative location of each Voronoi neighbor (more details in Section 3.3). In practical applications, the limited sensing capability of the robot is a constraint and rarely the sensing radius can be adjusted to detect sufficient information to compute the exact Voronoi diagram. An algorithm was proposed in [88] to efficiently compute the limited Voronoi cell. However, it is based on a reliable communication among the robots, which in real scenarios can not be always guaranteed (such as in underwater environments [93]). We consider a control strategy constrained by the information detected by the sensor (a communication network is not needed) and, thus, based on the Voronoi partitioning of the area within the sensing range (limited Voronoi diagram). An algorithm studied to perform efficiently this computation is described in [94].

Various studies focused on robots sensing capabilities have been proposed. In [58,

95, 96] the authors introduced agents heterogeneity into the standard coverage problem in order to consider the different sensing modalities. Differently, in [97] the authors focused on the anisotropic aspect of the limited sensing range. They proposed an alternative coverage problem based on a definition of a new proximity graph. It is worth noting that these works are based on an exact Voronoi partitioning of the environment: hence, the control algorithm requires the location of the Voronoi neighbors and the knowledge of the boundary of the environment to compute the Voronoi cell. The control strategy we propose can be applied to a group of robots deployed in an unknown environment, since the only information required by the control law is the one detected by the sensor. Moreover, only a few works can be found in literature in which this type of control is implemented and tested on real robots: an example is provided by [87], where the authors tested the proposed strategy on a swarm of mobile robots. Similarly, in this work we validate the proposed methodology in real experiments on ground robots.

3.2 Problem Statement

We consider a multi-robot system constituted by N robots that move in a two-dimensional space. This study can be extended to the 3-D case, but the computation of the Voronoi diagram will increase in complexity and different methods should be used. In addition, most of the coverage tasks require the deployment of the robots on a plane, which can also be at a given height, e.g., in the case of deployment of drones. Hence, the fact of considering a two-dimensional space does not reduce the applicability of the proposed method.

We assume that each robot can be modeled as a single integrator system, whose position $p_i \in \mathbb{R}^2$ evolves according to

$$\dot{p}_i = u_i, \tag{3.1}$$

where $u_i \in \mathbb{R}^2$ is the control input, $\forall i = 1, \dots, N$. The set of robots is represented by $\mathcal{P} = \{p_1, \dots, p_N\}$. We consider an environment to be covered by the multi-robot system, denoted by the convex polytope Q, which is not known a priori. The robots have the following characteristics:

1. *Limited sensing capabilities*: each robot is able to acquire its neighbors' positions, objects and to detect the boundaries of the environment Q, within its limited sensing range (defined by a ball with radius $R \in \mathbb{R}_{>0}$);

2. *Lack of communication network*: the only information used in the control strategy is the one directly detected by the robot inside its sensing range R. The only neighbors considered are the ones the robot can directly detect, i.e., neighbors on the r-disk graph $\mathcal{G}_{disk}(\mathcal{P}, R)$.

In this paper we provide a solution to the following problem:

Problem 3.2.1. *Define a distributed control strategy that allows a multi-robot system with limited sensing capabilities to optimally cover an unknown environment Q.*

3.3 Background on Coverage Control

We will now briefly summarize the *standard* solution to the coverage problem, as presented in [57]. We refer to Section 2.1.2 for the notation on graphs. Let $\mathbb{F}(\mathbb{R}^2)$ be the collection of finite point sets in $\mathbb{R}^2$. We can denote an element of $\mathbb{F}(\mathbb{R}^2)$ as $\mathcal{P} = \{p_1, \ldots, p_N\} \subset \mathbb{R}^2$, where $\{p_1, \ldots, p_N\}$ are points in $\mathbb{R}^2$. We denote, for $p \in \mathbb{R}^2$ and $r \in \mathbb{R}_{>0}$, the closed and open ball in $\mathbb{R}^2$ centered at p with radius r with $\overline{B}(p,r) = \{q \in \mathbb{R}^2 | \|q-p\| \leq r\}$ and $B(p,r) = \{q \in \mathbb{R}^2 | \|q-p\| < r\}$, respectively. Let 1_S represent the indicator function, which is defined as $1_S(q) = 1$, if $q \in S$, and $1_S(q) = 0$, if $q \notin S$. $Q \subset \mathbb{R}^2$ denotes a convex polygon: it will be used to denote the environment to be covered by the robots. An arbitrary point in Q is denoted by $q \in Q$.

The *Voronoi partitioning* can be defined on a polygonal environment in $\mathbb{R}^2$. Let $\mathcal{P}$ be a set of n points $\{p_1, \ldots, p_N\}$ in $\mathbb{R}^2$. The *exact* Voronoi partitioning generated by $\mathcal{P}$ consists in the set $\mathcal{V}(\mathcal{P}) = \{V_1(\mathcal{P}), \ldots, V_N(\mathcal{P})\}$, where:

$$V_i(\mathcal{P}) = \{q \in \mathbb{R}^2 | \|q-p_i\| \leq \|q-p_j\|, \forall p_j \in \mathcal{P}\} \tag{3.2}$$

In the following, for the sake of brevity, we will use the notation V_i to refer to $V_i(\mathcal{P})$. Two agents are said to be *Voronoi neighbors* if $V_i \cap V_j \neq \emptyset$. We refer to [98] for a major discussion about the Voronoi diagrams.

A performance function can be defined, to be maximized in order to obtain the optimal coverage of the group of robots over Q. The performance function is chosen to model how reliable is the measurement, at point $q \in Q$, of the sensor of robot i in proportion to the distance $\|q-p_i\|$. Therefore, the performance function can be defined as non-increasing differentiable function $f(\|q-p_i\|) : \mathbb{R}_{\geq 0} \to \mathbb{R}$.

Moreover, an integrable probability density function $\phi : Q \to \mathbb{R}_{\geq 0}$ can be defined in order to characterize the portions in Q where an event of interest occurs. Namely, ϕ is used to capture the relative importance of determined areas in the environment. For example, $\phi(q_1) > \phi(q_2)$ implies that q_1 is more important than q_2, or that the area around q_1 has a higher probability of being an area of interest. Such probability density function may be known a priori to all the robots, or may be estimated, at run-time, based on local measurements, as discussed in [87]: hence, without loss of generality, we will hereafter assume the probability density function to be known to all the robots.

Given an *exact* Voronoi partitioning $\mathcal{V}(P)$ of the environment Q in, so called, N Voronoi cells $\{V_1, \ldots, V_N\}$ (see Fig. 3.1a), the *optimization* function $H : Q \to \mathbb{R}$ can be formulated as follows:

$$\mathcal{H}(\mathcal{P}, \mathcal{V}) = \sum_{i=1}^{N} \int_{V_i} f(\|q-p_i\|)\phi(q)dq, \tag{3.3}$$

where each node is in charge of covering its own cell, and with a better coverage corresponding to a higher value of the function. In the literature, the performance function is usually chosen as: $f(x) = -x^2$. Therefore, the optimization function

becomes:

$$\mathcal{H}_{\mathcal{V}}(\mathcal{P}) = -\sum_{i=1}^{N} \int_{V_i} \|q - p_i\|^2 \phi(q) dq = -\sum_{i=1}^{N} J_{V_i,p_i}, \tag{3.4}$$

where $\mathcal{H}_{\mathcal{V}}(\mathcal{P}) = \mathcal{H}(\mathcal{P}, \mathcal{V})$ and J_{V_i,p_i} indicates the polar moment of inertia of the region V_i about the point p_i. In order to solve the optimization problem, the gradient of the optimization function can be computed, obtaining:

$$\frac{\partial \mathcal{H}_{\mathcal{V}}}{\partial p_i}(\mathcal{P}) = 2M_{V_i}(C_{V_i} - p_i), \tag{3.5}$$

where M_{V_i} and C_{V_i} denote respectively the mass and the center of mass with respect to the density function ϕ of the Voronoi cell $V_i \subset Q$ of the robot i in position p_i. Therefore, the mass and centroid can be computed as follows:

$$M_{V_i} = \int_{V_i} \phi(q) dq, \; C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} q\phi(q) dq. \tag{3.6}$$

For more details we refer the reader to [57, 87].

According to (3.5), the configuration of points $\mathcal{P}$ that maximizes the optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ coincides with the *centroids* of the respective Voronoi cells. In other words, the solution to the maximization problem is achieved when each agent is located at the centroid of its Voronoi region, such that $p_i = C_{V_i}, \; \forall i$. Such configurations are called *centroidal Voronoi configurations*, see [85]. At this point we can use $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ as a Lyapunov-like function, and we can design a control law that drives the group of robots to a centroidal Voronoi configuration, locally maximizing the coverage in the environment. In particular, the control input for each robot can be computed according to the Lloyd algorithm [57], as follows:

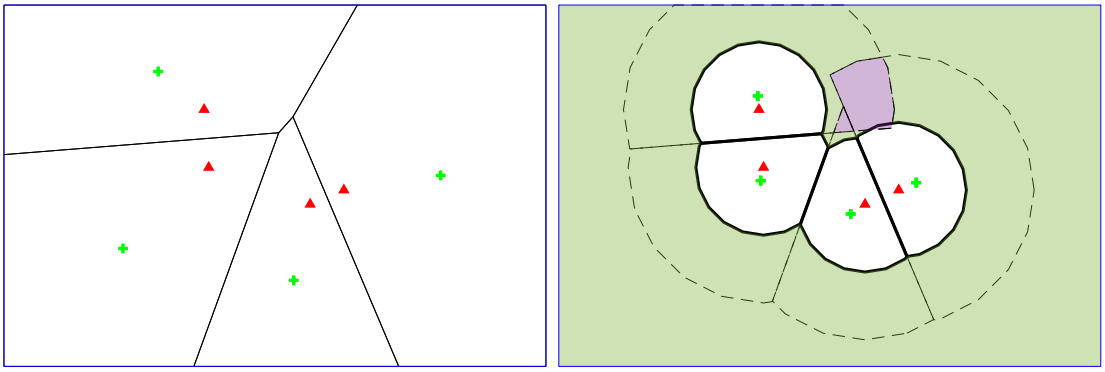
$$u_i = k(C_{V_i} - p_i) \tag{3.7}$$

where $k \in \mathbb{R}_{>0}$ is a proportional gain.

The exact Voronoi partitioning $\mathcal{V}(\mathcal{P})$ of the region Q is assumed to be continuously updated. Thus, it is assumed that the region Q is known or that the robot is able to measure it. The network of the group of robots is defined by the Delaunay proximity graph $\mathcal{G}_D(\mathcal{P})$ (see Section 2.1.2). To compute the i^{th} component of the $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ function, it is necessary for the robot i to know its own position and the positions of its neighbors over the graph $\mathcal{G}_D(\mathcal{P})$. Most of the works found in literature, which belong to the general category of coverage problem, are based on the presented methodology.

3.4 Limited Voronoi

In this section, we describe an alternative approach to obtain a Voronoi partitioning of the environment that considers the limited sensing range of the agents as a constraint, i.e., the *limited* Voronoi partitioning.



(a) *Exact* Voronoi $\mathcal{V}$. (b) *Limited* Voronoi $\mathcal{V}^r$.

Figure 3.1: Comparison between *exact* and *limited* Voronoi. The environment Q is reported in *blue*, the robots are the *red* triangles, and the centroid of the single Voronoi cell are the *green* crosses, the single Voronoi cell are delimited with *solid black* lines. Fig. 3.1b also reports in *dashed black* lines the wrong partitioning built considering the whole sensing range R, and in *purple* the overlapping areas.

We consider, for the coverage problem and the limited Voronoi partitioning, the ball with radius $r = \frac{R}{2}$, i.e., $\overline{B}(p_i, r)$, which is the half sensing range of robot i. Let $\mathcal{N}_{\mathcal{G}_{disk}(R)}(p_i, \mathcal{P})$ be the set of neighbors of the robot i with position p_i and let P_i be the set of points defined by $\{p_i \cup \mathcal{N}_{\mathcal{G}_{disk}(R)}(p_i, \mathcal{P})\}$. We can introduce the limited Voronoi partitioning generated by $\mathcal{P}$ as $\mathcal{V}^r(\mathcal{P}) = \{V_1^r(P_1), \ldots, V_N^r(P_N)\}$, where:

$$V_i^r(P_i) = \{q \in \overline{B}_{\cap Q}(p_i, r) \mid \|q - p_i\| \leq \|q - p_j\|, \forall p_j \in \mathcal{N}_{\mathcal{G}_{disk}(R)}(p_i, \mathcal{P})\}, \tag{3.8}$$

where $\overline{B}_{\cap Q}(p_i, r) = \{Q \cap \overline{B}(p_i, r)\}$ is the intersection between the environment and the half sensing range of robot i. Fig. 3.1 reports the difference between the exact Voronoi partitioning of Q, performed with the knowledge of the agents' positions, and the limited Voronoi partitioning we use in this paper, which is computed with the information directly sensed by the robot.

We can define the *r-limited Voronoi graph* $\mathcal{P} \mapsto \mathcal{G}_{LV}(\mathcal{P}, r) = (\mathcal{P}, \mathcal{E}_{\mathcal{G}_{LV}}(\mathcal{P}, r))$, where edges are defined as:

$$\mathcal{E}_{\mathcal{G}_{LV}}(\mathcal{P}, r) = \{(p_i, p_j) \in \mathcal{P} \times \mathcal{P} \backslash \mathrm{diag}(\mathcal{P} \times \mathcal{P}) \mid V_i^r(P_i) \cap V_j^r(P_j) \neq \emptyset\}.$$

Lemma 3.4.1. *The limited Voronoi partitioning defined in* (3.8) *is equivalent to the exact Voronoi partitioning limited to the portion of the environment Q that is covered by the union of the areas within the robots' half sensing ranges, i.e., $\cup_{i=1}^{N} \overline{B}_{\cap Q}(p_i, r)$*

Proof. The the r-limited Delaunay graph $\mathcal{G}_{LD}(R)$ is the proximity graph corresponding to the Voronoi partitioning limited to the robots' sensing ranges, namely

$$V_i(P) \cap \overline{B}(p_i, r).$$

According to [44, Prop. 2.9], $\mathcal{G}_{LD}(R)$ is spatially distributed over the graph $\mathcal{G}_{disk}(R)$. This implies that the r-disk graph encodes sufficient information to compute the r-limited Delaunay graph $\mathcal{G}_{LD}(R)$. Hence, it is possible to conclude that

$$V_i^r(P_i) = V_i(P) \cap \overline{B}(p_i, r). \tag{3.9}$$

which proves the statement. □

Remark 3.4.2. *Given Lemma 3.4.1, $\mathcal{G}_{LV}(r)$ is spatially distributed over $\mathcal{G}_{disk}(R)$.*

It is possible to show that the proposed definition of limited Voronoi diagram is well posed: namely, for a sufficiently large sensing range, or for a sufficiently large number of robots, the limited Voronoi diagram corresponds to the exact Voronoi diagram.

Lemma 3.4.3. *Consider the definition of limited Voronoi diagram given in* (3.8)*. There exist values $\overline{r}$ and $\overline{n}$ such that, for $r > \overline{r}$ or $n > \overline{n}$, then $\mathcal{V}^r(P) = \mathcal{V}(P)$.*

The intuition comes from the fact that, for any bounded environment Q, it is always possible to find $\overline{r}$ or $\overline{n}$ such that $Q \subseteq \cup_{i=1}^{N} \overline{B}(p_i, r)$.

Remark 3.4.4. *If we consider* (3.8) *with R instead of r, such that $q \in \overline{B}_{\cap Q}(p_i, R)$, the Voronoi partitioning generated is not a valid partitioning. In fact, $\mathcal{G}_D \cap \mathcal{G}_{disk}(R)$ is not spatially distributed over $\mathcal{G}_{disk}(R)$ [44, Sec. 2.2.1], as opposed to $\mathcal{G}_{LV}(r)$ (see Remark 3.4.2). An example is shown in Fig. 3.1b, where dashed lines show the invalid partitioning generated considering R, and overlapping areas are shown in purple. In fact, a valid partition of the environment is a collection of subsets of $\mathbb{R}^2$ that have disjoint interiors.*

In addition to what stated in Remark 3.4.4, there exists another reason to avoid using the whole sensing range R instead of r: the area of the partition changes in an abrupt manner when a neighbor is added or removed from $\mathcal{N}_{\mathcal{G}_{disk}}(p_i, \mathcal{P})$. This issue is illustrated in Fig. 3.2, where the difference of change from Fig. 3.2a to Fig. 3.2b is relevant if compared with the one from Fig. 3.2c to Fig. 3.2d. This sudden change would cause a chattering movement of the centroid of this partition, which would generate undesirable oscillations in the control law. Instead, if we consider the proposed *limited* Voronoi partitioning $\mathcal{V}^r(\mathcal{P})$ as defined in (3.8), we obtain a progressive change of the area, and consequently of the position of the centroid.

3.5 Proposed Control Strategy

In this section, we describe the proposed control strategy to solve Problem 3.2.1. For this purpose, we consider the environment Q to be partitioned according to an exact Voronoi partitioning $\mathcal{V}(\mathcal{P})$, as defined in (3.2). Hence, inspired by [84], we consider the following performance function:

$$f^r(x) = -x^2 1_{[0,r]}(x) - r^2 1_{(r,\infty)}(x), \tag{3.10}$$

which leads to optimally deploying robots in their corresponding Voronoi cell, within the ball $\overline{B}(p_i, r)$ (first term), and is constant outside the ball (second term).

Following the definition in (3.3), the optimization function can then be derived as

$$\mathcal{H}^r_{\mathcal{V}}(\mathcal{P}) = -\sum_{i=1}^{N} \int_{V_i(\mathcal{P})} \|q - p_i\|^2 1_{[0,r]} \phi(q) dq - r^2 \sum_{i=1}^{N} \int_{V_i(\mathcal{P})} 1_{(r,\infty)} \phi(q) dq. \tag{3.11}$$

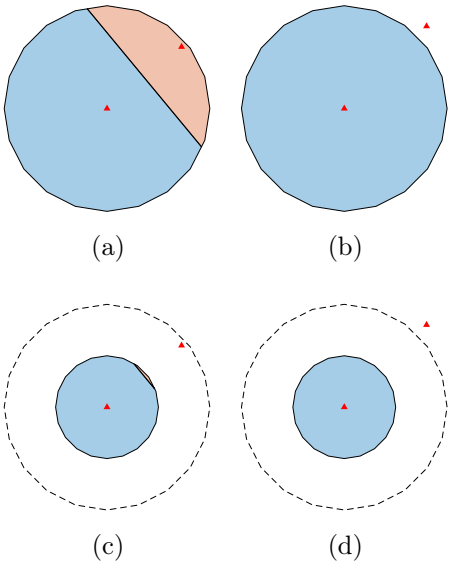


Figure 3.2: Local Voronoi partition when a neighbor is removed from $\mathcal{N}_{\mathcal{G}_{disk}}(p_i, \mathcal{P})$ (in the left images is still a neighbor and in the right ones is outside the sensing range R). The *blue* area is the partition to the i-th robot with respect which we calculate the local Voronoi, the *red* area is associated to the neighbor robot, when presents, and the robots are represented with the *triangles*. Fig. 3.2a and Fig. 3.2b report the local Voronoi cell, instead in Fig. 3.2c and Fig. 3.2d the cell is intersected with $\overline{B}(p_i, \frac{R}{2})$, and the *dotted* line represents the sensed area.

We introduce now the proposed control law:

$$u_i = k(C_{V_i^r} - p_i), \tag{3.12}$$

where $k \in \mathbb{R}_{>0}$ is a proportional gain, and $C_{V_i^r}$ is the centroid of the cell V_i^r defined by the r-limited Voronoi diagram of robot i.

Theorem 3.5.1. *Consider a multi-robot system composed of N robots with limited sensing range whose dynamics evolves according to* (3.1)*. Then, the control law* (3.12) *leads to maximize the optimization function* (3.11)*.*

Proof. We will show that control law (3.12) implements a gradient ascent of the optimization function (3.11). Namely, we will show that the time derivative of the optimization function, that is

$$\dot{\mathcal{H}}_{\mathcal{V}}^r(\mathcal{P}) = \sum_{i=1}^{N} \frac{\partial \mathcal{H}_{\mathcal{V}}^r(\mathcal{P})}{\partial p_i}^T \dot{p}_i, \tag{3.13}$$

is non-negative.

For this purpose, consider the definition of the optimization function in (3.11). Following the same computation as in (3.4), the first term can be rewritten by means of the polar moment of inertia. Furthermore, the second term represents the area,

weighted with function ϕ, of the portion of Q that is not covered by the union of the balls $\overline{B}(p_i, r)$ (in *green* in Fig. 3.1b). Hence, we can rewrite (3.11) as:

$$\mathcal{H}^r_{\mathcal{V}}(\mathcal{P}) = -\sum_{i=1}^{N} J_{V_i(P)\cap\overline{B}(p_i,r),p_i} - r^2 Area_\phi(Q \setminus \cup_{i=1}^{N} B(p_i, r)), \tag{3.14}$$

with $Area_\phi(S) = \int_S \phi(q)dq$, $S \subset Q$.

Given [84, Thm. 2.2], considering the choice of the performance function (3.10), we can compute the partial derivatives of the optimization function $\mathcal{H}^r_{\mathcal{V}}(\mathcal{P})$ as follows:

$$\begin{aligned} \frac{\partial \mathcal{H}^r_{\mathcal{V}}(\mathcal{P})}{\partial p_i} &= 2\int_{V_i(P)\cap\overline{B}(p_i,r)} (q - p_i)\phi(q)dq \\ &= 2M_{V_i\cap\overline{B}(p_i,r)}(C_{V_i\cap\overline{B}(p_i,r)} - p_i) \end{aligned} \tag{3.15}$$

where $M_{V_i\cap\overline{B}(p_i,r)}$ and $C_{V_i\cap\overline{B}(p_i,r)}$ are respectively the mass and the centroid of the cell given by the intersection of $\overline{B}(p_i, r)$ with the exact Voronoi partitioning of the environment.

Considering the control law (3.12) and the dynamics (3.1), the time derivative of the optimization function (3.13) can be written as follows:

$$\dot{\mathcal{H}}^r_{\mathcal{V}}(\mathcal{P}) = \sum_{i=1}^{N} \left[2M_{V_i\cap\overline{B}(p_i,r)}(C_{V_i\cap\overline{B}(p_i,r)} - p_i)\right]^T k(C_{V_i^r} - p_i). \tag{3.16}$$

Defining $\kappa_i = k\, 2M_{V_i\cap\overline{B}(p_i,r)} > 0$, we can rewrite (3.16) as follows:

$$\dot{\mathcal{H}}^r_{\mathcal{V}}(\mathcal{P}) = \sum_{i=1}^{N} \kappa_i (C_{V_i\cap\overline{B}(p_i,r)} - p_i)^T (C_{V_i^r} - p_i) \tag{3.17}$$

Consider now the results of Lemma 3.4.1, we obtain the following:

$$C_{V_i^r} = C_{V_i\cap\overline{B}(p_i,r)} \quad \forall i = 1, \ldots, N \tag{3.18}$$

Therefore, from (3.17) and (3.18), we can state that

$$\dot{\mathcal{H}}^r_{\mathcal{V}}(\mathcal{P}) = \sum_{i=1}^{N} \kappa_i \left\| C_{V_i\cap\overline{B}(p_i,r)} - p_i \right\|^2 \geq 0 \tag{3.19}$$

which proves the statement. □

According to Theorem 3.5.1, we can conclude that the proposed control law (3.12), which can be computed by each robot based only on directly sensed information, leads to optimally cover the environment.

During the execution of the control algorithm, every robot continuously updates the limited Voronoi diagram and moves towards the centroid of the respective Voronoi cell. The centroid computation and, hence, the control law (3.12), depend on the neighbors detected and the environment sensed. In other words, given a constant $\phi(q)$, every robot will tend to move far from other robots and from the environment boundary, while maximizing the coverage of the area.

3.6 Varying-Variance Control

In the proposed control strategy we consider a probability density function $\phi : Q \to \mathbb{R}_{>0}$ in order to measure the position in Q where an event of interest occurs. Following the control law presented in (3.12), each robot has to compute the centroid $C_{V_i^r}$ of its cell V_i^r weighted by the density function. We assume ϕ has the form of a Gaussian distribution[1]. Gaussian functions are, indeed, characterized by various useful properties making easier the computation of the weighted mass and centroid. In [99], for example, the authors propose a line integration method to compute mass and centroid of a Voronoi cell, which alleviates the computational issues compared with the spatial discretization method.

In order to keep the notation simple, we will first consider a one-dimensional density function $\phi : \mathbb{R} \to \mathbb{R}_{\geq 0}$ with the form:

$$\phi(x) = exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{3.20}$$

where $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_{>0}$ are the mean and the variance of the density function. The one-dimensional case can be easily extended to the two-dimensional one, as will be shown at the end of the section. The Gaussian function is characterized by a high gradient, such that for points far from the mean value the Gaussian function assumes almost endlessly small values. This feature typically causes numerical issues for experiments or simulations in large environments. For example, we consider a standard Gaussian distribution $\phi : \mathbb{R} \to \mathbb{R}_{\geq 0}$ ($\mu = 0$ and $\sigma = 1$). We have that $\phi(x = 38) \approx 2.7e^{-314}$, while the smallest positive double precision floating-point number in IEEE standard is $\approx 2.2e^{-308}$: this makes the portion of the area for $x \geq 38$ unusable. In a coverage problem, the σ parameter determines how scattered the group of robots should cover the area around the μ value. However, if the robot is located too far from the mean value μ, it faces numerical issues and it might be not able to compute the control input.

We propose a robot control characterised by a self adapting variance whose value depends on the position of the robot relatively to μ. We define as $\phi_i(x)$ the density function processed by the robot i, with σ_i and μ_i the variance and the mean, respectively. The robot's controller artificially increases the variance of the processed density distribution ϕ_i depending on the robot position and on the computational capacity of the robot's CPU, e.g., float or double variables. The variance value converges to the desired one as the robot moves close to the mean value. Thus, we have that $\mu_i = \mu$ since the controller acts only on the variance.

Given a generic Gaussian distribution $\phi(x)$ with variance σ and mean μ, we can define the corresponding standard Gaussian distribution $\phi(z)$ where $z = \left|\frac{x-\mu}{\sigma}\right|$. In this way we can define a single constant parameter z_{max} above which the Gaussian distribution gives numerical issues. This parameter depends on the computational capacity of the hardware platform, e.g., $z_{max} = 38$ for double variables or $z_{max} = 13$

[1]It is worth noting that the proposed concepts can be easily extended to general probability density functions, approximated as combination of Gaussian distributions.

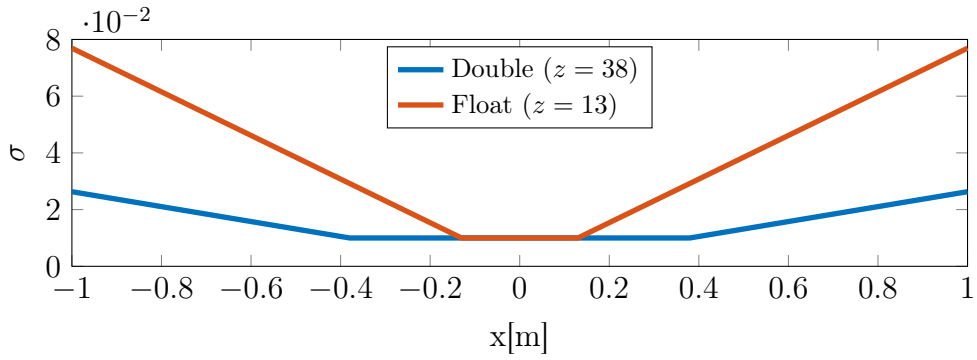


Figure 3.3: One-dimensional representation of the varying-variance with different values of z corresponding to the principal numerical representation ($\sigma = 0.01$, $\mu = 0$).

for float variables. Thus, we can easily determine if a value x_i brings to numerical issues and we can change the variance value of the processed Gaussian function $\phi_i(x_i)$ consequently. The processed variance σ_i becomes a function of x_i and assumes the form:

$$\sigma_i(x_i) = \begin{cases} \sigma & \text{if } z_i \leq z_{max} \\ \left| \dfrac{x_i - \mu}{z_{max}} \right| & \text{otherwise,} \end{cases} \tag{3.21}$$

where $z_{max} \in \mathbb{R}_{>0}$ is a parameter that depends on the CPU capacity. Fig. 3.3 graphically represents (3.21), which describes the proportional relation between robot position and variance. As the robot approaches the mean value μ, the variance σ_i decreases until it reaches the value of σ, i.e., the variance of density function ϕ over the environment.

The solution presented in (3.21) applies to one-dimensional density function. However, the same idea can be easily extended to a two-dimensional density function, as we have in the coverage problem. Given the density function $\phi : Q \to \mathbb{R}_{\geq 0}$, where $Q \subset \mathbb{R}^2$, we can define:

$$\sigma_i(p_i) = \begin{cases} \sigma & \text{if } z_{i,Q} \leq z_{max} \\ \dfrac{\|p_i - \mu\|}{z_{max}} & \text{otherwise,} \end{cases} \tag{3.22}$$

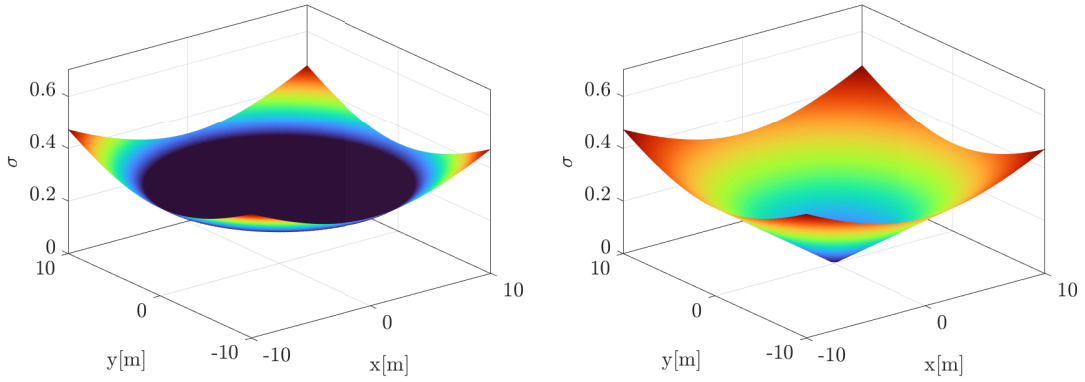
where:

$$z_{i,Q} = \frac{\|p_i - \mu\|}{\sigma} \tag{3.23}$$

Fig. 3.4 reports two examples of the surfaces that map the position p_i to the corresponding $\sigma_i(p_i)$.

3.7 Experimental Validation

In this section we report the simulations and the real experiments we carried out to verify the proposed algorithm.



(a) $\sigma = 0.3$ (b) $\sigma = 0.01$

Figure 3.4: Examples of two surfaces calculated with (3.21) ($z_{max} = 30$, $\mu = (0, 0)$).

3.7.1 Simulations

The simulations were carried out in MATLAB® and aimed at investigating the effectiveness of the proposed method, also in comparison to the standard method introduced in Section 3.3. When the density function ϕ is constant, the objective of the robots is to spread uniformly in the environment. To understand the difference between the proposed method and the standard one, we tested both in the same environment (Q is a square 6m $\times$ 6m).

Fig. 3.5 reports the optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ of the deployment of 4 robots controlled with (3.12), varying the sensing range, and with the standard method (3.7). We use the standard optimization function (3.4) as a benchmark for both our method and the standard one to obtain comparable measures. We take as reference the performance of the standard method (in *blue* in Fig. 3.5), which is not affected by the sensing range of the robots. Intuitively, robots with a small sensing range are not able to optimally cover a large environment, when compared to an ideal condition of unlimited sensing. Thus, in the proposed control strategy the robots converge to a sub-optimal solution, which is the best result the network can achieve given the limited sensing capabilities. Thus, the coverage performance with small sensing range robots (e.g., $R = 1$m) controlled by our methodology is worse with respect to the standard one. However, when the sensing range is sufficiently big (e.g., $R = 3$m) the performance is comparable with the one obtained with the standard method and, with $R = 6$m, the performance obtained is the same as with the standard method. To visually explain the difference at steady-state when the sensing range is small, we report, in Fig. 3.6, the initial (circles) and final positions of the robots (triangles), and the Voronoi partitions for both our case (in *blue*) and the standard method (in *red*). With our method it is clear how, with a sufficiently large environment, the robots converge to positions in which the neighbors $\mathcal{N}_{\mathcal{G}_{disk}}(p_i, \mathcal{P})$ are on the boundary of $\overline{B}(p_i, R)$, and the balls $\overline{B}(p_i, r)$ are tangent to each others, $\forall i$ (see Fig. 3.6).

Fig. 3.7 reports the optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ as a benchmark for different

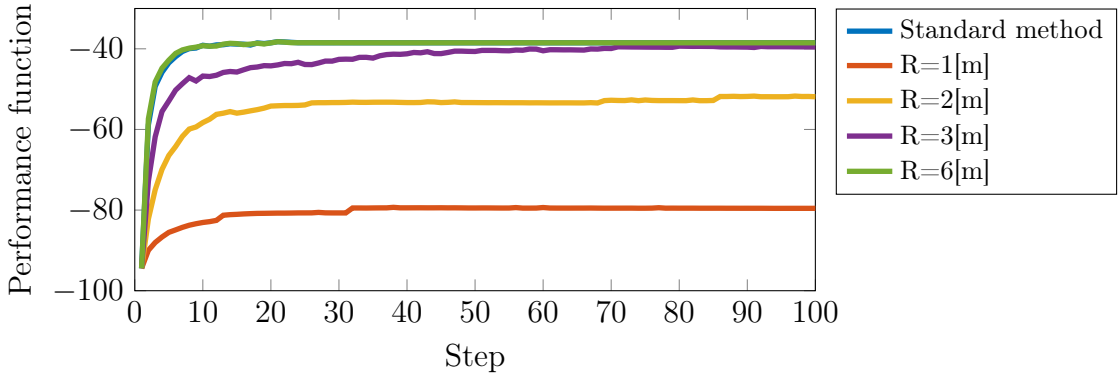


Figure 3.5: Optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ with 4 robots of our method with increasing sensing range compared to the standard method. The proposed method converges to the standard one for sufficiently high sensing range.

numbers of robots controlled with the standard method (*solid* lines) and with our method with sensing range $R = 2$m (*dashed* lines). If a group of robots with small sensing range is not able to optimally cover a large environment, increasing the number of robots, intuitively, will improve the optimization function consequently. Thus, in the proposed control strategy the low number of robots converge to a sub-optimal solution of the coverage problem when compared to the standard one, due to the limited sensing capabilities. In fact, with 5 robots (in *blue* in Fig. 3.7) our method does not perform as the standard one. Instead, increasing the number of robots the performance is comparable. The only difference is the transient, where the convergence is reached slowly with our method as the centroids are closer to the robots' position. In our method the centroid is in $\overline{B}(p_i, r)$, while in the standard method it is inside the Voronoi cell: hence, if the cell is big, the centroid can be further away from the position of the robot. It is worth noting that the steady-state value is higher with a higher number of robots given the chosen optimization function. In fact, if the number of robots is higher then the area is better allocated to the robots, namely each point $q \in Q$ is closer to a robot than in the case with fewer robots.

It is worth noting that these results allow us to confirm Lemma 3.4.3. In fact, increasing the half sensing range or the number of robots we obtain the same value of the optimization function because the underlying limited Voronoi diagram is equal to the exact one.

3.7.2 Implementation on the Robots

Some preliminary tests were carried out on real robotic platforms. The hardware set-up was made up of a group of TurtleBot 3 Burger robots, equipped with a Raspberry Pi 4 mounting Ubuntu 20.04, and of an OptiTrack tracking system. The proposed algorithm was implemented in C++ and, for the calculation of the limited Voronoi,

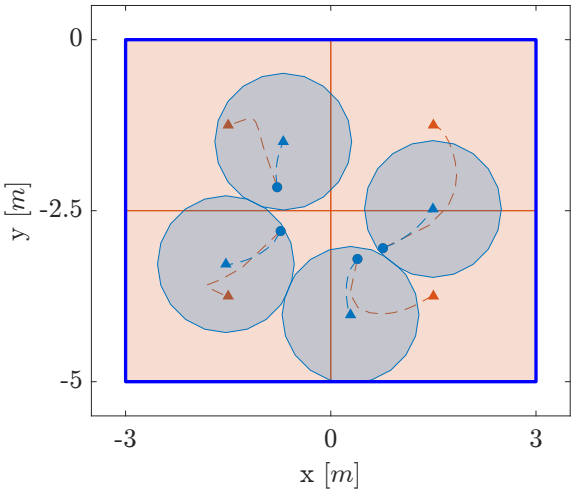


Figure 3.6: Difference between Voronoi partition with the standard method (in *red*) and the proposed method with $R = 2$m (in *blue*). Initial and final positions are represented with *circles* and *triangles*, respectively.

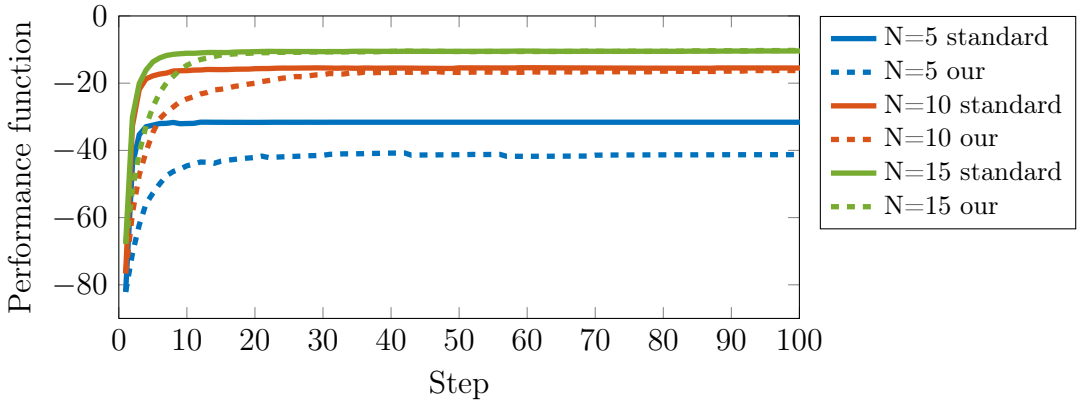
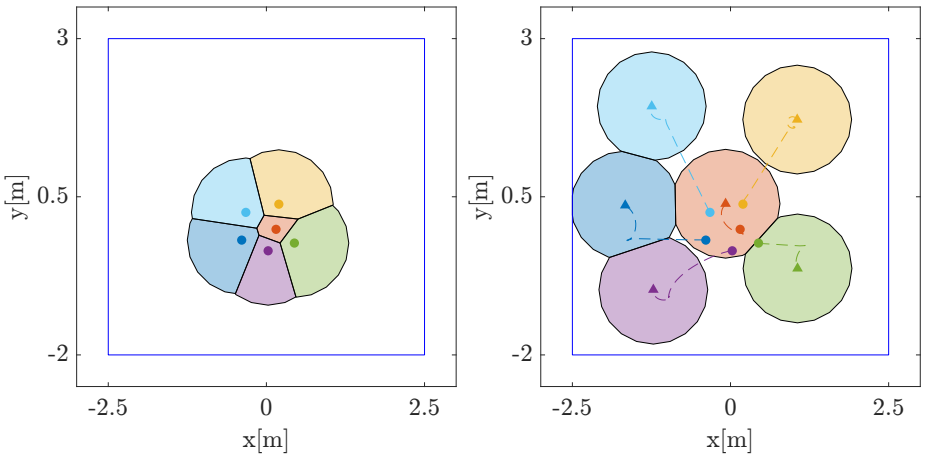


Figure 3.7: Optimization function $\mathcal{H}_{\mathcal{V}}(\mathcal{P})$ with different number of robots. The *solid* lines represent the standard method, instead the *dashed* lines represent our method.

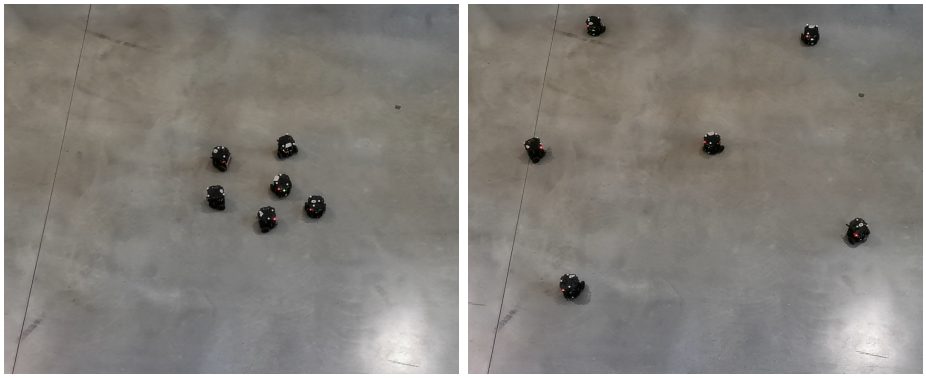
we adapted a library[2] developed with the Fortune Algorithm [69]. The interface with the robots was implemented in ROS2, which managed also the communication with the OptiTrack system. It is worth noting that the proposed method does not need any communication between the robots and that the external tracking system can be replaced by on board sensors. In fact, the OptiTrack system was used to emulate a limited range sensor that is able only to measure neighbors' position. This information was used to create the limited Voronoi diagram with the C++ library, which calculated also the centroid of the cell based on the density function. Hence, each robot was able to calculate on-board the limited Voronoi cell V_i^r, and the centroid $C_{V_i^r}$. To obtain a faster implementation we used the line integration method proposed in [99]. The input calculated in terms of $\dot{p}_i$ from (3.12)

[2]https://github.com/pvigier/FortuneAlgorithm



(a) Initial positions and limited Voronoi cells.

(b) Final positions and limited Voronoi cells.



(c) Initial position of the real robots.

(d) Final position of the real robots.

Figure 3.8: Experiments with uniform density function. In 3.8b and 3.8a every colored shape is the limited Voronoi cell of the respective robot. In 3.8b the *circles* are the starting positions, the *triangles* are the final positions.

was modified accordingly to the input-output state feedback linearization [68] to be correctly applied to the robot.

In the first experiment, the density function $\phi(q)$ was constant, hence, the aim of the robots is to spread into the environment. Fig. 3.8 reports the successful result of the experiment: initial and final positions of the robots are reported together with their respective limited Voronoi cell.

The second experiment aimed at testing the algorithm with a non uniform density function. We assume $\phi(q)$ has the form of a Gaussian distribution, with $\mu = (0, 0)$ and $\sigma = 0.3$. Fig. 3.9 reports the trajectories of the robots from the initial positions (circles) to the final positions (triangles). As expected, the robots converged to the center and they rounded up proportionally to the variance. The Gaussian distribution is represented with the *black* cross, which is the mean value, and the *black* concentric circles, which represent equidistant levels of the variance. The varying-variance control proposed in Section 3.6 was applied to consider the computation

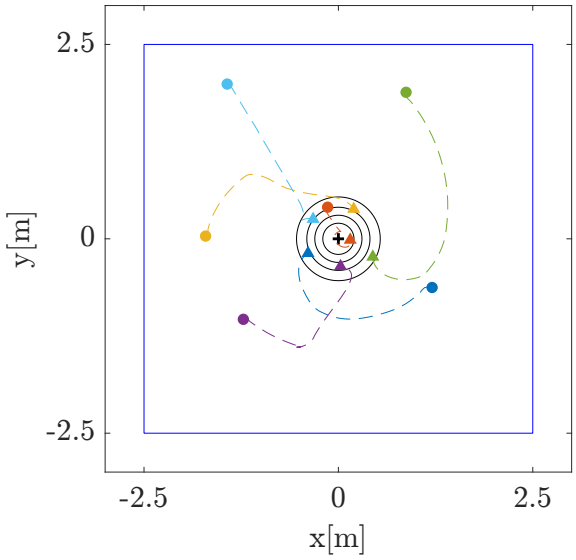


Figure 3.9: Experiments with not uniform density function $\phi(q)$. Initial and final positions with *circles* and *triangles*, respectively. The *black* cross is the center of $\phi(q)$ and the *black* concentric circles are different values of variance.

limit of the on board computer ($z_{max} = 37$ for calculation with double values). Hence, each robot modified $\sigma_i(p_i)$ accordingly to (3.22), if needed.

3.8 Conclusions

In this chapter, we presented a distributed coverage based control to coordinate a group of autonomous robots with limited sensing capabilities in an unknown environment. The proposed novel control strategy, based on a *limited* Voronoi partitioning, uses only the information directly detected by the robot, and it does not rely on a communication network. We formally show how the control law drives the group of robots to the respective centroid of the *limited* Voronoi cell, thus reaching an optimal coverage, taking into account the sensing capabilities of the robots. In addiction, we presented a methodology to deal with numerical issues that come from the Gaussian-type probability density function typically used in the coverage base control algorithms. The varying-variance methodology connects the control law with the computational capacity of the robot. A series of experiments and simulations were performed on a group of mobile robots to validate the performance of the proposed control method.

Chapter 4

Robust Stability: a Control Barrier Functions Approach

Passivity-based control ensures the implementation of a desired behavior in a controlled system, while preserving robust stability. In this chapter we propose a general purpose method for guaranteeing the passivity of a controlled system, while minimally affecting the desired behavior. This is achieved using both energy tanks and Control Barrier Functions (we refer to Section 2.1.3 for the theory on CBFs). We propose a method that can guarantee passivity in multiple domains and applications, and achieve this implementing the best approximation of a desired behavior. We consider systems that are intrinsically passive, such as mechanical systems, which are the majority of the systems used in robotics and in general applications (e.g., a robot is a mechanical system). However, control actions may be applied that lead the system to be non-passive. Hence, the application of the proposed method allows us to implement the best approximation of the desired input and to maintain passivity, and consequently robust stability. The method proposed in this chapter has been published in [100].

4.1 Introduction

Passivity-based control is a popular tool for guaranteeing robust stability of a system, such as in the presence of partially unknown dynamic elements or in case of interaction with a possibly unknown environment [101, 102, 62]. The concept of passivity can be effectively applied introducing physical analogies related to energy and power exchange among different parts of a system: this allows the easy application of these concepts to physical systems, such as robots. Physical systems are inherently passive, however, simple control actions can be defined that implement non-physical behaviors, such as time-varying potential fields [103] or variable admittance control [104], which may result into a non-passive system.

Passivity-based controllers are often associated with specific physical dynamics, and this strongly limits the flexibility that can be achieved without losing passivity. Recently, several approaches have been proposed in order to separate passivity from specific dynamics. In [105, 106, 107] the time domain passivity approach is used for

establishing passivity by activating variable dampers when some energy production is detected. In [108] passivity has been achieved by properly modulating the set point of the controlled system.

Energy tanks [109] have been used for disembodying passivity from any specific dynamics. They have been employed in multiple applications, such as force control [110], bilateral teleoperation [111], and in multi-robot systems for maintaining connectivity [112], for teleoperating a group of robots with time-varying interconnections [113], and for variable interconnections with modulated multi-port tank [114].

Similarly to other strategies to ensure passivity, tank-based methods require the application of constraints, typically modifying the desired control action (whose implementation would lead to loss of passivity), with subsequent performance decrease. In order to minimize this undesirable effect, it is useful to reduce, as much as possible, the modification of the desired control action, while taking into account the evolution of the energy in the tank, thus preventing undesired behaviors.

Control Barrier Functions (CBFs) [49, 50] are an appropriate tool to achieve these objectives. In fact, they ensure the maintenance of some properties, while at the same time enabling the modification of the desired input to the system in a minimally invasive manner. While other traditional methods typically consist of filtering out the desired control action as soon as the limit of a certain constraint has been reached (see, e.g., [111, 104]), CBFs enforce the system to evolve inside a *safe* set, i.e., in such a way that a desired constraint is always satisfied.

The application of the proposed method allows us to implement the best approximation of the desired input and to maintain passivity, and consequently robust stability. This is achieved exploiting an optimization-based framework. Along the same lines, an optimized admittance controller was proposed in [115, 116], which is guaranteed to be passive using energy tanks. Differently from [115, 116], in this chapter we propose a general formulation that can be used in multiple applications and we encode the constraint on passivity using CBFs, thus providing the advantages reported above.

CBFs have been used to ensure passivity also in [117], where the authors implemented an optimized damping injection, which produces undesired dynamic effects. In order to avoid this issue, in this chapter we design a CBF that guarantees the best approximation of the desired input, without damping injection: this is achieved using the energy stored in the tank in a more efficient way.

4.2 Passivity and Energy Tanks

Passivity is linked to the concept of energy conservation inside a system [101]: the energy that can be extracted from a system must be smaller than or equal to the energy that enters in the system, namely the system must not produce energy. Passivity can be used to implement desired behaviors in robotic systems and it does this in a robust stable way [62].

Consider a mechanical system represented by an input-output gravity compen-

sated Euler-Lagrange model:

$$\begin{cases} M(\chi)\ddot{\chi} + C(\chi,\dot{\chi})\dot{\chi} + D\dot{\chi} + \frac{\partial P(\chi)}{\partial \chi} = u \\ y = \dot{\chi} \end{cases} \tag{4.1}$$

where $\chi \in \mathbb{R}^n$ is the position of the system, the state of the system is represented by $x = \left[\chi^T, \dot{\chi}^T\right]^T$, and u, $y \in \mathbb{R}^n$ are the input and the output of the system, respectively. $M(\chi) \in \mathbb{R}^{n\times n}$ is the positive symmetric matrix that represents the inertia, $C(\chi,\dot{\chi}) \in \mathbb{R}^{n\times n}$ represents the Coriolis effects, and $D(\chi) \in \mathbb{R}^{n\times n}$ is the non-negative symmetric matrix that represents damping. $P(\chi) : \mathbb{R}^n \to \mathbb{R}$ is a potential field acting on the system (e.g., gravity, artificial potential field, elastic potential). Considering a Euler-Lagrange system does not reduce the generality of the proposed method, as most of the robotic systems can be represented with this model (e.g., a manipulator, a mobile robot, a multi-robot system). The system (4.1) is passive with respect to the pair $(u, y) = (u, \dot{\chi})$ if there exists a lower bounded C^1 function $H(x) : \mathbb{R}^n \to \mathbb{R}$, called a storage function, such that:

$$H\left(x(t)\right) - H\left(x(0)\right) \leq \int_0^t u(\tau)^T y(\tau) d\tau \tag{4.2}$$

It is well known [101] that (4.1) is passive with respect to the pair $(u, \dot{\chi})$, considering

$$H(x) = P(\chi) + \frac{1}{2}\dot{\chi}^T M(\chi)\dot{\chi} \tag{4.3}$$

as storage function. In fact, (4.1) is a mechanical system, which is intrinsically passive, and the storage function represents the (generalized) stored energy of the system. Calculating the total derivative of (4.3) and considering the dynamics (4.1), we obtain the power balance for (4.1):

$$\begin{aligned} \dot{H}(x) &= \frac{\partial P(\chi)}{\partial \chi}\dot{\chi} + \dot{\chi}^T M(\chi)\ddot{\chi} + \frac{1}{2}\dot{\chi}^T \dot{M}(\chi)\dot{\chi} \\ &= u^T\dot{\chi} - \dot{\chi}^T D(\chi)\dot{\chi} + \dot{\chi}^T(\dot{M}(\chi) - 2C(\chi,\dot{\chi}))\dot{\chi} \\ &= u^T y - P_d(x) \end{aligned} \tag{4.4}$$

where $P_d(x) \geq 0$ is the power dissipated by the system, and $u^T y$ is the (generalized) power exchanged by the system. The term relative to $\dot{M}(\chi) - 2C(\chi,\dot{\chi})$ is equal to zero for the property of the Euler-Lagrange model [101]. The dissipated power $P_d(x)$ is considered as a *passivity margin* that is used in energy tanks' approach [112, 111, 114]. In fact, energy tanks introduce an additional (fictitious) component to the system that (virtually) stores the energy dissipated, in such a way to be able to use this *passivity margin* for implementing passively a desired behavior, which otherwise could lead to a loss of passivity. The general formulation of an energy tank is:

$$\begin{cases} \dot{x}_t = u_t \\ y_t = \frac{\partial T(x_t)}{\partial x_t} = x_t \end{cases} \tag{4.5}$$

where x_t, u_t, $y_t \in \mathbb{R}$ are, respectively, the state, the input and the output of the tank, and $T(x_t) : \mathbb{R} \to \mathbb{R}$:

$$T(x_t) = \frac{1}{2}x_t^2 \tag{4.6}$$

is the energy stored inside the tank. Generally, if the connection between the system and the tank is power preserving, and the energy stored inside the tank never becomes null, then the overall system is passive (see [109, 111] for a more formal discussion). In fact, the tank integrates the power of the system and, in case of necessity, provides part of the stored energy to balance the overall exchange of energy.

4.3 Control Barrier Function for Passivity Preservation

The proposed CBF for passivity preservation is formulated in such a way to ensure that the energy inside the tank never goes below a given threshold, i.e., it never becomes empty. The tank is then attached to the system that must be kept passive. Attaching the tank means that we can monitor the energy dissipated by the system and the components that can carry to a non-passive system. This may be thought as a parallelism with the idea of port connection in the general theory of tanks [111, 112, 114]. In a similar way, also in the proposed method the tank needs to be defined in such a way that it exchanges power with the system.

For this purpose, we consider the system (4.1) in which we split the input in two terms: $u = u_e + u_c$, where u_e is an external input and u_c is the desired control input of the system. We use u_c to implement the desired dynamics (e.g., time-varying potential field, inertia and damping):

$$u_c(\chi, t) = -M_c(\chi, t)\ddot{\chi} - D_c(\chi, t)\dot{\chi} - \frac{\partial P_c(\chi, t)}{\partial \chi} \tag{4.7}$$

Hence, we consider the modified Euler-Lagrange model:

$$\begin{cases} \bar{M}(\chi, t)\ddot{\chi} + \bar{D}(\chi, t)\dot{\chi} + \frac{\partial \bar{P}(\chi, t)}{\partial \chi} = u_e \\ y = \dot{\chi} \end{cases} \tag{4.8}$$

where $\bar{M}(\chi, t) = M(\chi) + M_c(\chi, t)$, $\bar{D}(\chi, t) = D(\chi) + D_c(\chi, t)$, and $\bar{P}(\chi, t) = P(\chi) + P_c(\chi, t)$ consider the original model (4.1) together with the new desired dynamics, introduced by u_c. We omitted the obvious time dependence of χ and $\dot{\chi}$. Similarly to (4.3), we define the storage function $\bar{H}(x, t)$ for the modified model (4.8) as:

$$\bar{H}(x, t) = \bar{P}(\chi, t) + \frac{1}{2}\dot{\chi}^T \bar{M}(\chi, t)\dot{\chi} \tag{4.9}$$

We calculate the total derivative of $\bar{H}(x, t)$:

$$\begin{aligned} \dot{\bar{H}}(x, t) &= \frac{\partial \bar{P}(\chi, t)}{\partial \chi}\dot{\chi} + \dot{\chi}^T \bar{M}(\chi, t)\ddot{\chi} + \frac{\partial \bar{H}(x, t)}{\partial t} \\ &= u_e^T(t)y(t) - P_d(x, t) + \frac{\partial \bar{H}(x, t)}{\partial t} \end{aligned} \tag{4.10}$$

where the last term $\frac{\partial \bar{H}(\chi,t)}{\partial t}$ is sign indefinite, as it includes all the time-varying components of (4.8). Hence, integrating (4.10) we are not sure to respect the passivity condition (4.2) for (4.8) with respect to the pair (u_e, y), considering (4.9) as storage function. For ease of notation, we will hereafter omit the explicit dependence on time when not strictly necessary. We can rewrite (4.10) as:

$$\dot{\bar{H}}(x) = -\Xi(x) + \Omega(x) + u_e^T y \tag{4.11}$$

where $\Omega(x) : \mathbb{R}^n \to \mathbb{R}$ represents the components in the system that can lead to a non-passive system, namely the sign of $\Omega(x)$ is not defined, and $\Xi(x) : \mathbb{R}^n \to \mathbb{R}$ is the dissipated power, that is $\Xi(x) \geq 0$ and it includes the term $P_d(x)$. $\Omega(x)$ incorporates the energetic contribution of the control input u_c. In order to restore the passivity of the system, which may be violated by (4.11), we would like to apply the following input to the tank:

$$u_{t,des} = \frac{1}{x_t}\left(\Xi(x) - \Omega(x)\right) \tag{4.12}$$

in such a way to store energy in the tank with $\Xi(x)$ and to release or to store energy with $\Omega(x)$. In fact, using (4.12) and (4.5), we can see that the variation of energy in the tank (4.6) is given by:

$$\dot{T}(x_t) = \Xi(x) - \Omega(x) \tag{4.13}$$

We can define the whole exchange of power between the system and the tank as $\Sigma(x)$:

$$\Sigma(x) = -\Xi(x) + \Omega(x) \tag{4.14}$$

As it will be shown in Theorem 4.3.3, the augmented system (tank + system) is passive until the tank has energy, i.e., $T \neq 0$. To ensure this condition we propose the following CBF $h(x_t)$, considering the system (4.5):

$$h(x_t) = \frac{1}{2}x_t^2 - \varepsilon \tag{4.15}$$

where $\varepsilon \in \mathbb{R}_{>0}$ is a threshold value for the lower value of the energy. Hence, the safe set in which we want to keep the system is: $\mathcal{C}_t = \{x_t | T(x_t) \geq \varepsilon\}$.

In order to find the controller $u_t^* \in K_{cbf}(x_t)$ we have to solve the QP problem (2.6), considering the system (4.5) and the CBF (4.15):

$$\begin{aligned} u_t^* = \underset{u_t \in \mathbb{R}}{\text{argmin}} \quad & \frac{1}{2}\|u_t - u_{t,des}\|^2 \\ \text{s.t.} \quad & x_t u_t \geq -\alpha\left(\frac{1}{2}x_t^2 - \varepsilon\right) \end{aligned} \tag{4.16}$$

The desired input $u_{t,des}$ is calculated based on the connection between the tank and the system (4.12).

Remark 4.3.1. *To obtain a well-defined optimization problem, we should ensure that $u_{t,des}$ is Lipschitz continuous. We note that the singularity caused by $x_t = 0$ is avoided thanks to the proposed CBF, see Remark 4.3.4 and Section 4.5.2. The other terms of* (4.12)*, namely $\Xi(x)$ and $\Omega(x)$, should be chosen Lipschitz continuous.*

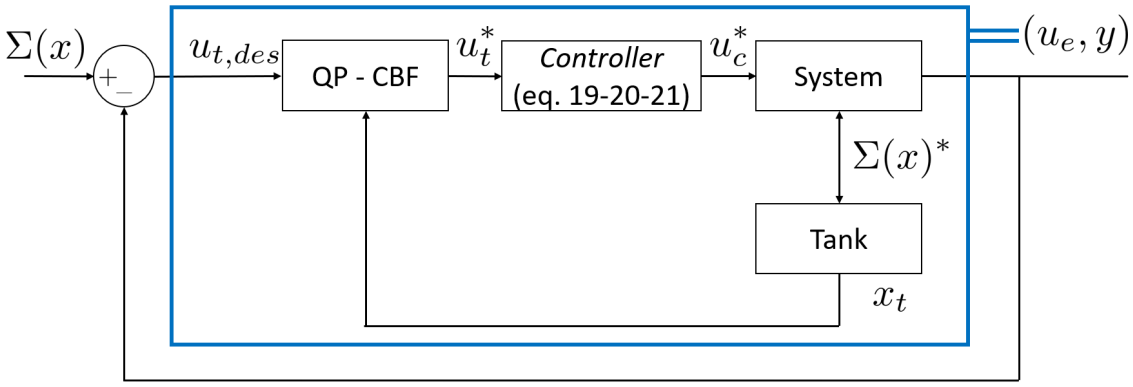


Figure 4.1: Diagram of the different components of the system. The *blue* box frames the overall system, which is passive with respect to the pair (u_e, y). $\Sigma(x)$ is considered as a set point for the system, given from an external controller that decides the desired behavior. From $\Sigma(x)$ we calculate the desired input for the tank $u_{t,des}$. With the solution of the QP problem (4.16), computed based on the state of the tank x_t, we find u_t^*. Using (4.17), (4.18), and (4.19) we calculate $\Omega(x)^*$ and from this we can obtain the modified control input u_c^*. The system and the tank exchange the allowed flow of power $\Sigma(x)^*$.

From the solution u_t^* of the QP problem (4.16) we can calculate the corresponding exchange of power $\Sigma(x)^* = -\Xi(x)^* + \Omega(x)^*$, which is the whole exchange of power between the system and the tank achieved applying the proposed CBF control, i.e., the behavior that we can implement in a passive manner considering the energy stored in the tank. Generally speaking, we can consider u_t^* as a scaled version of $u_{t,des}$, and we can calculate the scaling factor $\gamma \in \mathbb{R}$ as:

$$\gamma = \begin{cases} 0 & \text{if } u_{t,des} = 0 \\ \frac{u_t^*}{u_{t,des}} & \text{otherwise} \end{cases} \tag{4.17}$$

Hence, the allowed exchange of power is:

$$\Sigma(x)^* = \gamma \Sigma(x) \tag{4.18}$$

We choose to scale $\Omega(x)$ and to keep $\Xi(x)^* = \Xi(x)$, namely we change the control input u_c, applying instead u_c^*. Considering (4.14) and (4.18), we can calculate the scaled value $\Omega(x)^*$:

$$\Omega(x)^* = (1 - \gamma)\Xi(x) + \gamma\Omega(x) \tag{4.19}$$

It is worth noting that the dependence of $\Omega(x)^*$ on u_c^* can not be defined in general, but can be determined once the specific case study has been chosen. An example is provided in Section 4.4.1. Fig. 4.1 reports the diagram of the proposed structure to obtain passivity applying the CBF (4.15).

Remark 4.3.2. *We choose to scale just the component $\Omega(x)$ because we wanted to implement a behavior that can turn the system into a non-passive one, always*

preserving passivity, in a system with a given dissipative contribution, namely there exists $P_d > 0$. However, the proposed method is general and the scaling can be done based on the objective and on the type of controller of the system. For example, if the damping coefficient can be modified, it is possible to adjust it, while keeping the desired value of $\Omega(x)$.

Theorem 4.3.3. *Given the system* (4.8) *with storage function $\bar{H}(x)$ defined in* (4.9)*, connecting an energy tank to the system as in* (4.12)*, applying the CBF* (4.15) *through the solution of* (4.16)*, and calculating the allowed power exchange $\Sigma(x)^*$ as in* (4.18)*, then the overall system is guaranteed to be passive with respect to the pair (u_e, y).*

Proof. Consider the whole system made of the system (4.8) and of the tank (4.5). The total energy $W(x, x_t)$ will be equal to:

$$W(x, x_t) = \bar{H}(x) + T(x_t)$$

Considering the results of (4.16), calculating $\Sigma(x)^*$ as in (4.18) and applying (4.19), we obtain:

$$\dot{\bar{H}}(x) = -\Xi(x) + \Omega(x)^* + u_e^T y$$

Hence, we can calculate the total derivative of $W(x, x_t)$:

$$\begin{aligned}\dot{W}(x, x_t) &= \dot{\bar{H}}(x) + \dot{T}(x_t) \\ &= -\Xi(x) + \Omega(x)^* + u_e^T y + u_t^* y_t \\ &= \Sigma(x)^* + u_e^T y + u_t^* y_t\end{aligned}$$

Knowing that $\Sigma(x)^*$ is calculated according to (4.16), (4.17), (4.18):

$$\Sigma(x)^* + u_t^* y_t = 0$$

and by integrating $\dot{W}(x, x_t)$ we find that:

$$W(t) - W(0) = \int_0^t u_e(\tau)^T y(\tau) d\tau$$

which proves that the system is passive with respect to the pair (u_e, y), considering the definition of passivity (4.2). □

Remark 4.3.4. *The singularity that can be caused by $x_t = 0$ is avoided initializing the state of the tank to a value $x_t(0)$ such that $T(x_t(0)) > \varepsilon$. Then, to avoid singularity, it is necessary that the sign of x_t never changes. This is verified by the forward invariance of the set $\mathcal{C}_t$, namely the energy inside the tank may never become null, and hence the state of the tank can not change sign.*

Remark 4.3.5. *The CBF could have been defined to control directly the state of the tank x_t, but, as passivity operates in the energy domain, it is more accurate to force the constraint on the energy of the tank, rather then on the state of the tank.*

Usually the energy stored in the tank is also upper bounded in order not to be able to implement practically unstable behaviors [108]. This upper limit $\overline{T} \in \mathbb{R}_{>0}$ for $T(x_t)$ can be checked before the update of the state imposing the following saturation to x_t:

$$x_t(t) = \min\left\{\int_0^t u(\tau)^* d\tau,\ 2\sqrt{\overline{T}}\right\} \tag{4.20}$$

4.4 Case studies

4.4.1 Case Study 1: Time-Varying Stiffness

As a simple case of the implementation of the proposed CBF, we consider a system made of a mass, a time-varying spring, and a damper. It is worth noting that a time-varying stiffness system is a prototype example for a variable admittance control, which can be used to control manipulators and other robotic systems. The dynamics of the system is given by:

$$m\ddot{\chi} + b\dot{\chi} + k(t)\chi = F_{ext} \tag{4.21}$$

where $\chi = \chi_{actual} - \chi_0$ (χ_{actual} is the position of the mass and χ_0 is the rest length of the spring, $\chi, \chi_0, \chi_{actual} \in \mathbb{R}$), $m \in \mathbb{R}_{>0}$ is the mass, $b \in \mathbb{R}_{>0}$ is the damping coefficient, $k(t) \in \mathbb{R}$ is the (variable) stiffness of the spring, and $F_{ext} \in \mathbb{R}$ is the possible external force acting on the system. The state $x \in \mathbb{R}^2$ of the system is defined as $x = [\chi, \dot{\chi}]^T$ and the desired control input $u_c = -k(t)\chi$ is the contribution of the time-varying stiffness[1]. The energy of the system $H_1(x) : \mathbb{R}^2 \to \mathbb{R}$ is equal to:

$$H_1(x) = \frac{1}{2}\chi^2 k(t) + \frac{1}{2}\dot{\chi}^2 m$$

We can calculate the total derivative of the energy of the system as follows:

$$\begin{aligned} \dot{H}_1(x) &= k\chi\dot{\chi} + m\dot{\chi}\ddot{\chi} + \frac{1}{2}\chi^2\frac{\partial k(t)}{\partial t} \\ &= F_{ext}\dot{\chi} - b\dot{\chi}^2 + \frac{1}{2}\chi^2\frac{\partial k(t)}{\partial t} \end{aligned} \tag{4.22}$$

In (4.22) we notice that the only term that can make the system non-passive is, as expected, the component linked to the variation of the spring stiffness over time. Hence, for this simple problem, we have that the components of (4.12) are:

$$\Xi(x) = b\dot{\chi}^2, \quad \Omega(x) = \frac{1}{2}\chi^2\frac{\partial k(t)}{\partial t}. \tag{4.23}$$

The desired input $u_{t,des}$ is calculated from the desired variation of stiffness:

$$u_{t,des} = \frac{1}{x_t}\left(b\dot{\chi}^2 - \frac{1}{2}\chi^2\frac{\partial k(t)}{\partial t}\right)$$

[1]Referring to the general definition of u_c in (4.7), we consider M_c and D_c equal to zero.

Finding the solution u_t^* of the QP problem (4.16), and solving (4.19), we can find $\Omega(x)^*$. Replacing $\Omega(x)$ with $\Omega(x)^*$ in (4.23), we obtain:

$$\frac{\partial k(t)^*}{\partial t} = \begin{cases} \Omega(x)^* \frac{2}{\chi^2} & \text{if } \chi \neq 0 \\ 0 & \text{otherwise} \end{cases},$$

and subsequently integrating we find $k(t)^*$, which is the actual value of stiffness that is allowed from an energetic point of view. Hence, we can calculate the control input u_c^* that we are allowed to apply to the system as follows:

$$u_c(t)^* = -k(t)^* \chi.$$

4.4.2 Case Study 2: Multi-Robot Control

Potential fields are one of the methods to control a group of robots [118]. They can be used to obtain several objectives, such as cohesive behavior [119], formation control [120], and collision avoidance [121]. We hereafter consider a time-varying potential field that can lead to a non-passive system. The time-varying nature of a potential field can be used to vary the control, to change the formation or the general behavior of the system.

We consider a multi-robot system composed by N robots that are able to move in a m-dimensional space. The dynamics of the i-th robot may be modeled as:

$$M_i \ddot{\chi}_i + B_i \dot{\chi}_i + \frac{\partial V_i}{\partial \chi_i} = F_{ext,i} \tag{4.24}$$

where $\chi_i \in \mathbb{R}^m$ is the position of the robot, $M_i \in \mathbb{R}^{m \times m}$ is the inertia matrix of the robot, $B_i \in \mathbb{R}^{m \times m}$ is the damping matrix, $\frac{\partial V_i}{\partial \chi_i} \in \mathbb{R}^m$ is the gradient of the potential field applied to the robot, namely the control input is $u_{c,i} = -\frac{\partial V_i}{\partial \chi_i}$, and $F_{ext,i} \in \mathbb{R}^m$ is a possible external force. We define the state of i-th robot $x_i = \left[\chi_i^T, \dot{\chi}_i^T\right]^T \in \mathbb{R}^{2m}$ and we stack all the single dynamics to define the overall state of the system $x = \left[x_1^T, \ldots, x_N^T\right]^T \in \mathbb{R}^{2mN}$. The external force forms, together with the velocity of the robot, the pair for the single robot $(u_{e,i}, y_i) = (F_{ext,i}, \dot{\chi}_i)$ and with the overall external force $F_{ext} = \left[F_{ext,1}^T, \ldots, F_{ext,N}^T\right]^T \in \mathbb{R}^{mN}$ we can define the pair for the system as $(u_e, y) = (F_{ext}, \dot{\chi})$, where $\dot{\chi} \in \mathbb{R}^{mN} = \left[\dot{\chi}_1^T, \ldots, \dot{\chi}_N^T\right]^T$ is the stack of the velocities of the robots.

The potential field $V_i(\chi_i, t)$ establishes an attractive force among the robots and it is time-varying, namely the desired distance among the robots may vary during time ($d_0 = d_0(t)$):

$$V_i(\chi_i, t) = \frac{1}{2} \sum_{j=1}^{n} k_{att} \left(d_{i,j} - d_0(t)\right) \tag{4.25}$$

where $d_{i,j}$ is the Euclidean distance between the i-th and the j-th robot, and $k_{att} \in \mathbb{R}_{>0}$ is a constant to tune the magnitude of the potential.

For each robot i, the energy $H_i(x_i) : \mathbb{R}^{2m} \to \mathbb{R}$ is:

$$H_i(x_i) = \frac{1}{2} \dot{\chi}_i^T M_i \dot{\chi}_i + V_i(\chi_i, t)$$

We calculate the total derivative of H_i:

$$\begin{aligned}\dot{H}_i(x_i) &= \dot{\chi}_i^T M_i \ddot{\chi}_i + \frac{\partial V_i}{\partial \chi_i}^T \dot{\chi}_i + \frac{\partial V_i(t)}{\partial t} \\ &= \dot{\chi}_i^T F_{ext,i} - \dot{\chi}_i^T B_i \dot{\chi}_i + \frac{\partial V_i(t)}{\partial t}\end{aligned}$$

where the components of (4.12) are: $\Xi(x_i) = \dot{\chi}_i^T B_i \dot{\chi}_i, \quad \Omega(x_i) = \frac{\partial V_i(t)}{\partial t}$ because the sign of the derivative with respect to time of the potential field is not known a priori. Hence, to enforce passivity, we can use the proposed CBF defining the dynamics of the tank as:

$$\dot{x}_{t,i} = \frac{1}{x_{t,i}} \left(\dot{\chi}_i^T B_i \dot{\chi}_i - \frac{\partial V_i(t)}{\partial t} \right) \tag{4.26}$$

The desired input of the tank $u_{t,des,i}$ is calculated based on (4.26), recalling that $u_t = \dot{x}_t$. Finally, the control applied to the robot $\frac{\partial V_i}{\partial \chi_i}$ is obtained from the solution of the QP problem (4.16). We add one tank for each robot, in such a way to ensure passivity for the overall system with respect to $(F_{ext}, \dot{\chi})$. In fact, the interconnection of passive systems remains passive [62].

4.5 Experimental Validation

In this section we report the results of the simulations and the experiments of the case studies described in Section 4.4. The validation aims at highlighting the flexibility of the proposed CBF in different applications.

In both simulations, carried out in Matlab, and real experiments we set the extended class $\mathcal{K}$ function α equal to the identity and the parameters of the tank were empirically chosen as follows: $\varepsilon = 0.1$, $\overline{T} = 100$ J, and $T(x_t(0)) = 0.125$ J (where $T(x_t(0))$ is the initialization value of the energy of the tank).

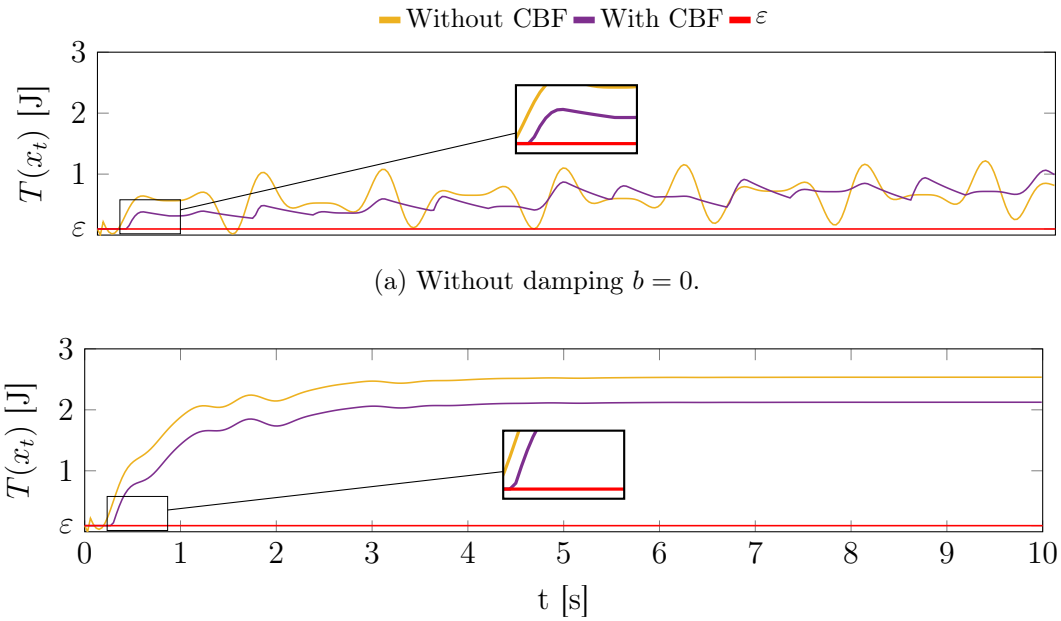
4.5.1 Case Study 1: Time-Varying Stiffness

System Performance Evaluation

In these simulations we investigated the behavior of the case study reported in Section 4.4.1. The values of the parameters of the system were set as: $m = 1$ kg, the desired variation of the stiffness followed the rule:

$$k(t) = k_0 + sin(\omega t) \tag{4.27}$$

with $k_0 = 4$ N/m, and $\omega = 10$ 1/s (reported in Fig. 4.3a). We did not consider any external force interacting with the system, i.e., $F_{ext} = 0$. We tested both the absence and the presence of damping, with $b = 0$ and $b = 1$ Ns/m.



(a) Without damping $b = 0$.

(b) With damping $b = 1$ Ns/m.

Figure 4.2: Evolution of the energy in the tank in the different cases.

Without Damping

The case without damping is more difficult for the system because it has no element that allows to inject energy in the tank, aside from the variation of the spring, i.e., when the stiffness decreases it injects energy in the tank.

In Fig. 4.2a it can be seen that, without the action of the CBF, the tank's energy goes below the given threshold ε, differently from the case with the CBF active.

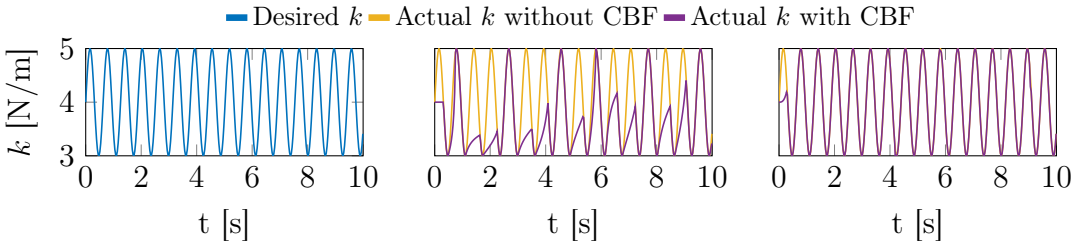
The action of the CBF can be seen in Fig. 4.3b where the evolution of the stiffness of the spring is equal for the desired behavior, which follows (4.27), and for the case without CBF. Instead, with the CBF active, the initial variation is not allowed because it would lead to a value of energy inside the tank lower than the given threshold. The same happens for the following request of increasing k. It is worth noting that the decreasing evolution of k injects energy in the tank, and hence it is always allowed.

With Damping

In this case the tank stores the energy dissipated from the system caused by the presence of the damping, namely $\Xi(x) \neq 0$.

In Fig. 4.2b, the same issue reported for the case without damping can be seen: namely, without CBF, the tank's energy goes below the threshold, while, with the proposed CBF, the passivity condition is respected. It is possible to see also the injection of energy caused by the damping. At the end of the experiment, the injection is not very large because the velocity of the mass is almost zero (the system is over-damped).

Comparing Fig. 4.3c with Fig. 4.3b it is possible to see that the presence of the



(a) Desired $k = k(t)$. (b) Without damping $b = 0$. (c) With damping $b = 1$ Ns/m.

Figure 4.3: Evolution of the stiffness $k = k(t)$ in the different cases.

damping allows a smaller deviation from the desired behavior, always respecting the passivity condition.

Interaction with an Unknown Environment

We report a simulation of interaction with an unknown environment, highlighting the robust stability of the proposed method. The interaction with the unknown environment results as a contribution of the external force $F_{ext} \neq 0$. In particular, it is well known in literature that a system may result unstable if it interacts with a rigid environment, e.g., a wall. For this reason we simulated the system depicted in Fig. 4.4.

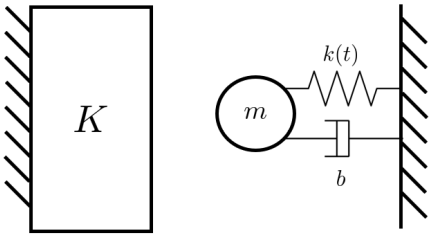


Figure 4.4: Example of the simulation with the unknown environment with high rigidity.

The desired variable stiffness $k(t)$ has a linear behavior:

$$k(t) = k_0 + \omega t$$

with $k_0 = 4$ N/m and $\omega = 10$ 1/s. The wall is modeled as a spring with constant and high stiffness $K = 10000$ N/m and it is located at $\chi = 0.5$ m.

Fig. 4.5 reports the position of the mass in the case with and without the proposed method. It is clear that the proposed method reduce the fluctuations as a consequence of the passivity of the system.

Fig. 4.6 reports the energy inside the tank during the simulation. We can see how the proposed method keeps the energy inside the tank above the threshold ε. Without the CBF, on the other hand, the threshold is passed over multiple times.

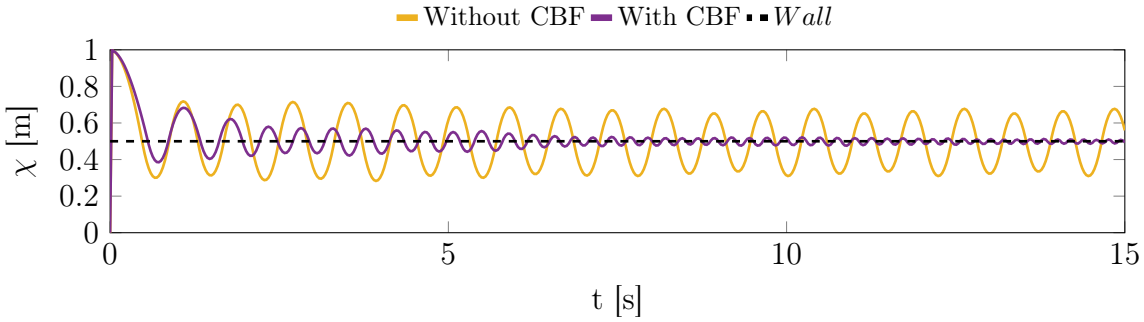


Figure 4.5: Position of the mass in the case with and without the CBF.

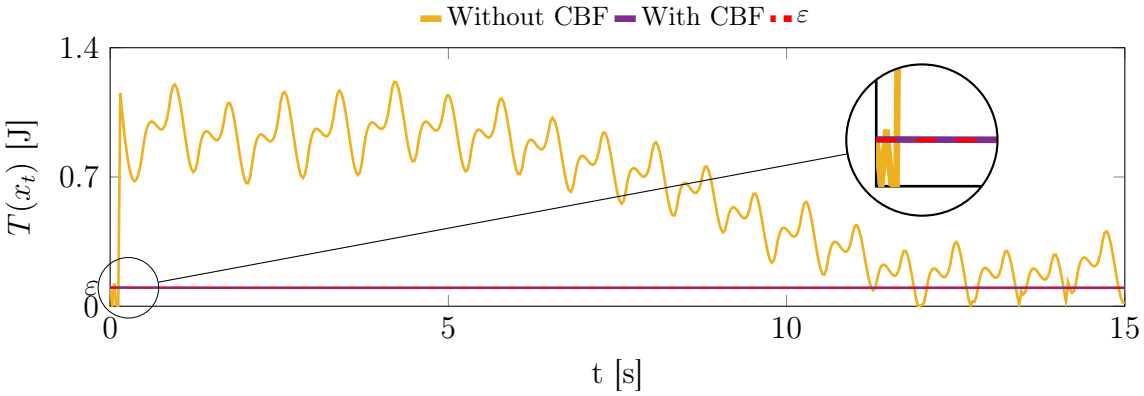


Figure 4.6: Evolution of the energy inside the tank in the case with and without the proposed method.

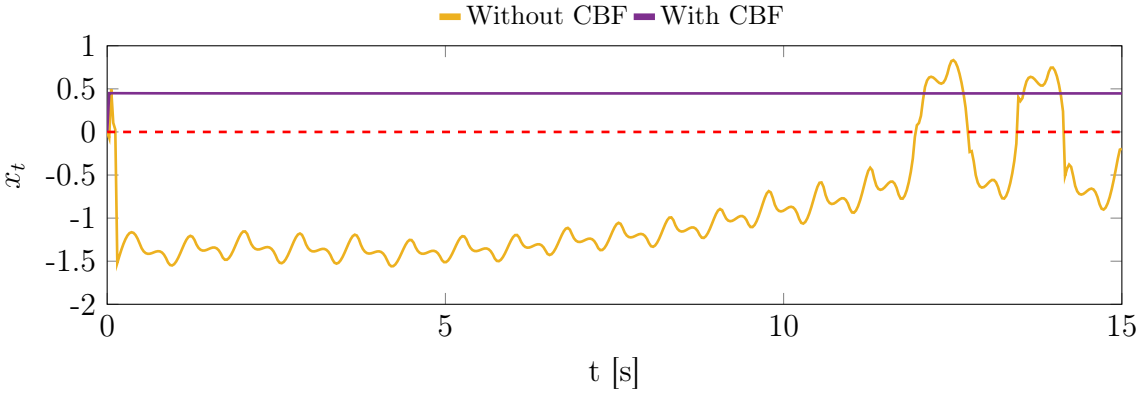


Figure 4.7: Evolution of the state of the tank in the case with and without the CBF. The value $x_t = 0$ is highlighted in *dashed red*.

The influence of the proposed method is even more evident in Fig. 4.7 where the change of sign of x_t happens multiple times in the case without CBF, and it is instead avoided with the proposed method.

Comparison with an Existing Method [117]

In this section we compare the proposed method with the one presented in [117] in the simple case of the time-varying stiffness. Hence, the system is the one described in Section 4.4.1, with the parameters used in Section 4.5.1.

In [117] the QP problem to solve in order to obtain the *passivation objective* is:

$$\begin{aligned} &\underset{\tilde{u},\sigma}{\operatorname{argmin}} \ \|\tilde{u} - u_{des}\|^2 + \kappa|\sigma|^2 \\ &\text{s.t. } \tilde{\Xi}(x) - y^T\tilde{u} + \alpha\left(h(x,t)\right) \geq 0 \end{aligned} \tag{4.28}$$

where $\tilde{u}$ is the input of the system, together with the additional possible external input v ($u = v + \tilde{u}$), σ is the additional damping injected into the system, κ is a *transparency* coefficient, and $\tilde{\Xi}(x) = \Xi(x) + \sigma\|y\|^2$ is the overall energy dissipated by the system. The time-varying CBF proposed in [117] is:

$$h(x,t) = \int_{t_0}^{t} \left(\Xi(x(\tau)) - \tilde{u}(\tau)^T y(\tau)\right) d\tau$$

The desired input for the problem of the time-varying spring is: $\hat{u} = -\frac{k(t)}{m}x$. In the comparison both the tank and the integral of the CBF (4.28) were initialized at the same value. In fact, the integral component of the constraint corresponds to the tank: they both integrate the power dissipated by the system and the exchanged power between the system and the tank.

While both methods lead to preserving passivity, the deviation from the desired input is much larger with the method proposed in [117], with respect to the method proposed in this chapter. This is due to damping injection, which introduces dynamic effects producing large deviations. In fact, as shown in Fig. 4.8, the stiffness obtained with [117] (in green), is very different from the desired one, reported in Fig. 4.3a.

Even though, by changing the value of the *transparency* variable κ, it is possible to obtain an almost perfect implementation of the desired spring (as reported in Fig. 4.8 in light blue), the disturbing dynamic effects introduced by damping injection generate large deviations in the motion of the mass, as shown in Fig. 4.9. Conversely, with the method proposed in this chapter, the deviation from the nominal behavior of the mass is very limited, as shown in Fig. 4.9: this is due to a wiser use of the energy stored in the tank, which does not require injection of additional energy.

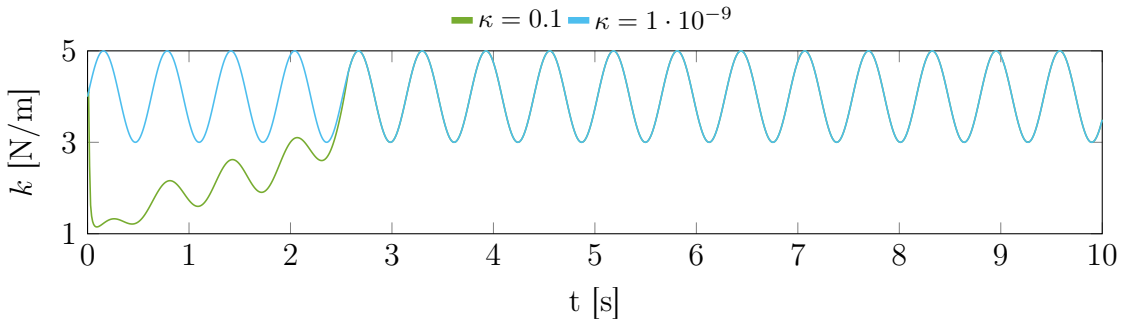


Figure 4.8: Evolution of the stiffness $k = k(t)$ with the method proposed in [117] and with different values of the *transparency* constant κ. With $\kappa = 0.1$ the stiffness is very different from the desired one, reported in Fig. 4.3a. Instead, with $\kappa = 1 \cdot 10^{-9}$ the stiffness tracks the desired one.

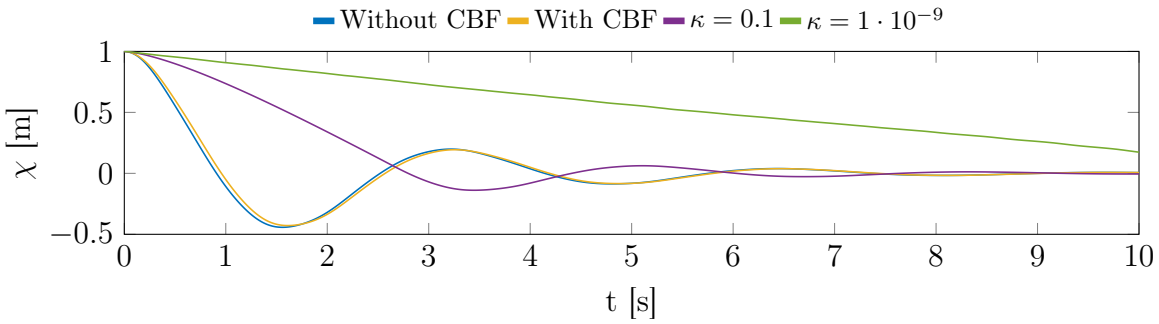


Figure 4.9: Comparison of the position of the mass. "Without CBF" means that $k = k(t)$ is directly applied. "With CBF" represents a trial of the simulation with the proposed method: the mass tracks the desired motion. The other two plots are obtained with the method proposed in [117] with different values of *transparency* κ. In both cases the mass is overdamped and the behavior is very different from the desired one.

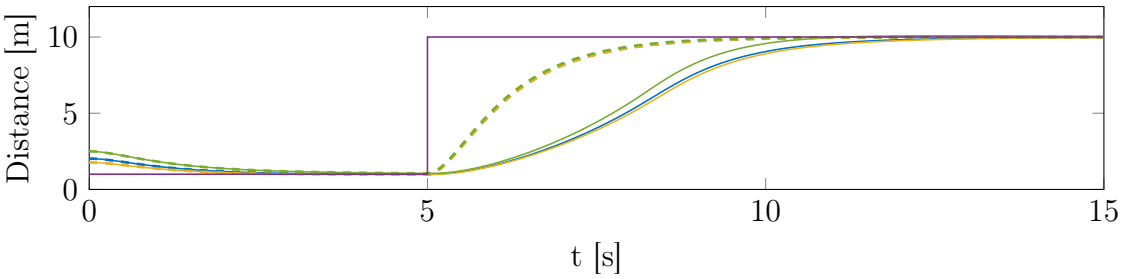


Figure 4.10: Reciprocal distances of the three robots in the case with (*solid line*) and without (*dashed line*) the proposed CBF. In *purple* the desired variation of d_0. In both cases the desired distance is reached.

4.5.2 Case Study 2: Multi-Robot Control

In the multi-robot case, the time-varying part is represented by the desired distance among the robots: in fact, we consider the situation where such desired distance can be changed, to modify the cohesive behavior in the group or to change the shape of the formation.

Simulations

We considered a group of three robots controlled with the potential field V_i described in (4.25). We set the parameters of the system as $M_i = diag\{1, 1\}$ kg, $B_i = diag\{4, 4\}$ Ns/m, $\varepsilon = 0.1$, and we did not consider any external force. For the time-varying potential, we set $k_{att} = 1$ N/m and the desired distance among the robots d_0 was changed from 1 m to 10 m at $t = 5$ s: Fig. 4.10 reports the distance among each pair of robots, compared with the desired distance. It is worth noting that the desired distance values are achieved both with and without the CBF.

The effectiveness of the proposed CBF is evaluated in Fig. 4.11 in which the energy inside the tanks never goes below the given threshold. In Fig. 4.12, the behavior of the state of the tanks is reported, and it is clear how the proposed system prevents the change of sign of x_t, which instead happens if the system is controlled just with the given time-varying potential. This corroborates what stated in Remark 4.3.4.

Real Experiments

In the real experiments we implemented the time-varying potential (4.25) changing the desired distance among the robots in order to obtain a given formation. Matlab was used together with an Optitrack motion capture system, which gave the positions of the robots, and ROS, which allowed to send the input commands to the robots. We used as robots the ePucks, which are small differential drive mobile robots. As the dynamics considered in the simulations was for an omni-directional robot we had to use an input-output state-feedback-linearization in order to calculate the input for the real robots, namely the velocity of the two wheels [81].

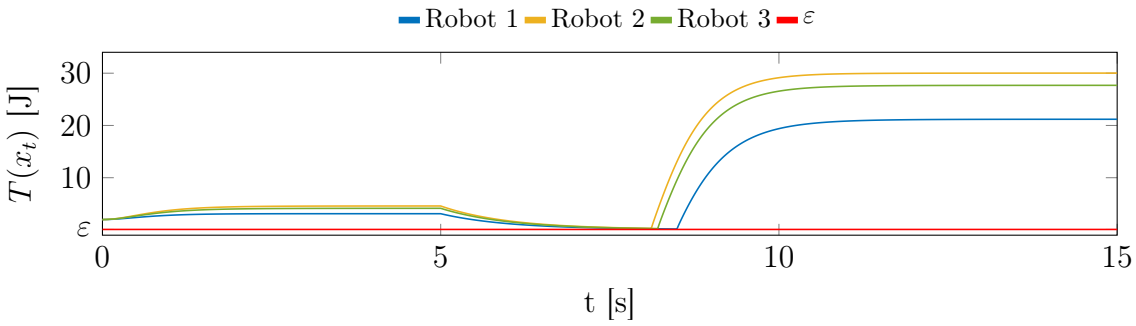


Figure 4.11: Evolution of the energy inside the tanks of the robots. With the proposed CBF we prevent the discharge of the tanks.

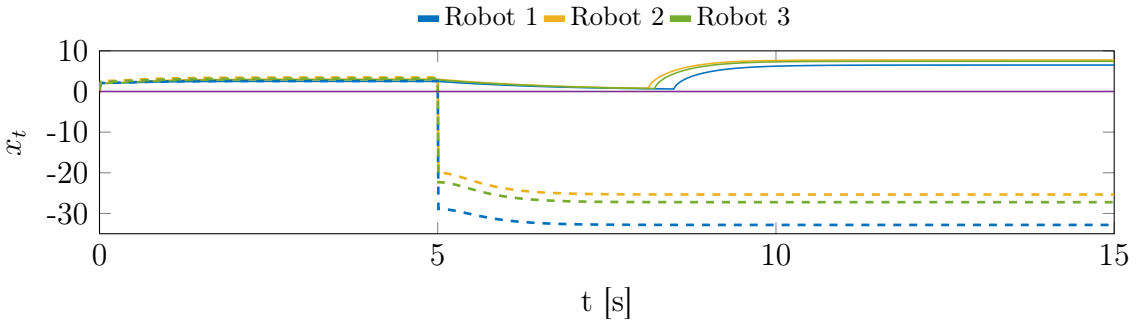
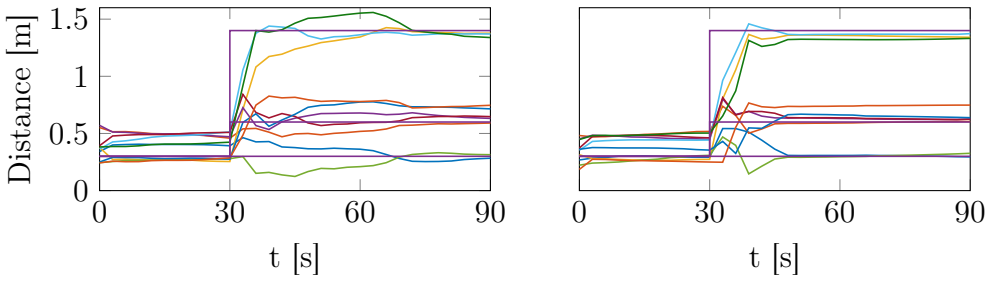


Figure 4.12: Evolution of the state of the tanks in the case with (*solid line*) and without (*dashed line*) the proposed CBF. The value $x_t = 0$ is highlighted in *purple*. It is clear how the proposed method avoids the change of sign of x_t, which happens applying just the desired time-varying potential.



Figure 4.13: Real experiments configuration.



(a) With CBF. (b) Without CBF.

Figure 4.14: Reciprocal distances of the robots in the case with (on the left) and without (on the right) the proposed CBF. In *purple* the desired variation of d_0, which carries to the desired formation reported in Fig. 4.13. In both cases the desired distance is reached.

Fig. 4.13 reports the configuration of the experiments. The values used in the experiments are the following: $N = 5$, $M_i = \text{diag}\{1, 1\}$ kg, $B_i = \text{diag}\{2, 2\}$ Ns/m, and $k_{att} = 2$ N/m.

Fig. 4.14 reports the trend of the reciprocal distances with respect to the desired one in the case with and without the proposed method. We can see that in both cases the desired distances are reached, but the proposed method leads to a slower convergence. This slowness is caused by the proposed CBF that allows to implement just the variation of potential that prevents the draining of the tank. In fact, in Fig. 4.15 we can see the evolution of the energy inside the tanks. Fig. 4.16 reports the behavior of the state of the tanks applying just the time-varying potential and with the proposed method. It is easy to see how the CBF prevents the change of sign of x_t, which would lead to a singularity.

4.6 Conclusions

In this chapter we have proposed a method, based on Control Barrier Functions, to ensure the passivity of a system, changing the desired behavior in a minimally invasive way. We demonstrated that the proposed method preserves passivity, and we have reported both simulations and real experiments to verify it. We tested the method in two different case studies to state the flexibility of the proposed method.

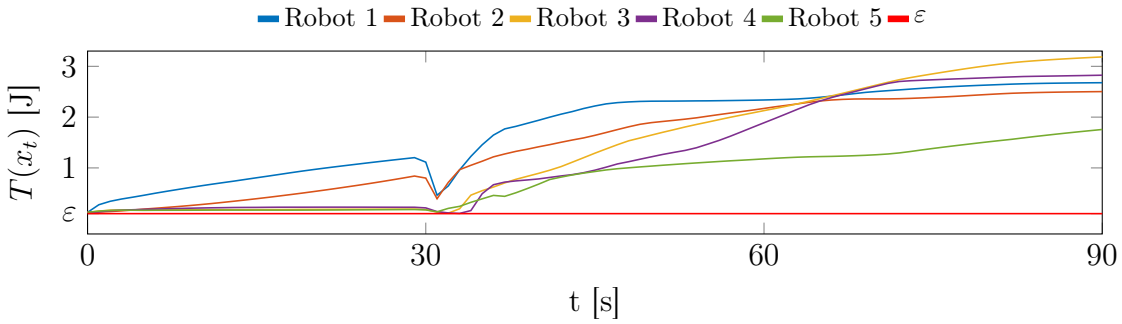


Figure 4.15: Evolution of the energy inside the tanks of the robots. With the proposed CBF we prevent the discharge of the tanks.

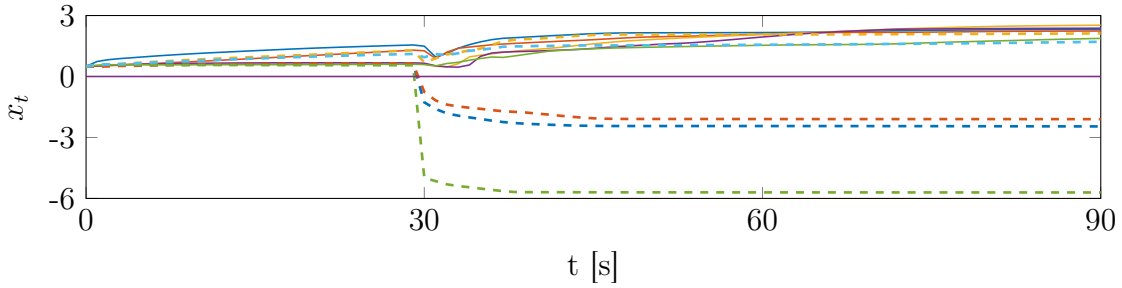


Figure 4.16: Evolution of the state of the tanks in the case with (*solid line*) and without (*dashed line*) the proposed CBF (the same color is used for the same robot). The value $x_t = 0$ is highlighted in *purple*. It is clear how the proposed method avoids the change of sign of x_t, which happens applying just the desired time-varying potential.

Chapter 5

Human-Multi-Robot Interaction: Legibility of Multi-Robot Systems

In this chapter we deal with the topic of legibility of multi-robot systems. Legibility is defined as the ability of a robot, or of a group of robots in the case of this chapter, to communicate its intent to the user. Legibility of multi-robot system is a quite unexplored topic in the literature. However, it will become a very important aspect of the control of this type of systems once they will be more frequent around us. In fact, the communication between the robots and the users will increase importance and every possibilities will be used to improve the efficacy of this collaboration. The ability of communicating without the use of external equipment is very important, as implicit communication is part of our everyday life and it is fundamental to use it.

The contributions of this chapter are:

- Section 5.2: we introduce the definition of legibility for multi-robot systems;
- Section 5.3: we investigate the possibility for a multi-robot system to communicate just with its motion its spatial goal, namely where it is moving towards. We analyzed in a user study if some parameters of the motion of the multi-robot system affect its legibility. We call these parameters *motion variables*, and we found that with a proper control law a multi-robot system can indeed communicate its spatial goal;
- Section 5.4: we further investigate the legibility of multi-robot system considering multiple groups together, and even in this case with a proper tuning it is possible to communicate implicitly with the user;
- Section 5.5: to include more complex information to be communicated, we addressed the possibility of communicating the coordination behavior of a multi-robot system. We identified three major behavior that can represent prototype tasks for these systems. We found out that the users were able to identify the coordination behavior, also simultaneously with the spatial goal.

Hence, the problems addressed in this chapter span from the definition of legibility of a multi-robot system to the study of which type of motion can provide the

highest legibility, both for a single group and multiple groups. We worked also on different type of information that can be communicated, i.e., the spatial goal and the coordination objective of the multi-robot system. Namely, where are the robots going and what are they doing?

5.1 Introduction to Legibility

As robotic platforms become increasingly affordable, reliable, and efficient, multi-robot systems are expected to slowly infiltrate many activities in our everyday lives, blurring the divide between human and robot spaces. The inherent redundancy of multi-robot systems and robotic swarms, afforded by their multiple, and often interchangeable, individual components and their spatial flexibility will likely increase their deployment in a variety of applications, including precision agriculture, logistics, environmental monitoring, or surveillance, among others (see, e.g., [122] for a survey).

While an efficient and safe coordination of the robots is imperative for the fulfillment of each specific task objective, the intersection of robot and human work-spaces demands effective communication among them. The need for such communication, which has become a prominent subject for the interaction between humans and individual robots (see [123], for an example with manipulators), poses further challenges for the multi-robot scenario due do the complexity associated with the coordination among numerous individual robots. This hurdle becomes apparent in both communication directions: how can a human user exert control over the multi-robot system and how can the robots give some information to the human? While some authors have addressed the former question (e.g., [124, 125]), literature addressing the latter is lacking.

As communication is the key to a fruitful interaction between robots and humans, in this chapter we focus on *legibility* of multi-robot systems. In particular, we consider legibility as the ability of a robot, or in this case of a group of robots, to manifest its intention without the use of explicit communication. To this end, this work studies a series of standard multi-robot coordination algorithms and their adequacy in conveying spatial goals of the multi-robot system to a human observer. Furthermore, the different algorithms are analyzed to determine if a human observer is capable of distinguishing the coordination objective of the multi-robot system, and how different parameters that characterize the control algorithm may affect the interpretability of the multi-robot team motions.

5.1.1 Literature Review

Proximal Interaction

When considering Human-Robot Interaction (HRI), a key factor is the differentiation between proximal and remote interaction [126], i.e., the users share the same environment with the robot or they do not, respectively. In applications with remote interaction, the user and the robotic system are in different physical locations [127],

e.g., search and rescue operations [128], aerospace repairs [129], bomb disposals [130], or surgical procedures [131]. This type of interaction is characterized by a user-robot communication by means of a physical instrument, such as a teleoperation tool or a joystick, that allows the user to provide the robot with input commands and, at the same time, allows the user to receive information from the robot, in terms of visual, audio, or force feedback.

Instead, in proximal interaction, the environment is shared by the human user and the robotic platform, and the communication is carried out directly through, e.g., gesture recognition, speech, and gaze engagement. This condition of proximal interaction will be the focus of this chapter. As the robots share the same environment with the user, in addition to communication, we need to take into account the safety of the user. For example, in physical-HRI, the robot, usually a manipulator, has to safely interact with the user, which is achieved by imposing velocity limits as a function of the distance to the user [132]. Even though safety is an issue in proximal interaction applications, in this work we will not address this aspect due to the fact that the user study was carried out in virtual reality, and that the focus was mainly on legibility.

Implicit Communication

From the point of view of the communication, the bi-lateral question still arises: how should the user communicate to the robot and how should the robot itself refer important information back to the user? This work focuses on the latter question. Along the lines of [133], the approaches can be divided into explicit and implicit cues, based on the use of ad-hoc apparatus for communication purposes. For example, explicit cues that have been used in the literature include LEDs [134], light indicators [135], voice [136], gaze movement of a humanoid robot [135], or Augmented Reality (AR) [137, 138, 139]. Although similar to a user interface, AR has the advantage of merging the information directly into the real sight of the user. However, the presence of a tool (e.g., smart glasses or a tablet), places an additional entity in the communication channel: there is no longer direct communication between the robot and the user. Furthermore, using an additional tool may not always be possible, due to task-related constraints (e.g., a tablet can not be used because the user is wearing gloves or is using some tool) or may require some specific training (e.g., the user needs to learn how to use smart glasses). The need for task-specific training also emerges with the use of explicit cues linked to colors or shapes, as some sort of alphabet is needed to match the cues to their meanings. The aforementioned considerations become even more impactful in the case of multi-robot systems: the cognitive complexity, defined in [126] as the amount of cognitive resources necessary for a user to understand and provide input to a multi-robot system, significantly increases with the number of robots. Furthermore, considering the simple mobile robots typically adopted in multi-robot applications, the lack of anthropomorphism impedes the use of cues mimicking human communication. For instance, using voice cues would imply choosing which robot communicates, what information needs to be delivered, and how the communication channel is established.

Implicit cues, conversely, use directly the movement of the robots to convey information, thus leveraging human understanding of movement and the tendency of people to discover intentionality in everything, even in robots [140]. Along these lines, the authors of [141] proposed the use of animation principles to communicate intent in assistive free-flyers. By manipulating the velocities and trajectories of the UAVs, they discovered how to effectively improve communication and that the users preferred motions with ad-hoc modification to infer some significance. The work presented in [123, 142, 143] focused on manipulators and studied their legibility, i.e., their ability to communicate intent through movement, to improve collaborative tasks. In [123] the authors investigated the difference between legibility and predictability, in [142] they studied how to generate legible motions, and in [143] they verified the utility of these legible motions into a collaborative task. In their studies the motion of the robot is generated by means of a planner that optimizes legibility itself. We took inspiration from their definition of legibility to proceed in our study towards multi-robot applications. Hence, preliminary works were presented in [144, 145, 146], where the characteristics of the motion of a multi-robot system (see Section 5.1.1 for more details) were directly studied. However, no planner to generate legible motions was defined. Furthermore, high-level coordination objectives were not considered in these previous works, and are explicitly addressed in Section 5.5. Such high-level objectives may be linked to the studies on the communication of robots' plan, which is a topic studied in the Artificial Intelligence (AI) field [139, 147, 148]. In this field, in particular in the context of Human Machine Interface (HMI) and with mechanical interfaces, such as in teleoperation tasks, legibility is often referred to as *transparency*. In [149] a detailed description of the differences and of the implementations of legibility and transparency in AI can be retrieved.

Legibility in Multi-Robot Systems

In the context of social applications, robots must effectively interact with people in an engaging manner [150], a requirement that oftentimes demands expressing emotion through their actions. In robotic swarms and multi-robot systems composed by simple, non-anthropomorphic robots, conveying emotions to an observer can pose an increased difficulty. In these cases, the emotional expressivity and readability of the movement of the robotic swarm has been linked to different parameters related to the individual control of the robots in the swarm and the inter-robot interactions [151, 152], under the context of standard collective behaviors such as cyclic pursuit or coverage [153] or relating them to dancing [154]. Building upon these ideas, in this chapter we investigate the influence of the choice of control parameters on the capability, for the user, to infer spatial and coordination objectives of the multi-robot systems.

Legibility of a multi-robot system was introduced in our previous works [144, 145] as the ability of the multi-robot system to communicate to the user its spatial goal. Such ability was characterized based on two performance indices: 1) the correctness of the communication and 2) the time needed for this communication. The first

question to answer was whether the motion of a multi-robot system was able to communicate some information to the user. We investigated this issue in our previous work [144], where we found out that some *motion-variables* (namely the trajectory, the dispersion and the stiffness of the multi-robot system) were statistically relevant for the communication of the spatial goal of the group. The *motion-variables* were chosen to describe some important characteristics of the multi-robot system. The user study was conducted in a virtual reality set-up where the user was able to share the same environment with the robots. The kind of controller used to define the movement of the robots was based on artificial potential fields.

The following step was to understand if the same *motion-variables* could be used to communicate the same information, i.e., the spatial goal, of multiple groups of robots at the same time [145]. The results of this study were similar to those of the previous user study and the *motion-variables* confirmed to be statistically relevant with respect to the legibility of a multi-robot system.

In [155] the legibility of multi-robot systems was further investigated, together with glanceability, defined as the ability to communicate a spatial goal throughout the whole movement of the robots. The authors focused on different types of control laws, similar to those implemented in this study, but they investigated also the effect of target difficulty, namely whether the spatial goals had the same number of neighboring goals or not, and initial radius, namely how much the robots were spread out at the beginning of the experiment. Only the legibility with respect to the spatial goal was analyzed, with rendezvous (i.e., consensus) being the most legible controller and the trajectory-based motion the best for glanceability. The user studies were carried out on a standard screen and with a top-down view that is not suitable with our idea of proximal interaction. In fact, the user studies reported in this chapter were conducted in a virtual reality setup with the objective of imitating as closely as possible a proximal interaction scenario.

5.2 Definition of Legibility in Multi-Robot Systems

What does legibility mean for a multi-robot system? Can we define it in terms of *spatial goal* (where are the robots going) and *coordination objective* (what are the robots doing)? These can be useful when sharing the space with the user. Legible motion endows robots with the ability to communicate information directly, without the need of a specific user interface. While user interfaces continue to be the standard in many human-robot interaction scenarios, here we want to understand if it is possible to communicate implicitly this information. The prototype coordination objectives we choose to investigate are: consensus, formation, and coverage. We chose these controllers due to their ubiquitous use in multi-robot systems [83].

Given that the legibility of multi-robot systems is still in its early stages, we need to understand what legibility means in this context. We consider legibility as the ability of a group of robots to communicate to the user its spatial goal and/or its coordination objective. Hence, we propose the following definition:

Definition 5.2.1. *A multi-robot is legible if it has the ability to communicate to the user (without the use of explicit methods):*

- *Its* spatial goal*, namely the overall direction of motion of the multi-robot system;*
- *Its* coordination objective*, namely how robots are moving with respect to each other and how they are aggregating.*

We will investigate the legibility of multi-robot systems in terms of *spatial goal* in Section 5.3, 5.4, and in the first part of Section 5.5. Instead, we deal with the legibility in terms of *coordination objectives* in Section 5.5.

5.3 Single Group Legibility

In this chapter we focus on the legibility of a group of robots. We aim at understanding whether a multi-robot system is able to communicate its spatial goal, and what are the parameters within the motion of the group that affect this ability. The work reported in this section has been published in [144].

5.3.1 Methods

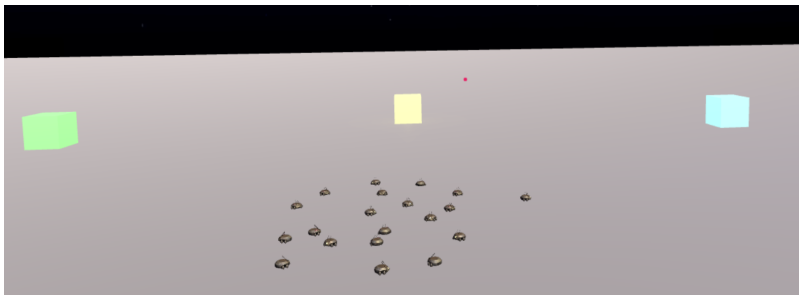
The experiments aimed at understanding whether these *motion-variables* significantly affect the legibility of a multi-robot system.

Aim of the Experiment

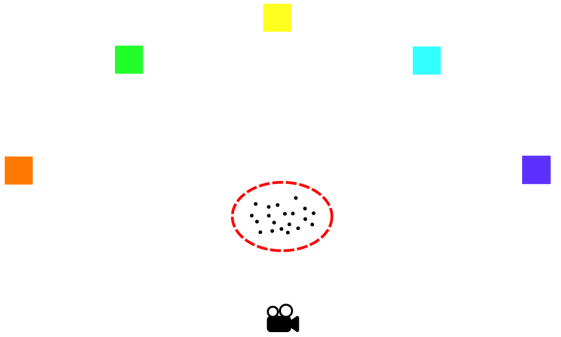
During the experiment, a group of mobile robots traveled towards random goals with different types of movement, based on the values of the *motion-variables*. To evaluate the legibility of the multi-robot system, the user was asked to indicate, as soon as possible, the goal that she/he thought as the most likely. The aim was then to answer in the correct way as soon as possible.

Experimental Scenario

To collect the data, we built a virtual reality set-up where the user shared the space with a group of robots (Fig. 5.1a). We chose virtual reality because it is possible to build a wide environment and to use a large number of robots, which are characteristics very difficult to obtain in a real set-up. In addition, virtual reality ensures higher repeatability with respect to a real scenario because it does not introduce localization errors, slippage problems or other disturbing factors that are not easy to model. As a consequence, in our opinion, the use of virtual reality represents a significant improvement over experiments presented in [141] and [156], where the movement of robots was shown to users only through an off-line video or a sequence of photos. On the contrary, virtual reality provides an immersive real-time interaction that reproduces real environments faithfully and allows us to replicate the interaction experience, as proved in our previous work [157].



(a) Example of the user's view during the experiments.



(b) Disposition of the goals (*colored cubes*) and of the robots (*circled in red*) with respect to the user (*camera icon*).



(c) Oculus Remote and the button used (*circled in red*).

Figure 5.1: Virtual reality environment.

The environment was developed in Unity, which is a cross-platform game-engine, and the experiments were reproduced with the Oculus Rift headset. The set-up consisted in a big area, approximately with the dimension of a tennis court (22×11 m), populated with some objects that represented the possible goals for the multi-robot system. The objects were cubes that emitted light of different colors and they were purposely designed with a very simple shape, because they did not have to confuse or distract the user during the experiment. The same goes for the rest of the environment: minimal background and no sound. All the focus of the user was moved towards the object of the experiment. The multi-robot system was made up of twenty omnidirectional robots, with a cylindrical shape and a radius equal to 0.1 m.

We chose not to register a video, as in [141], or to make a series of photos, like in [156], because the focus was about proximal interaction and we thought that virtual reality was a better means for this purpose.

The input used during the experiment was the Oculus Remote (Fig. 5.1c) combined with the gaze of the user. This solution was chosen because it is very intuitive and fast. In fact the user just had to look towards the object and press a button. For example, a keyboard or a joystick could have introduced some errors due to the fact that the user could not see the input device. Instead, the Oculus Remote has a very simple layout and the button used for the experiment is very difficult to miss.

Factorial Experiment

In general, when a series of experiments are used to understand the influence of some variables, called independent variables, over the response, which can be represented by multiple dependent variables, a factorial plan is built [158].

Table 5.1: Factorial plan.

	$\mathcal{T}$	$\mathcal{D}$	$\mathcal{S}$	$\mathcal{TD}$	$\mathcal{TS}$	$\mathcal{DS}$	$\mathcal{TDS}$
1	-	-	-	+	+	+	-
2	+	-	-	-	-	+	+
3	-	+	-	-	+	-	+
4	+	+	-	+	-	-	-
5	-	-	+	+	-	-	+
6	+	-	+	-	+	-	-
7	-	+	+	-	-	+	-
8	+	+	+	+	+	+	+

For these experiments, we had three independent variables (trajectory, dispersion and stiffness) and, for every variable, we chose to investigate two levels. In order to study the effect of each variables and of the interactions between them, the plan

must be fully investigated (2^3 combinations). Table 5.1 reports the full factorial plan that we used for the experiments. The number in the first column represents each trial: as there are three variables and each variable has two possible value (+ or -), we have 2^3 combinations. The columns in blue report all the combinations of the three independent variables (main-effects) and define, for each trial, the values that must be set. The remaining columns represent the two-way and three-way interactions. They are the additional variables created by considering, respectively, two and three main-effects at the same time.

To avoid the influence of disturbing factors, some precautions must be taken during the experiments. These disturbing factors must be controlled as much as possible or they will influence the response. We identified the factors that we thought could affect the experiments and we tried to reduce their effects:

Point of View

In a virtual reality set-up, the user is immersed in the environment and she/he can look around in every direction and, in some applications (like in video-games), she/he can also move inside the space. In this particular case, we chose to fix the position in the middle of the goals and to allow the user only to look around. In fact, if the user had the possibility to move around, this would have introduced too much variability. The movement of the head is a good trade off between the full freedom and a completely fixed point of view. Furthermore, the movement of the head is fundamental to avoid the motion sickness caused by the decoupling of the sense of sight and the sense of balance [159].

The position of the user with respect to the robots was like she/he was standing among them. This position is neutral in such a way that the user did not feel neither inferior nor superior towards the robots [156]. Furthermore, it is the most likely position for a collaborative proximal interaction task.

Position of the Goals

The goals were positioned equally spaced in front of the user. This limited the area of interest for the user and, in this way, it limited the need of looking around, also improving the previous issue (point of view) (Fig. 5.1b). The goal in every trial was randomized.

Differently from [142], we introduced more than two goals because, in general, in a collaborative task with a multi-robot system it is more frequent to have multiple goals, given the intrinsic multiplicity of the system. In addition, we believe that inferring a goal that is surrounded by others is more difficult and, hence, this would be a more generic study. For this purpose, the possible goals were reduced to the three central, because they were equally distributed, namely they all had two neighbors.

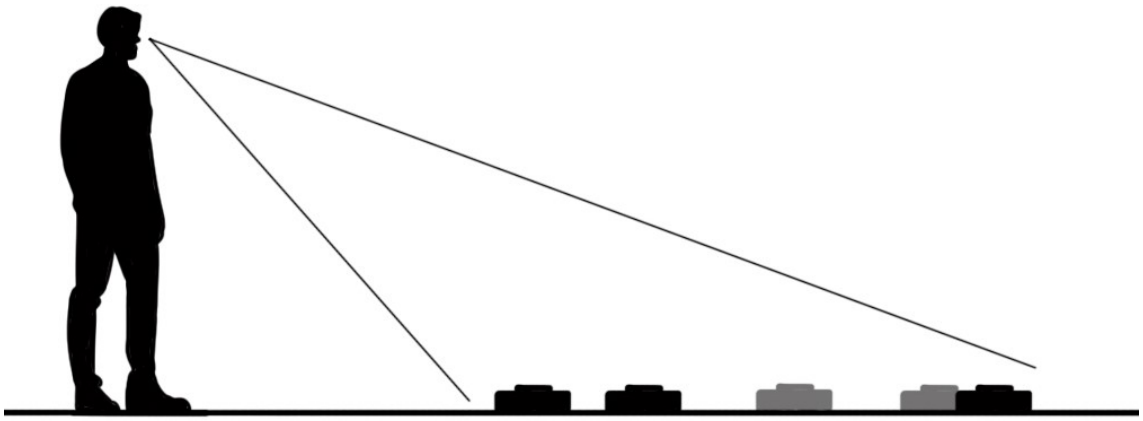


Figure 5.2: Point of view of the user (height approximately 1.7 m). It replicates the proximal interaction scenario in which the user is working alongside the robots.

Learning Effect

In order to limit the impact of this factor, every user received a brief training about the input method. In addition, everyone had a trial at the beginning, where she/he tried to interact with some objects in the same way they were supposed to interact with the cubes (representing the goals). The order of the trials was randomized to reduce the influence of the learning effect.

Movements of the Robots

The trajectories of the robots were saved offline and they were reproduced exactly in every experiment. The repeatability of the motions of the robots is fundamental, since the core of the study was about the ability of the robots to communicate through their motion.

Independent Variables

We describe the state of the system through three *motion-variables*:

- $\mathcal{T}$: trajectory of the center of the group;
- $\mathcal{D}$: dispersion of the group;
- $\mathcal{S}$: stiffness of the group.

These are only some of the possible variables that can be chosen to describe a multi-robot system. We choose trajectory because it is fundamental for a group of moving robots and, in general, for every type of robot. Moreover, trajectory is also a parameter already analyzed in other studies about legibility [123], [142], [143]. On the other hand, considering the common control architecture based on artificial

potential fields [118], dispersion and stiffness are parameters used to define the motion characteristics of a multi-robot system. In particular, dispersion represents the desired distance between the robots and stiffness is the strength of the potential.

Hence, the *motion-variables* (trajectory ($\mathcal{T}$), dispersion ($\mathcal{D}$), and stiffness ($\mathcal{S}$)) are the independent variables of the study. They are defined in two levels, which have been chosen sufficiently distant to recognize the difference between them, as detailed below.

The multi-robot system is composed by N mobile robots, each one modeled as a dynamic system in a n-dimensional space:

$$M_i \ddot{x}_i = w_i \quad i = 1, \ldots, N \tag{5.1}$$

where $M_i \in \mathbb{R}^{n \times n}$, positive definite, and $x_i \in \mathbb{R}^n$ are the inertia matrix and the position of the i-th robot, respectively. The control input and all the external forces that act on the robot are included in the term $w_i \in \mathbb{R}^n$. For simplicity of notation, we will hereafter consider $M_i = m\mathbb{I}$, where $\mathbb{I} \in \mathbb{R}^{n \times n}$ is the identity matrix.

The control architecture is based on a potential field, which allows us to move a group of robots with a cohesive behavior and to avoid collisions. At the same time, it enables to drive the multi-robot system through an environment, following a virtual agent that moves along a desired trajectory.

Trajectory

We chose two of the infinite possible trajectories that link two points in a n-dimensional space. The trajectory specifies the position of the virtual agent in each instant of time.

The first is a *minimum-jerk* trajectory, namely a trajectory that minimizes the square of the magnitude of the jerk (rate of change of acceleration). We chose to investigate this trajectory because it is the typical movement that a human does while grasping an object [160]. Furthermore, in [161] it has been discovered that, for a user, it is easier to interact with a manipulator that moves with a *minimum-jerk* trajectory instead of a trapezoidal one. However, in standard industrial applications with robotic arms, the trapezoidal velocity profile is preferred [68]. To the best of the authors' knowledge, no works can be found about the correlation between *minimum-jerk* trajectory and legibility of a multi-robot system. If we consider a point that follows a one-dimensional trajectory $\varphi(t)$ from a start point $\varphi(t) = \varphi(0)$ to a final point $\varphi(t) = \varphi(f)$, with $f > 0$, its position is described, in every instant of time t, by:

$$\varphi(t) = \varphi(0) + (\varphi(f) - \varphi(0)) \left(10 \left(\frac{t}{f} \right)^3 - 15 \left(\frac{t}{f} \right)^4 + 6 \left(\frac{t}{f} \right)^5 \right) \tag{5.2}$$

This can be easily adapted to a n-dimensional trajectory, applying (5.2) componentwise.

The second trajectory, called *arc-trapezoidal*, follows the animation principles of "Slow In and Ou" and "Arc" from [162]. These principles have already been investigated in [141] for "assistive free-flyers", but not yet for multi-robot system.

The trajectory is drawn with an arc of circle and with a trapezoidal velocity profile. The direction, counter-clockwise or clockwise, is random.

Both trajectories lead the virtual agent from the starting point to the desired goal in 60 seconds. The time interval is chosen in such a way that the overall experiment can not last more than 10 minutes to avoid discomfort to the user for the use of the head-mounted display.

Dispersion and Stiffness

While the virtual agent follows the given trajectory, the real robots are linked to the virtual agent and among each other with a potential field and a damping element. If we consider the i-th robot, we can rewrite (5.1) as:

$$M_i\ddot{x}_i = -B_i\dot{x}_i - \nabla V_i \tag{5.3}$$

where $B_i \in \mathbb{R}^{n\times n}$, positive definite, is the damping factor, which must be introduced for a smooth movement, and $\dot{x}_i \in \mathbb{R}^n$ is the velocity of the i-th robot. We consider a homogeneous friction coefficient along all directions, hence $B_i = b\mathbb{I}$, with $b \in \mathbb{R}_{\geq 0}$.

The term ∇V_i is the gradient of the potential field V_i applied to the i-th robot. The potential field V_i is described by[1]:

$$V_i = \sum_{j=1;j\neq i}^{N} V_{a_{i,j}}\left(x_i, x_j\right) + \sum_{j=1;j\neq i}^{N} V_{rep_{i,j}}\left(x_i, x_j\right) + V_{v_i}\left(x_i, x_v\right) \tag{5.4}$$

where $V_{a_{i,j}}$ is the cohesive part of the potential and is equal to:

$$V_{a_{i,j}} = \begin{cases} \frac{1}{2}K_{a_{i,j}}\left(d_{ij} - d_0\right)^2 & \text{if } d_{ij} \leq d_1 \\ 0 & \text{otherwise} \end{cases} \tag{5.5}$$

and $d_{ij} = \|x_i - x_j\|$ is the Euclidean distance between the i-th and the j-th robot.

Then $V_{rep_{i,j}}$ is the repulsive component, whose purpose is to avoid collisions between the robots:

$$V_{rep_{i,j}} = \begin{cases} \frac{1}{2}K_{rep_{i,j}}\left(d_{ij} - d_{min}\right)^2 & \text{if } d_{ij} \leq d_{min} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

At the end, V_{v_i} is the attractive potential between the i-th robot and the virtual agent:

$$V_{v_i} = \begin{cases} \frac{1}{2}K_{v_{i,v}}\left(d_{iv} - d_{0_v}\right)^2 & \text{if } d_{iv} \leq d_{1_v} \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

All the potentials are defined with a positive constant, respectively $K_{a_{i,j}}$, $K_{rep_{i,j}}$ and $K_{v_{i,v}}$. Their action is different from zero only in a limited area near the i-th robot, which is represented by the terms d_1, d_{min} and d_{1_v}.

[1] In the following formulation we omit the dependence on time of the variables, e.g., $V_i\left(t\right) = V_i$, for ease of notation.

The independent variables dispersion and stiffness are defined by the parameters of the potential field V_i. In particular, dispersion is represented by the distance d_0, that is the desired distance between the robots. The level *Small* (-) requires the group to maintain a distance ten times the dimension of the single robot (≈ 0.1 m of diameter), while the level *Large* (+) thirty times. The stiffness parameter defines the dynamic behavior of the multi-robot system, namely the constants that state the strength of the potential field ($K_{a_{i,j}}$, $K_{rep_{i,j}}$), and the values of mass and damping (m and b are the scalar values inside the diagonal elements of M_i and B_i). These parameters were tuned empirically after preliminary experiments and are reported in Table 5.2. Table 5.3 reports the other parameters, which are equal for all the levels of both variables.

Table 5.2: Dispersion and stiffness levels.

		−	+
d_0	[m]	1	3
$K_{a_{i,j}}$	[N/m]	0.005	0.5
$K_{rep_{i,j}}$	[N/m]	10	1000
m	[kg]	0.1	10
b	[Ns/m]	0.01	1

Table 5.3: Constant variables.

$K_{v_{i,v}}$	[N/m]	0.4
d_1	[m]	5
d_{1_v}	[m]	5
d_{min}	[m]	0.5
d_{0_v}	[m]	0.1

The movement of the robots has been saved for every combination of the independent variables and for every possible goal (24 possible combinations), and Fig. 5.3 reports some examples. These positions have been then used directly in the Unity program to replicate the exact behavior.

Dependent Variables

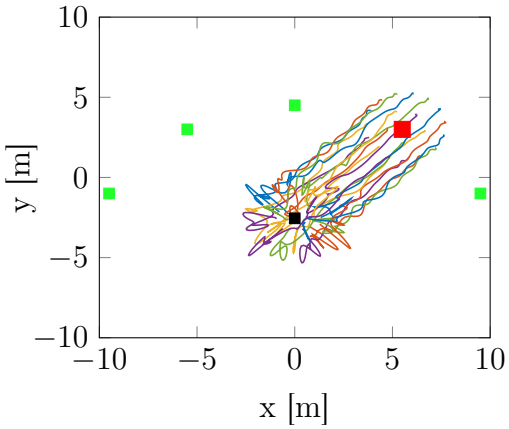
The dependent variables are the correctness of the answer and the time the user took to answer. We chose to evaluate both the variables because, in a collaborative task, it is important to understand the intentions of the robots, but also to understand them in a reasonable amount of time.

Statistical Analysis

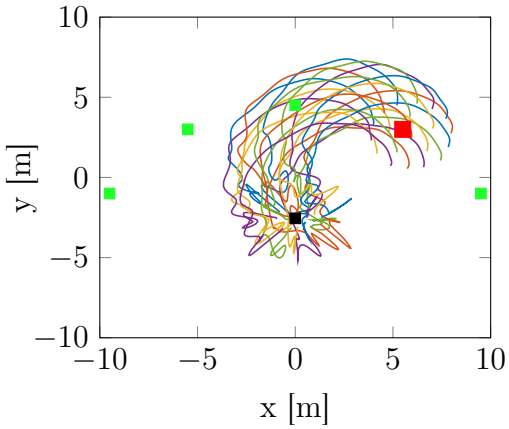
In order to evaluate the statistical relevance of the *motion-variables* upon legibility of the system, a statistical analysis was carried out. In particular, Fisher's exact test and 2-way ANOVA were conducted to test the null hypothesis, respectively for the correctness and the time of the answers:

H0 - Null hypothesis: *there is no relation between the motion-variables and the legibility of the system (correctness and response time).*

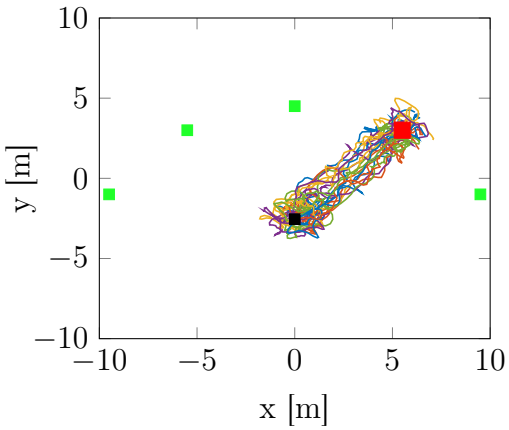
The Fisher's exact test allows us to investigate the presence or absence of random association between two categorical variables, while the 2-way ANOVA allows us to analyze the effect of two variables over a dependent variable that, as null hypothesis,



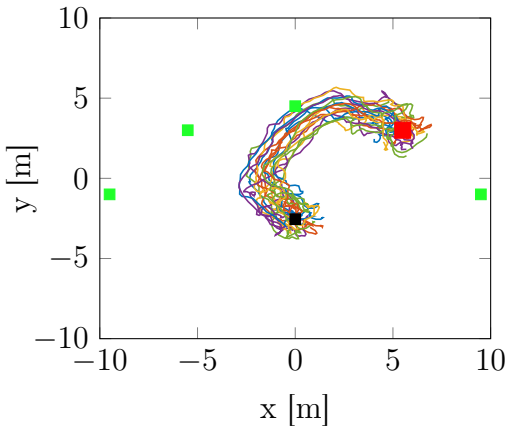
(a) $\mathcal{T}$ =*Minimum-jerk*, $\mathcal{D}$ =*Large*, $\mathcal{S}$ =*Hard*.



(b) $\mathcal{T}$ =*Arc-trapezoidal*, $\mathcal{D}$ =*Large*, $\mathcal{S}$ =*Hard*.



(c) $\mathcal{T}$ =*Minimum-jerk*, $\mathcal{D}$ =*Small*, $\mathcal{S}$ =*Soft*.



(d) $\mathcal{T}$ =*Arc-trapezoidal*, $\mathcal{D}$ =*Small*, $\mathcal{S}$ =*Soft*.

Figure 5.3: Some examples of robots' movement in different trials: each line represents the trajectory of a robot. In *black* the starting point, in *red* the chosen goal and in *green* the other goals.

can be approximated by a random distribution (namely, it can assume continuous values and it is not categorical).

Users

The experiment was replicated by 30 users (age 26.76 ± 5.2, 8 females and 22 males). They are students and researchers of our engineering department, but not everyone was from the robotic field because we wanted to evaluate this interaction method also for subjects that are not used to work with robots. All enrolled subjects were new to the experiment. Before starting the experiment they signed a consent form and they were trained about the input method and the goal of the experiment. In particular, they knew that: 1) a series of trials with different characteristics would be performed by a group of mobile robots, 2) the goals were all equally probable (even thought, as stated in Section 5.3.1, the external ones were never selected), and 3) all the robots moved towards a goal, never changing their destination.

According to the within-subjects experimental methodology [163], every user performed all the eight possible combinations of the independent variables and she/he was asked to comment on the movement of the robots or to communicate any other relevant remark. This technique is called "think aloud" and it is frequently used in usability tests to retrieve additional information during the experiments, such as possible issues or solutions never addressed by the authors [164, 165].

It is worth to say that most of the users had already tried virtual reality, only one user stated that he never used it. For this reason we may suppose that the result are not influenced by the use of virtual reality as means for this experiment.

5.3.2 Results

In the following, we report the results of the experiments in terms of correctness and response time, which are the dependent variables under analysis.

Correctness

Correctness is a categorical variable as it can assume only two values, i.e., right or wrong. Hence, for every independent variable, a Fisher's exact test was carried out. The first step was to build the contingency table that collected the answers with respect to each independent variable. As an example, Table 5.4 reports the contingency table of the trajectory variable.

From the contingency table, it is possible to approximate the p-value based on the chi-squared distribution. In order to obtain a more precise result, we decided to calculate the exact p-value using the Fisher's exact test, whose results are reported in Table 5.5. If the p-value is smaller than 0.05, which corresponds to a significance level of 5%, we can state that the variable is statistically relevant. Hence, according to the results in Table 5.5, trajectory is the only independent variable that affects the correctness of the answer. The test had been replicated also with a significance level equal to 1% and the variable remains significant. This means that we can

Table 5.4: Contingency table of correctness of answers with respect to trajectory variable.

Trajectory \ Correctness	Wrong	Right	Total
Minimum-jerk	11	109	120
Arc-trapezoidal	115	5	120
Total	126	114	240

claim, with a confidence equal to 99%, that the trajectory variable influences the legibility of a multi-robot system.

Table 5.5: p-value from the Fisher's exact test.

Independent variable	p-value
Trajectory	< 0.0001
Dispersion	0.8972
Stiffness	0.5182

From the Fisher's exact test we can state that the choice of the trajectory influences the ability of the multi-robot system to communicate its goal to the user. Indeed, it is legitimate to ask which trajectory is more suitable for this type of communication. A preliminary answer can be retrieved from Table 5.4, where it is clear that the *minimum-jerk* trajectory outperforms the *arc-trapezoidal* one in terms of correctness of the answer. In our opinion, this result is caused by the fact that the *arc-trapezoidal* trajectory tricks the user because, at the beginning of the movement, the group of robots moves towards a different goal, so often the user thinks that the goal is the wrong one (see Fig. 5.3a with respect to Fig. 5.3b, which differ only by the trajectory variable). Conversely, the *minimum-jerk* trajectory is straightforward from the starting point to the goal. Hence, we believe that the results depend on the geometrical trajectory and not necessarily on the velocity profile, which is less noticeable by the user. This came out from the think-aloud process: a lot of users stated that they chose the first goal towards which the group was moving, but no one reported any comment about the velocity of the robots.

Response Time

To analyze the dependent variable of the response time, a 2-way ANOVA has been performed. We considered only the right answers and we did not apply any penalty function because the core of the experiment was that of identifying the legibility. Applying a penalty function would have meant to change the results, namely add an arbitrary amount of time to the wrong answers. In general, penalty function is more suitable for a task oriented experiment, where giving a wrong answer will lead to a worst performance of the overall interaction. Instead, these trials aimed at

investigating whether the independent variables influence legibility and modifying the data will not result in a correct analysis.

We verified that the new factorial plan was balanced between the cases: if this was not verified, the test would be distorted. The ANOVA was then conducted only about dispersion and stiffness because the trajectory had a very unbalanced factorial plan, since most of the right answer were with the *minimum-jerk* trajectory. Hence, for this analysis, Table 5.1 is reduced removing all the rows with the value *arc-trapezoidal* (+) in the column of the trajectory variable ($\mathcal{T}$). Not considering the trajectory, the two-way and three-way interactions that combine this main-effect lose their meaning. Hence, also the columns of the interactions variables $\mathcal{TD}$, $\mathcal{TS}$ and $\mathcal{TDS}$ are removed from the factorial plan.

The ANOVA, similarly to the Fisher's exact test, allows us to calculate a p-value that represents the smallest value of significance with which we can reject the null hypothesis. In addition ANOVA can also investigate if any interaction exists between two variables (namely, the dependent variable depends also on the relation between the two independent variables)[2].

Table 5.6: p-value from the 2-way ANOVA.

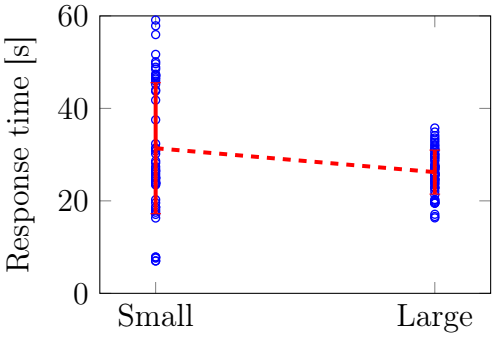
Independent variable	p-value
Dispersion	0.0002
Stiffness	< 0.0001
Dispersion-Stiffness	< 0.0001

From Table 5.6 we can state that both dispersion and stiffness are significant for the response time and also their interaction.

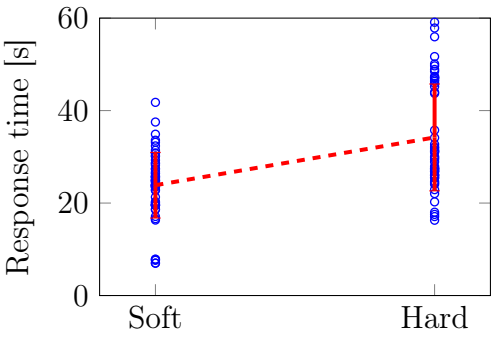
Fig. 5.4a reports the response time of the experiments (in blue) and the mean (μ_T) and the standard deviation (σ_T) of both levels of the dispersion variable (in red). Apparently, having a bigger dispersion improves legibility, which is a counter-intuitive result. However, from the think aloud we discovered that, in the trials with high dispersion, some users felt confused because, at the beginning, the robots seemed to move in opposite directions.

Fig. 5.4b reports the results for the stiffness: also in this case the result is partially counter-intuitive, because the best results seem to be produced in the case of a system with low stiffness. In general, we might think that a high value of stiffness produces an almost ordered formation, while a low value would lead to a more chaotic motion. However, it is worth noting that a high value of stiffness generates large control input on the robots, and hence large acceleration. This leads to a sudden big variation in the geometric shape of the multi-robot system, which probably causes confusion to the user.

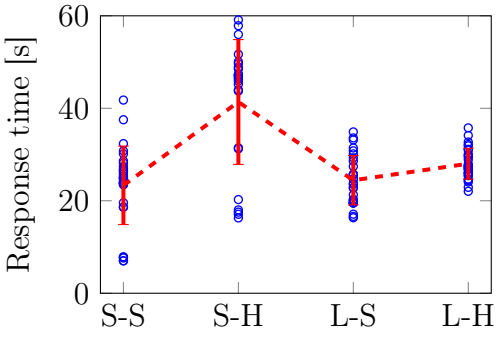
[2]For example, if we want to consider the frequency of ordination at a restaurant and we consider as independent variables: food (pasta or ice-cream) and topping (tomato or chocolate). Probably the food and the topping would not have a significant relevance, meaning that on average both kinds of food are ordered with the same frequency, but their interaction would be very significant, namely no one will order pasta and chocolate or ice-cream with tomato.



(a) Dispersion level:
Small: $\mu_T = 31.4$, $\sigma_T = 14.1$
Large: $\mu_T = 26.2$, $\sigma_T = 4.8$



(b) Stiffness level:
Soft: $\mu_T = 23.8$, $\sigma_T = 7.1$
Hard: $\mu_T = 34.2$, $\sigma_T = 11.5$



(c) Dispersion-Stiffness level:
Small-Soft: $\mu_T = 23.3$, $\sigma_T = 8.5$
Small-Hard: $\mu_T = 41.4$, $\sigma_T = 13.5$
Large-Soft: $\mu_T = 24.4$, $\sigma_T = 5.3$
Large-Hard: $\mu_T = 28.0$, $\sigma_T = 3.3$

Figure 5.4: Response time analysis (in blue scatter plot of the trials, in red the mean (μ_T) and the standard deviation (σ_T) of each level).

The results for the interaction variable (Dispersion-Stiffness, $\mathcal{DS}$) can be retrieved in Fig. 5.4c. The combination with the lowest response time is *Small-Soft*. It is worth noting that the best combination considering the results of the analysis of single variables, namely *Large-Soft*, has also a very low mean value. However, it is important to consider the interaction $\mathcal{DS}$ because, as can been seen in Fig. 5.4c, the *Small* value of dispersion is significantly affected by the different value of the stiffness variable ($\mu_T = 23.3$ for *Small-Soft* and $\mu_T = 41.4$ for *Small-Hard*), and the same holds for the *Hard* value of stiffness.

Moreover, most of the users with the combination *Large-Hard* moved backwards because they felt like the robots were moving towards them and they found the initial dispersion of the robots very confusing. Thus, on the one side, this anecdote confirms the fact that virtual reality is a very effective means for this type of experiments. On the other side, it highlights the necessity for such type of interaction to consider the user as part of the multi-robot system and adapt the behavior consequently [166].

These results allow us to establish, with a sufficient degree of certainty, that

the chosen *motion-variables* influence the legibility of a multi-robot system. In particular, the preliminary analysis presented in this paper shows that the *minimum-jerk* trajectory and the values *Small-Soft* for dispersion and stiffness return the highest correctness and the lowest response time, thus increasing the legibility of a multi-robot system. However, a more detailed analysis is required to build a response surface and to eventually confirm the trends reported in the Fig. 5.4. In fact, for this additional type of analysis, more levels of each variables must be considered. In this case, a penalty function should be introduced, as the wrong answers should be included to investigate the response time in a complete way.

5.3.3 Conclusions

In this section, a first approach to legibility of a multi-robot system has been presented. In particular, we investigated the possibility of communicating robots' intentions, in terms of *spatial goal*, to the user through some features that describe the movement of a multi-robot system. These features are called *motion-variables* and are: trajectory, dispersion and stiffness. Our aim was to answer the question: do these *motion-variables* significantly affect the legibility of a multi-robot system?

The experimental validation was carried out in a virtual reality set-up and the retrieved data were analyzed by standard statistical methods. The trajectory variable was found to be relevant for the correctness of the communication from the multi-robot system to the user, while dispersion and stiffness were relevant considering the time that users took to correctly understand robots' intentions.

5.4 Multiple Groups Legibility

The natural evolution of the previous section is to consider more than one group of robots at the same time, because in a general application of multi-robot system it is frequent to divide the robots in multiple groups and to assign to them different goals. In this section we report the work published in [145], where we studied the legibility, always in terms of *spatial goal*, of multiple groups of robots.

5.4.1 Methods

The influence of the *motion-variables* over the legibility of a multi-robot system composed by multiple groups has been investigated through experiments carried out in a virtual reality set-up. We only report in the following the main differences with respect to the experimental method that has been presented in Section 5.3.1.

Experimental Scenario

The experimental scenario used in this user study was very similar to the one reported in Section 5.3.1. However some differences were introduced: the goals were increased from five to seven (see Fig. 5.5a), as a consequence of the presence of multiple groups, and an interface for choosing the answers 5.5b was introduced.

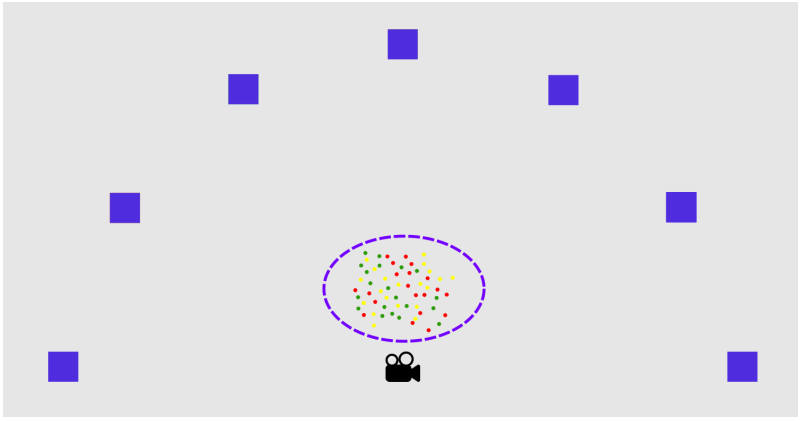
Factorial Experiment

We decided to investigate a full factorial plan with three independent variables and each had two possible values, which leads to 2^3 trials. This type of plan has allowed us to investigate the main effect, namely the influence of the *motion-variables*, and also of the interactions between them. The other measures introduced in Section 5.3.1 were followed also in this user study to respect the rules of the design of experiments [158].

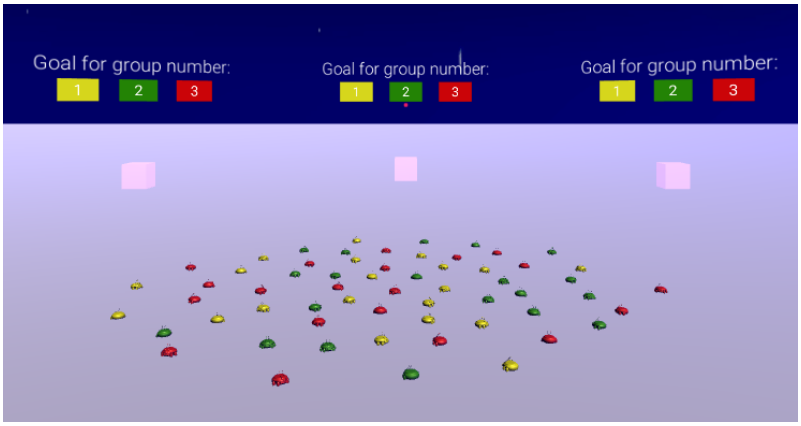
Independent Variables

The independent variables of the experiments are the *motion-variables* (trajectory, dispersion and stiffness), each defined on two levels in such a way to make the difference recognizable by the user. The multi-robot system is the same described in Section 5.3.1, and the trajectories chosen to be investigated,namely the *minimum-jerk* and the *arc-trapezoidal* were built in the same way. The difference is in the way the potential fields are built to deal with the interaction among the different groups. In fact, the attractive potential $V_{a_{i,j}}$, described in (5.5), acts only among the robots that belong to the same group, and the repulsive potential is slightly modified to:

$$V_{rep_{i,j}} = \begin{cases} K_{rep_{i,j}} \left(\frac{1}{3} d_{ij}^3 - d_{min}^3 \ln d_{ij} - \frac{1}{3} d_{min}^3 + d_{min}^3 \ln d_{min} \right)^2 & \text{if } d_{ij} \leq d_{min} \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$



(a) Top-down view of the position of the goals (*colored cubes*) and of the robots (*circled in blue*) with respect to the user (*camera icon*).



(b) Interface for the choice of the answers. The user should choose towards which goal each group is moving.

Figure 5.5: Virtual reality environment for the multiple groups experiment.

The repulsive potential acts among all the robots, but not with the virtual agent of each group. Tables 5.7 and 5.8 report all the values employed during the experiments, which were tuned empirically after preliminary trials.

Table 5.7: Dispersion and stiffness levels.

		−	+
d_0	[m]	1	4
$K_{a_{i,j}}$	[N/m]	0.1	1
$K_{rep_{i,j}}$	[N/m]	1	10

Table 5.8: Constant variables.

$K_{v_{i,v}}$	[N/m]	30
m	[kg]	0.01
b	[Ns/m]	10
d_1	[m]	5
d_{1_v}	[m]	10
d_{min}	[m]	0.8
d_{0_v}	[m]	0.1

Fig. 5.6 represents some examples of the trajectory of the robots. The positions, previously saved offline, have been then used directly in the Unity program to replicate the exact behavior.

Statistical Analysis

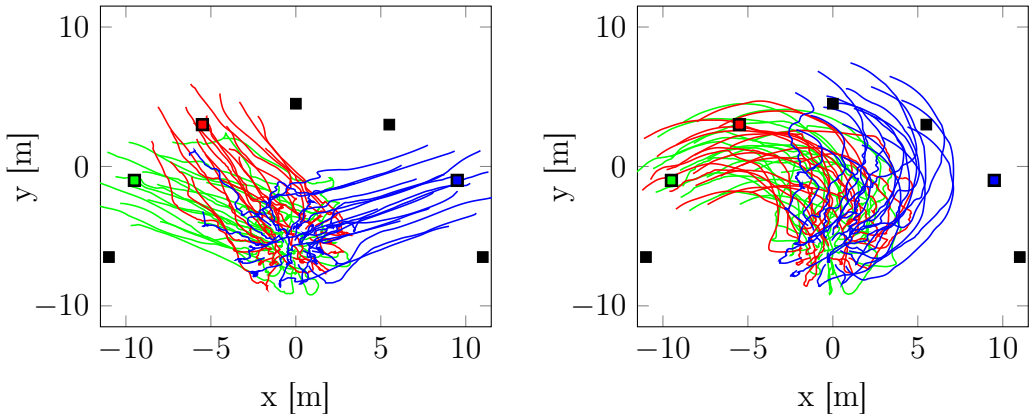
To analyze the relevance of the *motion-variables* over the legibility of the system, a 3-way and a 2-way ANOVA were carried out. In both analysis we tested the null hypothesis over the dependent variable:

H0 - Null hypothesis: *there is no relation between the motion-variables and the legibility of the system (correctness and response time).*

In general, a n-way ANOVA allows us to discover if any effect exists between the n independent variables and the dependent ones, which can be approximated by a random distribution in the case of validation of the null hypothesis. In fact, both variables (correctness and time response), can assume continuous values.

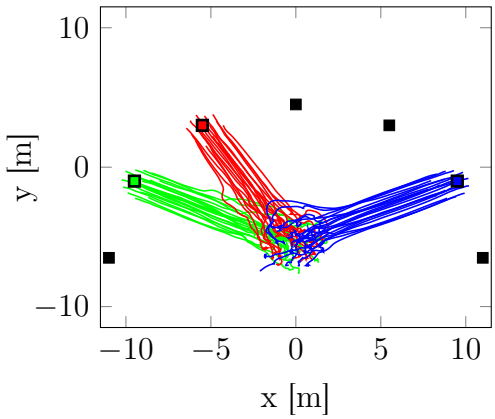
Users

The study was conducted over 15 users (age 26.73 ± 3.9, 5 females and 10 males). None was involved in the design of the experiments and they all were new to the trials. They are students and researchers of our engineering department. The experimental procedure followed the one reported in Section 5.3.1. We applied also here the within-subjects methods [163], hence, every user was exposed to all the combinations of the *motion-variables*. During the user-study we discovered that all the users had already tried virtual reality, hence we can state, with a sufficient grade of certainty, that the results are not influenced by the novelty of the means used to carry out the experiments.

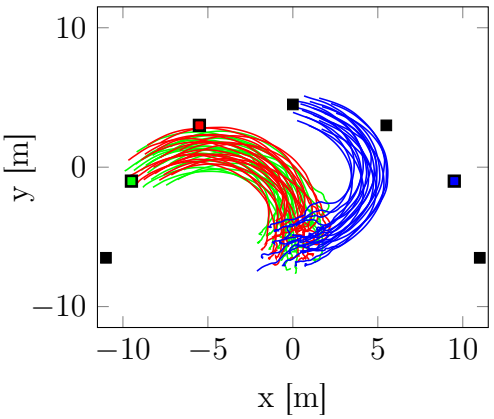


(a) $\mathcal{T}$=*Minimum-jerk*, $\mathcal{D}$=*Large*, $\mathcal{S}$=*Hard*.

(b) $\mathcal{T}$=*Arc-trapezoidal*, $\mathcal{D}$=*Large*, $\mathcal{S}$=*Hard*.



(c) $\mathcal{T}$=*Minimum-jerk*, $\mathcal{D}$=*Small*, $\mathcal{S}$=*Soft*.



(d) $\mathcal{T}$=*Arc-trapezoidal*, $\mathcal{D}$=*Small*, $\mathcal{S}$=*Soft*.

Figure 5.6: Some examples of robots' movement in different trials: each line represents the trajectory of a robot and the color represents the robot's group. The squares represent the goals.

5.4.2 Results

The dependent variables, correctness and response time, are analyzed with a statistical analysis and, in the following, the results are presented.

Correctness

Correctness is represented by the number of correct answers that each user gave. The 3-way ANOVA over the *motion-variables* outlines that trajectory ($\mathcal{T}$) and dispersion ($\mathcal{D}$) are the main effects that statistically influence the correctness. While the only interaction that results significant is trajectory-dispersion ($\mathcal{TD}$).

Table 5.9: p-value from the 3-way ANOVA.

Variable	p-value
$\mathcal{T}$	< 0.0001
$\mathcal{D}$	0.0004
$\mathcal{S}$	0.4107
$\mathcal{TD}$	0.0039
$\mathcal{TS}$	0.2906
$\mathcal{DS}$	0.7241
$\mathcal{TDS}$	0.9063

Table 5.10: p-value from the 2-way ANOVA.

Variable	p-value
$\mathcal{D}$	0.2924
$\mathcal{S}$	0.0001
$\mathcal{DS}$	0.0619

Table 5.9 reports the p-value of all the main effects and also of the interaction variables. In particular, the p-value represents the smallest level of significance at which the null hypothesis would be rejected. A smaller p-value corresponds to a stronger evidence in favor of the alternative hypothesis. This means that we can reject the null hypothesis for the variables $\mathcal{T}$, $\mathcal{D}$ and $\mathcal{TD}$, namely we can assess that these variables influence the correctness. The ANOVA was carried out with a significance level equal to 0.05, hence the results can be stated with a confidence equal to 95%. In fact all the p-values in Table 5.9 smaller than 0.05 allow us to reject the null hypothesis.

As this study is only a preliminary inquiry it is difficult to retrieve a function that links the answer with the independent variables. However, for the trajectory variable the mean of correct answers for the *minimum-jerk* is 1.47 and for the *arc-trapezoidal* is 0.35. The difference is such that we can state that the *minimum-jerk* trajectory outperforms the other one. This confirms the results obtained in the case of a single group of robots, as reported in Section 5.3.2.

Response Time

The response time is analyzed using only the data of the trials where the user answered correctly for all the groups. This choice is motivated by the fact that the

interaction should be effective, rather than quick. To analyze the response time, a penalty function is usually used to consider the wrong answers, or the errors, of the user [167]. But in this case it is not suitable because we should not alter the results before analyzing it in a statistical manner. The analyzed variables are dispersion ($\mathcal{D}$) and stiffness ($\mathcal{S}$) (and their interaction ($\mathcal{DS}$)). The trajectory variable is excluded because the data were too unbalanced in favor of the *minimum-jerk* trajectory.

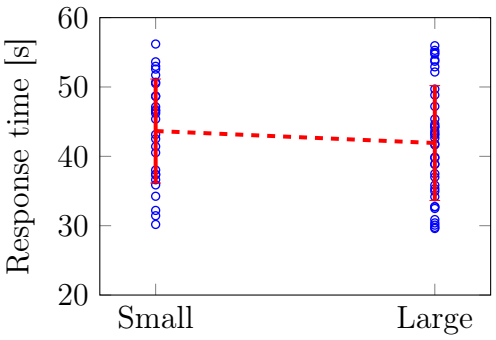
From Table 5.10 we can state that $\mathcal{S}$ is the only *motion-variable* that affects the response time. In fact, it is the only one with a p-value smaller than 0.05. However, the interaction variable $\mathcal{DS}$ has a p-value pretty close to the minimum significance level and, as can be retrieved in Fig. 5.7c, the mean value of the different combinations of $\mathcal{DS}$ are visually different. Comparing the graphs in Fig. 5.7 with the same graphs in Section 5.3.2, it is possible to find the same trends. However, the slope of the lines is less pronounced, especially for $\mathcal{D}$ and $\mathcal{DS}$: this is directly connected to the fact that the two variables are not significant for the response time. If we consider the mean values of the levels of the significant variable $\mathcal{S}$, we can say that the best configuration is the *Soft* one.

The statistical analysis reports that the considered *motion-variables* influence significantly the legibility of a multi-robot system, also in the case of multiple groups. From the preliminary analysis reported above we can assert that the best combination is *minimum-jerk* for the trajectory variable, *Large* for dispersion and *Soft* for stiffness. This combination is also one of the best for the single group configuration that we have discovered in Section 5.3.2.

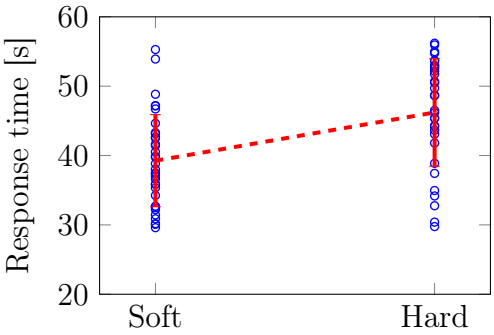
However, to define a more precise relation between the *motion-variables* and legibility, a wider factorial plan is needed. In particular, it would be interesting to consider more levels for each variables, except for the trajectory variable for which it is quite clear that it is better choose a path that links directly the starting point to the ending point, instead of one that follows a curve.

5.4.3 Conclusions

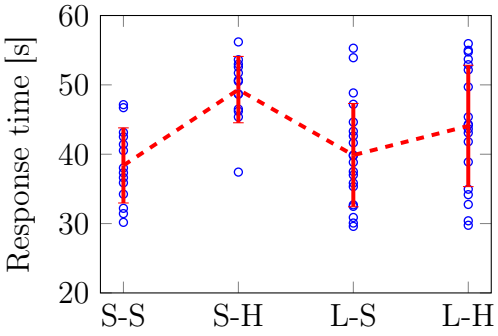
In this section, we studied the influence of a set of *motion-variables* upon the legibility, in terms of *spatial goal*, of a multi-robot system, specifically divided in multiple groups. The results were similar to those of Section 5.4, and strengthen the thesis of being able to communicate implicitly the spatial goal of a multi-robot system, or of more multi-robot system. Hence, the results of this section, together with those of Section 5.4, corroborate the first part of Definition 5.2.1.



(a) Dispersion level:
Small: $\mu_T = 43.6$, $\sigma_T = 7.5$
Large: $\mu_T = 41.9$, $\sigma_T = 8.3$



(b) Stiffness level:
Soft: $\mu_T = 39.3$, $\sigma_T = 6.6$
Hard: $\mu_T = 46.2$, $\sigma_T = 7.7$



(c) Dispersion-Stiffness level:
Small-Soft: $\mu_T = 38.4$, $\sigma_T = 5.4$
Small-Hard: $\mu_T = 49.3$, $\sigma_T = 4.8$
Large-Soft: $\mu_T = 39.8$, $\sigma_T = 7.4$
Large-Hard: $\mu_T = 44.1$, $\sigma_T = 8.7$

Figure 5.7: Response time analysis (in blue scatter plot of the trials, in red the mean (μ_T) and the standard deviation (σ_T) of each level).

5.5 Coordination Behaviors and Legibility

In this section we extend the study on the legibility with respect to spatial goals, using different types of control laws, instead of potential fields as in Section 5.3 and 5.4, and different numbers of robots. The choice of analyzing different control laws was dictated by the fact that as legibility is a macroscopic characteristic of the system, it seemed reasonable to understand whether a holistic view of the multi-robot system would have improved its legibility. In fact, on the other hand, the *motion-variables* studied in the previous sections are microscopic characteristics of the motion of the multi-robot system. In addition, we investigate the second part of Definition 5.2.1, namely the possibility to communicate implicitly the *coordination behavior* of the multi-robot system. The study reported in this section is currently under review in [100].

5.5.1 Multi-Robot Coordination Objectives

A large body of work on multi-robot systems and robotic swarms deals with how to coordinate large teams of robots to produce collective behaviors that go beyond the capabilities of the robots as individuals [168]. As such, the aim is to perform collective, emerging behaviors that arise from rules executed by individual robots acting on local information. The fulfillment of objectives is thus typically analyzed from the team-level behavior, rather than from the individual actions of the robots.

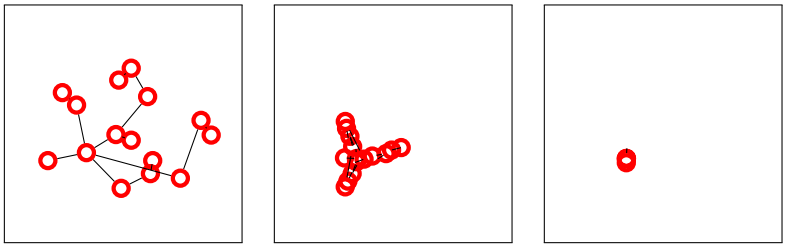
Taking the taxonomy outlined in [83] as a reference, we analyze the legibility of standard multi-robot coordination objectives, namely consensus, formation, and coverage. These algorithms are posed such that the group's coordination objective is encoded through a cost that represents the performance of the team as a whole, and the individuals cooperate towards that objective by following a direction of the gradient of the cost.

In this section, we outline the control laws executed for each of the coordination objectives utilized in the user study. To this end, let us consider a team of N planar robots[3], with the position of robot i being denoted as $x_i \in \mathbb{R}^2$, $i \in [N]$. While many kinematic configurations can be considered for the individual platforms in robotic swarms (e.g., omnidirectional [169] or differential drive [170, 171, 172]), we here opt for omnidirectional robots to minimally distort the ideal collective behavior considered in the control laws. Consequently, the movement of the robots can be described through single integrator dynamics, i.e.,

$$\dot{x} = u, \tag{5.9}$$

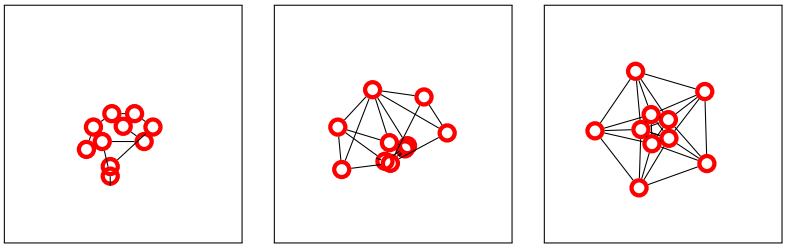
with $x = [x_1^T, \dots, x_N^T]^T \in \mathbb{R}^{2N}$ being the aggregate state of the system, and $u = [u_1^T, \dots, u_N^T]^T \in \mathbb{R}^{2N}$, its control. If a coordination objective is to be fulfilled, the control input of each particular robot, u_i, $i \in [N]$, will be influenced by the state

[3]Note that, while the control laws outlined in this section are generalizable to an n-dimensional space, i.e., $x_i \in \mathbb{R}^n$, $i \in \{1, \dots, N\}$; we present them here with $x_i \in \mathbb{R}^2$ to relate to the planar robots considered in this study.



(a) $t = 0$s. (b) $t = 2$s. (c) $t = 10$s.

Figure 5.8: Evolution of the consensus protocol in (5.10) on a team of 15 robots. The red circles represent idealized robots with no physical footprint, which allows them to overlap. The black lines represent the underlying graph, i.e., the pairwise connections among the robots. Given that the graph is connected, the consensus protocol makes the robots to converge to a point.



(a) $t = 0$s. (b) $t = 2$s. (c) $t = 10$s.

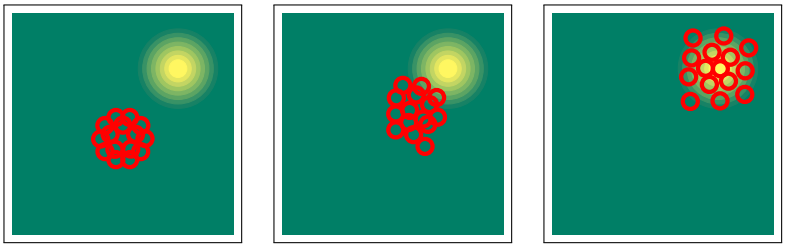
Figure 5.9: Evolution of the formation control protocol in (5.12) on a team of 10 robots. The red circles represent idealized robots with no physical footprint and the black lines represent the underlying graph. A connection between robots i and j, represented by a graph edge, corresponds to a desired distance, δ_{ij}, that is to be maintained to achieve the desired geometry. The desired formation, composed by two concentric pentagons, is achieved in (c).

and the actions of its neighbors. This flow of information among adjacent robots is typically represented through a graph $\mathcal{G} = (V, E)$ (refer to Section 2.1.2 for the notation on graphs).

The remainder of this section provides an outline of the different multi-robot coordination objectives analyzed in the user study in Section 5.5.2, i.e., consensus, formation and coverage. For each of them, we include a general description of the expected behavior as well as the laws to be executed by the robots.

Consensus

The consensus protocol deals with the spatial aggregation of agents in a common point through local coordination of neighbors [173, 174, 175]. Assuming the graph describing inter-robot sensing, $\mathcal{G}$, is connected, the team converges to a common



(a) $t = 0$s. (b) $t = 2$s. (c) $t = 10$s.

Figure 5.10: Coverage control from (3.7) executed by a team of 15 robots. The robots are tasked with covering the domain with respect to a density function, whose values are depicted through a heat map where higher values of the density function are represented in yellow and lower ones in green. As the robots move according to the coverage control law, the team concentrates around areas where the density is higher.

location if each individual executes the control law

$$\dot{x}_i = \kappa \sum_{j \in \mathcal{N}_i} (x_j - x_i), \quad \forall i \in [N], \tag{5.10}$$

where $\kappa \in \mathbb{R}_{>0}$ is a proportional gain. Executing the control law in (5.10) implies that robot i will be moving in the direction that averages the relative positions of its neighbors, $x_j - x_i$, $\forall j$ adjacent to robot i. Asymptotically, this will make them converge to a point $\bar{x} = x_1(\infty) = \cdots = x_N(\infty)$, that is, the robots will reach consensus about where to meet, as long as the graph stays connected.

An instance of the consensus protocol in (5.10) is shown in Fig. 5.8, where a group of ideal robots, modeled as points, starts scattered over the domain and agree on a location where to meet. Note that, in the experiments included in Section 5.5.2, the robots have a physical footprint and, therefore, are unable to finish stacked on top of each other. As opposed to the ideal case presented in Fig. 5.8, when executing the consensus protocol on a team of real robots, they will aggregate as much as possible without incurring inter-robot collisions, thanks to the implemented collision avoidance.

Formation Control

A standard problem when controlling a multi-robot team is that of moving the individual robots so that they display a particular shape. This is typically achieved through formation control strategies, e.g., [176, 30, 177], that specify how the individual robots should move in order to collectively display a shape.

The specification of the geometry to be realized by the team is key to the operation of a formation controller. To this end, we can specify a 2D formation Δ through a set of relative, desired inter-agent distances,

$$\Delta = \{\delta_{ij} \in \mathbb{R} \mid \delta_{ij} > 0, i, j = [N], i \neq j\}. \tag{5.11}$$

We assume here that Δ is a feasible formation, that is, that the distances are not conflicting and, thus, the geometry can can be realized (see [178] for details on feasible formations).

Having specified the formation, we can recover the structure of the consensus protocol in (5.10) and make each robot move according to a weighted average of the relative positions of its neighbors,

$$\dot{x}_i = \sum_{j \in \mathcal{N}_i} w_{ij}(\|x_i - x_j\|, \delta_{ij})(x_j - x_i), \quad \forall i \in [N], \tag{5.12}$$

where the weight w_{ij} is positive if the neighboring robot is further than the desired distance, i.e., if $\|x_i - x_j\| > \delta_{ij}$, and negative otherwise. For the experiments in Section 5.5.2, the weights are calculated as

$$w_{ij}(\|x_i - x_j\|, \delta_{ij}) = \frac{\|x_i - x_j\| - \delta_{ij}}{\|x_i - x_j\|}, \quad (i,j) \in E. \tag{5.13}$$

Note that, in order to execute the control law in (5.12), each robot needs to be cognizant of which robots conform its neighborhood, $\mathcal{N}_i$, their relative positions, $x_j - x_i$, and which distance is to be maintained with respect to each of them, δ_{ij}, as reflected in Δ. Therefore, under this protocol the robots are no longer anonymous: each robot needs to know the identities of other robots to discriminate which ones it needs to coordinate with.

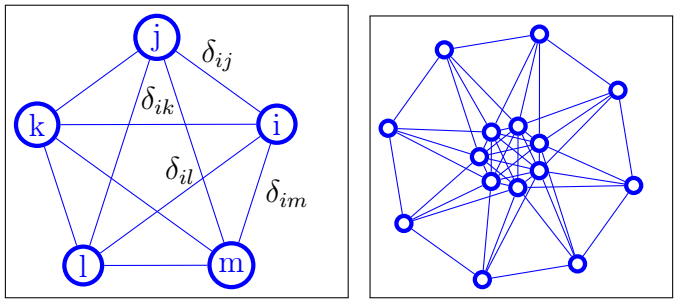
Fig. 5.9 shows the evolution of a team of 10 robots achieving a formation composed by two concentric pentagons, as specified by a formation Δ. The formation control protocol in (5.12) ensures that the inter-robot distances are maintained but it does not influence the rotation of the formation with respect to the global frame.

In order to execute the formation controller in (5.12), we need a formation Δ specifying the desired pairwise distances between robots. For the user study in Section 5.5.2, we choose formations where the robots are arranged in concentric regular polygons. In particular, for $N = 5$ we choose a pentagon and for $N = 15$, two concentric polygons (one heptagon and one octagon). For the achieved formation see Fig. 5.11a and Fig. 5.11b, respectively.

The formations are generated as follows: let d_g be the distance between goals (they are set equally spaced out), then the formation is circumscribed by a circle of radius $R_{out} = d_g/3$. For the case of $N = 15$, the inner polygon is concentric to the outer octagon and circumscribed by a polygon of radius $R_{in} = R_{out}/4$. With the geometries defined, the formation Δ is obtained by placing the robots on the vertices of the polygons. The pairwise distances δ_{ij} for a robot i will correspond to the Euclidean distance between its corresponding vertex and the rest of the robots in the polygon. An example of how these distances are generated is shown in Fig. 5.11a.

Coverage Control

For the standard coverage control we refer to Section 3.3, where we described the common approach to distribute a multi-robot system over a domain $Q \subset \mathbb{R}^2$. In order to distribute the robots around a point that has particular importance, we recall



(a) $N = 5$. (b) $N = 15$.

Figure 5.11: Formation for the different number of robots. These formation are obtained applying the formation control law (5.12), without any spatial drive, in order to clarify the achieved shape. The blue lines correspond to distances, δ_{ij}, to be maintained between robots.

the possibility of defining a probability density function $\phi : Q \to \mathbb{R}_{\geq 0}$. An example of this density function is depicted in Fig. 5.10, with *yellow* shades corresponding to higher values of ϕ.

Time-Varying Coverage

In some situations, the importance of the points in the domain may evolve over time. As a result, the coverage strategy outlined in Section 3.3 needs to adapt to reflect possible dynamic changes to the density function. Considering a dynamic density, $\phi : (q,t) \in Q \times \mathbb{R}_{>0} \to \phi(q,t) \in \mathbb{R}_{\geq 0}$, the optimization function can be reformulated as,

$$\mathcal{H}(x,t) = \sum_{i=1}^{N} \int_{V_i(x)} \|x_i - q\|^2 \phi(q,t) dq, \tag{5.14}$$

where the performance function is chosen as $f(x) = x^2$, and, hence, the purpose of the controller will be to minimize the optimization function $\mathcal{H}(x,t)$. The introduction of a dynamic density in the optimization function requires a modification of the coverage controller. As investigated in [59], one can minimize the optimization function in (5.14) by setting

$$\dot{x} = \left(\mathbb{I} - \frac{\partial C}{\partial x}\right)^{-1} \left(\kappa(C(x,t) - x) + \frac{\partial C}{\partial t}\right), \tag{5.15}$$

where $C = [C_{V_1}^T, \ldots, C_{V_N}^T]^T \in \mathbb{R}^{2N}$ is the vector containing the centers of mass calculated as in (3.6) but with respect to the time-varying density function, $\kappa \in \mathbb{R}_{>0}$ is a proportional gain, and $\mathbb{I} \in \mathbb{R}^{2N \times 2N}$ is the identity matrix.

Spatial Drive of the Multi-Robot System

A core legibility aspect studied in this section relates to the inference of the spatial goal of the multi-robot system, i.e., where is the team headed towards within the

domain. In the consensus or the formation control protocols, one way of displacing the multi-robot system throughout the domain is to independently command one of the robots in the team, a *leader*, to move towards a certain location. As a consequence of the inter-agent interactions encoded in the control laws (5.10) and (5.12), the movement of this leader robot influences the positioning of the whole team in the domain. If we denote as $x_L \in \mathbb{R}^2$ the position of the leader robot, then we can make the consensus and formation coordination objectives move towards a spatial goal by executing the following go-to-goal controller,

$$\dot{x}_L = \kappa_{goal}(x_{goal} - x_L), \tag{5.16}$$

where $x_{goal} \in \mathbb{R}^2$ denotes the desired location of the team, and $\kappa_{goal} \in \mathbb{R}_{>0}$ is a proportional gain, respectively. With the purpose of minimally disturbing the motions corresponding to the multi-robot coordination objectives, we discard selecting the leader among of the N robots displayed in the experiment and, rather, pick the leader robot to be virtual, similarly to what we did with potential fields in the previous sections.

Coverage, on the other hand, naturally affords the spatial allocation of the multi-robot system within the domain. Under the control law presented in Section 3.3, the robots are displaced towards the spatial goals by specifying the density function, ϕ, as a Gaussian distribution of the form

$$\phi(q) = \frac{k}{\sigma^2} \exp\left(-\frac{\|q-\mu\|^2}{2\sigma^2}\right), \tag{5.17}$$

where $q \in Q$ is a point in the domain, $k \in \mathbb{R}_{>0}$ is a proportional constant, $\sigma \in \mathbb{R}_{>0}$ is the variance, and $\mu \in \mathbb{R}^2$ is the mean of the function. By making the mean, μ, coincide with the spatial goal, x_{goal}, the team moves towards the objective location when executing the control law in (3.7).

For the case of time-varying coverage, we consider a density function $\phi(x,t)$ of the form in (5.17), where the dependency on t stems from considering a time-varying mean, $\mu(t)$. The evolution of the mean serves the purpose of guiding the multi-robot team from its initial location towards the spatial goal. Consequently, μ coincides with the mean position of the robots at the start and moves on a straight line towards the spatial goal as the experiment progresses according to

$$\mu(t) = \bar{x}(t_0) + \frac{x_{goal} - \bar{x}(t_0)}{t_F - t_0}(t - t_0), \tag{5.18}$$

where $\bar{x} \in \mathbb{R}^2$ denotes the mean position of the robots, $\bar{x}(t) = \sum_{i\in[N]} x_i(t)/N$, and t_0 and t_F the initial and final time, respectively. For the user study, $t_0 = 0$s and $t_F = 60s$.

In addition, the collision avoidance among the robots is implemented using Control Barrier Functions, as they allow us to minimally modify the behavior and motion of the multi-robot system. In fact, in the previous sections we used repulsive potential fields to avoid collisions, however, introducing additional forces introduce modifications of the inter-robot behavior. thus possibly modifying the legibility of the system. We implemented the CBF for collision avoidance as the one used in Section 2.2.3, see (2.19).

5.5.2 Methods

In this section we describe the user study that was conducted in order to investigate the legibility of multi-robot systems. Following the preceding studies in Section 5.3.1 and 5.4.1, this user study investigates how the different controllers described in Section 5.5.1, as well as how the number of robots executing them, affected the legibility of the system, both in terms of spatial goal and coordination objective. Hence, this study takes a step further in understanding the legibility of multi-robot systems by analyzing how distinguishable the different coordination objectives are for the human observer. To this end, users were asked to recognize the spatial goal of the multi-robot system and the different kinds of controllers, namely we wanted to discover if the users can understand the coordination objective of the multi-robot system, in addition to the spatial goal. As in the previous section, we will report only the difference with respect to the experimental method reported in Section 5.4.1.

Experimental Procedure

The experiment was divided in two parts: one related to the spatial goal of the multi-robot team and one to its coordination objective.

First Part

Users were asked to recognize as fast as possible the goal towards which the robot team was moving. The answer was given by means of the Oculus Remote (see Fig. 5.1c) combined with the direction of the head of the user. Once the users had identified with a sufficient degree of confidence the probable goal, they had to point towards the chosen goal (in the field of view of the user, a point was shown that corresponded to the direction tracked by the system) and press the central button of the Oculus Remote.

Second Part

Users had to recognize both the spatial goal and the coordination objective of the multi-robot system. The coordination objectives were introduced to the users via three representative examples shown in video. No information about the kind of controllers or other features of the coordination objectives were disclosed to the users, with the objective of not to inserting any bias in the experiments. Furthermore, the examples were significantly different from the movements of the second part of the experiment. In fact, the number of robots did not correspond to neither used in the experiments nor the directions of the movements correspond to one of the goals. The examples consisted of a team of 8 robots moving right to left in front of the user. After this introduction, the second part of the experiment started and the users were asked to recognize the spatial goal and the coordination objective of the group. They were informed that the amount of time was measured separately, namely they had to give the answers (spatial goal and coordination objective) as soon as they understood one of the two. Each trial ended when the user gave both answers. The answer for the spatial goal was provided in the same way as in the first part of

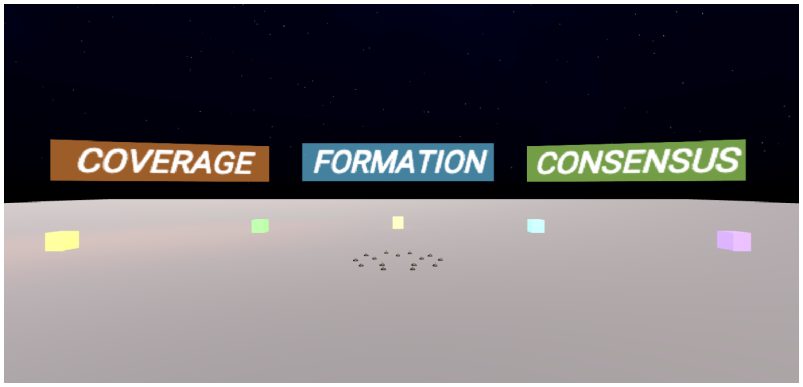


Figure 5.12: Panels with the coordination objectives' name in the second part of the experiment. In this part, the user had to understand the spatial goal and the coordination objective of the multi-robot system.

the experiment, instead the chosen coordination objective was specified through the interface in Fig. 5.12.

The experiment ended with a survey with general questions for the user, including demographic information and previous experiences with virtual reality, and some specific questions regarding the experiment itself. While these data do not influence the statistical analysis of the experiment, these questions clarify the perception of the experiments from the users. The questionnaire was based on a 5-point Likert scale.

Experimental Variables

The *independent variables* of this study are the coordination objectives of the multi-robot system and the number of robots in the group. In the first part, the coordination objectives were: consensus, formation, and coverage. In the case of coverage, both static and time-varying density functions were evaluated with the objective of analyzing whether the two corresponding controllers affect or not the legibility of the multi-robot system. On the other hand, in the second part, time-varying coverage was not utilized given that, visually, the robots behaved too similarly to the consensus protocol for this scenario, as seen in preliminary tests. In fact, in both cases the robots agglomerate at the beginning and then they move towards the goal. The only difference was in terms of dispersion in the environment: the time-varying coverage had a higher dispersion with respect to the consensus.

For the coordination objectives in Section 5.5.1, each control law has certain parameters that define the behavior of the multi-robot system. These parameters are distinctive of the controllers and they should not be considered as independent variables. The parameters used for each coordination objective are reported in Table 5.11. The saturation on the velocity, $\dot{x}_{sat}$, was chosen to produce a realistic behavior of the robots, which have a maximum speed at which they can move. The coverage parameter σ was chosen to produce a similar spatial occupancy, in terms of area occupied by the convex hull of the positions of the robots, as the formation

Table 5.11: Parameters for the Coordination Strategies.

Parameter	Consensus	Formation	Coverage
κ	1	1.5	1
κ_{goal}	$1.5 * \kappa * N$	$0.5 * \kappa$	-
$\dot{x}_{sat}$ [m/s]	0.4	0.4	0.4
d_{min} [m]	0.1	0.1	0.1
σ	-	-	1

control (see Section 5.5.1 for further details). On the other hand, consensus has a smaller dispersion given its inherent operation and the agglomeration of the robots was only affected by the safety distance, d_{min}. The parameters corresponding to the proportional gain κ, which properly scales the input, and κ_{goal} were set to make the team achieve the spatial goal within the specified time frame of 60s.

Note that, differently from the user studies reported in the previous Section 5.3 and 5.4, *trajectory*, which is the path followed by the group of robots during the movement, is not an independent variable in this study. In fact, based on the results of the previous sections, the best trajectory to communicate the spatial goal turned out to be the one that connects with a straight line the starting point of the robots to their goal. The implementation of these movements is described in Section 5.5.1.

On the other hand, the number of robots we tested for were 5 and 15, and we chose these numbers so that they were distinguishable between themselves and also different from the number used in the previous section, where we the experiments were run on teams of 20 robots.

In contrast, the dependent variables of the experiments are the correctness of the answers and the time to answer taken in both parts of the experiments. The term *dependent variables* refers to the quantities that we measure during the experiments and upon which we will carry out the statistical analysis. We consider these dependent variables because they represent the effectiveness of a legible system, namely we define the ability to communicate in Definition 5.2.1 in terms of providing the correct information and in a short time.

Factorial Design

To investigate the statistical correlation between the independent and the dependent variables, we built a factorial plan [158]. Given the independent variables of the two parts of the experiment, as reported in Section 5.5.2, we had 4*2 and 3*2 combinations in the two parts, respectively. In addition, following the design of experiment methodology [158], we took some precautions to reduce the disturbing factors that might influence the answer of the experiments. We report some additional precautions that we took during this user study.

Position of the Goals

We replicated the placement that we used in Section 5.3, see Fig. 5.1b. The goals were randomized and the same combination of goal and trial could not happen twice for a given user. Thus, in the second part of the experiment the goal used in the first part for a given combination of coordination objective and number of robots was excluded.

Learning Effect

To reduce the effects of the learning effect that takes place as participants advance through the user study, we introduced some additional parts to the experimental procedure described in Section 5.5.2. In particular, the experiments started with a small tutorial to familiarize the user with the input method. Then a practice scenario allowed the users to test the input method and to get used to the surrounding environment. In the second part of the experiment, the users were able to see again the demonstrative examples as many times as they wanted. In addition, the order of the trials of the factorial plan were randomized.

Movement of the Robots

The controllers of the robots, described in Section 5.5.1, were implemented in MATLAB®, and the trajectories of all the robots were saved offline in order to be replicated exactly in the Unity environment during the experiments. This affords high repeatability during the experiments, which results in a reduction of the disturbing factor that may be introduced from random modification of the trajectories in a real-time implementation.

All the trajectories lasted 60s, namely the group of robots reached the spatial goal in the given time. We experimentally found out that this is a reasonable time (given also the results of the previous sections). The initial positions of the robots were equal for all the trials and were chosen to reach the specified formation as quickly as possible. This was done to balance out the formation objective with the consensus and coverage objectives in terms of time required for the multi-robot team to achieve a distinguishable behavior, since we discovered that, for the set of parameters chosen in the experiments, the formation was the slowest to converge to a recognizable pattern.

Users

We performed the experiments on a group of 21 users (age 25.76 ± 5.37 y.o., 2 females and 19 males). All the participants were students or researchers within the engineering department, but none of them were involved in the project nor they partook in the building of the experiment. Regarding their robotics background, six users had a multi-robot background and five did not have any robotic knowledge, namely they were completely new to the subject. All the users signed a consent form and they were instructed on the experimental procedure as described in Section 5.5.2. We conducted a within-subjects study (see [163] for details) and, according to this

methodology, each user tests all the possible combinations of the independent variables in both parts of the experiment. This choice allowed us to obtain a sufficient amount of data from a limited number of participants. The main drawback of this method may be the learning effect or the fatigue, which we addressed with the precautions described above.

In addition, the users were asked to express any possible thoughts about the motion of the robots or what they perceived during the trials. This follows the "think-aloud" protocol [179, 165] that allows us to gather some additional information from the user during the experiments. This protocol is beneficial to retrieve comments, opinions, and thoughts that are not directly contemplated in the survey.

Lastly, the possible disorientation that may arise through the use of a virtual reality headset was counterbalanced by the fact that the majority of the users claimed to have tried at least once virtual reality (10 users had used it once or twice, 7 more than twice, and 4 users had never used it before).

5.5.3 Results

This section outlines the statistical analysis and discussion of the data acquired during the experiments. The objective is to understand whether the multi-robot system investigated in the experiments is legible, in terms of Definition 5.2.1, by analyzing the relationship between the independent variables and legibility.

Statistical Tools

The data obtained from the experiments was processed using standard statistical tools [158]. These tests aim at investigating the impact of the coordination objective and of the number of robots on the legibility of the system. In this section, we briefly introduce the statistical tools used in the analysis.

Usually, the first step in a statistical analysis is to check the significance of the independent variables over the dependent variables. The null hypothesis is that there is no statistical relevance, which can be rejected or supported based on the calculation of the p-value. Considering a significance level of 0.05 implies that, if the p-value is smaller than 0.05, we can reject the null hypothesis with a 5% of risk of asserting a wrong assumption. To calculate the p-value, two different approaches are used depending on the kind of dependent variable: categorical, for variables like blood type or binary answers, or continuous, for variables such as time or temperature. In particular, for the analysis of our user study we have one categorical variable (correctness of the answer given by the users) and one continuous variable (time taken to give the answer).

For categorical variables, the significance tests build upon a contingency table, which is a table where the rows are the levels of the independent variable and the columns are the categories of the dependent variable. The cells are filled based on the result of each trial (i.e., based on the value of the independent variable and of the value of the dependent variable, we add an entry to the corresponding cell). Consequently, each cell has a number that corresponds to the frequency of that given combination of independent and dependent variable. Once the contingency

table is built, a significance test can be carried out, with the Pearson's Chi-Square test being a typical choice to determine the significance. However, Fisher's exact test is used instead when one frequency of the contingency table is less than five. Both tests measure the independence between the column and row variables of the given contingency table.

For continuous variables the standard tool is ANOVA (Analysis of variance), which explains whether the independent variables are statistically relevant with respect to the dependent variable. Additionally, a post-hoc test, such as the Tukey-Kramer multiple comparison procedure, can be carried out. This test brings out the differences among the groups (i.e., the combination of the levels of the independent variables) and it can highlight if the statistical relevance is caused by a particular group or not.

Spatial Goal

Along the lines of Definition 5.2.1, the spatial goal of the multi-robot system is the first information that the system should be able to communicate in order to be legible. To study the statistical relevance of the independent variables over the ability of the multi-robot system to communicate its spatial goal, we identify the following null hypotheses:

H1 - Null hypothesis: *there is no statistical relationship between the independent variables and the legibility of the system, in terms of correctness for the understanding of the spatial goal.*

H2 - Null hypothesis: *there is no statistical relationship between the independent variables and the legibility of the system, in terms of response time for the understanding of the spatial goal.*

For analyzing the correctness of the goal, we built the corresponding contingency tables, Tables 5.12 and 5.13. Since correctness is a categorical variable, we need two separate tables, one for each independent variable. Given that the results are not suitable for Pearson's Chi-Square test (see Section 5.5.3 for details), we instead carried out a Fisher's exact test. The p-value for both results was high (see Table 5.14), as expected from the contingency tables. In fact, most of the answers were correct, independently from the coordination objective or the number of robots (no evident differences among the rows of the contingency tables considering the same column). Actually, the percentage of success 98.81% (166 right answers over 168 total) would bring to a lack of relevance of any independent variable. This result is consistent with the outcome of the previous sections because it seems that the main parameter that affects the communication of the spatial goal is the trajectory of the multi-robot system, and in this experiment all the trials had a straight trajectory towards their goal, independently from the coordination objectives or the number of robots. As the Fisher's exact tests have not highlighted any statistical relevance, we did not further investigate this relationship and we can not reject **H1**. This result is also confirmed by the survey, in which the answer to the question *Do you think you have correctly understood the spatial goal of the robot?* was 4.238 ± 0.625.

Table 5.12: Contingency table of correctness of answers with respect to coordination objective.

Coordination objective \ Correctness	Wrong	Right	Total
Consensus	0	42	42
Coverage	1	41	42
Formation	0	42	42
Coverage time-varying	1	41	42
Total	2	166	168

Table 5.13: Contingency table of correctness of answers with respect to the number of robots.

Number of robots \ Correctness	Wrong	Right	Total
5	0	84	84
15	2	82	84
Total	2	166	168

As regards the response time, given its continuous nature, we used ANOVA to discover if the variables chosen are statistically relevant. We discarded the trials with a wrong answer because a penalty function would have to be considered, which can interfere with the data. Table 5.15 reports the p-values for the independent variables considered in the ANOVA, from which we can deduce that coordination objective and its interaction with the number of robots are relevant (in the Tables the relevant variables are highlighted in grey). Instead, the number of robots is not relevant. However it is worth noting that the Tukey-Kramer post-hoc test highlighted that the trial with formation and 5 robots is the one that influenced most the results, namely it was the hardest to understand. This is reported in Fig. 5.13. As a result, we reject **H2** for the coordination objective and its interaction with the number of robots.

Table 5.14: p-values from the Fisher's exact test on correctness.

Independent variable	p-value
Coordination objective	0.6856
Number of robots	0.4970

Table 5.15: p-values from the ANOVA on time response.

Independent variable	p-value
Coordination objective	0.0309
Number of robots	0.0811
Coordination objective-Number of robots	< 0.0001

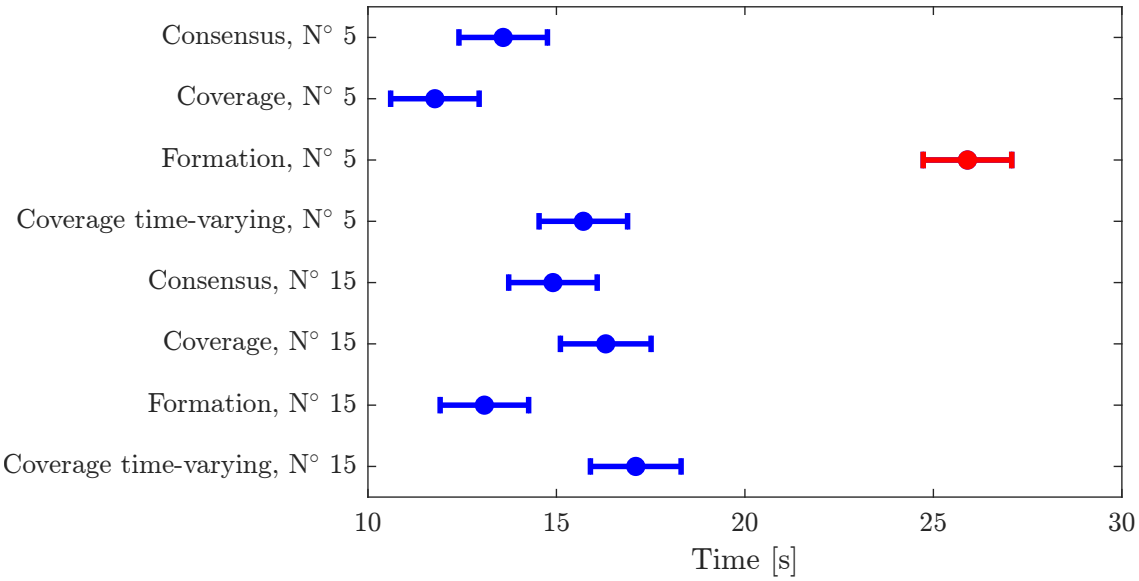


Figure 5.13: Results of the post-hoc test. The trial formation with 5 robots, which is the one significantly different from the other trials, is highlighted in *red*. The probable cause for this disparity is the time needed to achieve the formation, which affects the time performance of this coordination objective.

Influence of Coordination Objective on Spatial Goal Identification

The results of Section 5.5.3 about **H1** were verified also with the data of the second part of the experiment, even if one of the coordination objectives was not considered (time-varying coverage). The percentage of success was 97.6% (123 correct answers in understanding the spatial goal over 126 total), and all the tests for the response time achieved the same results. To verify that the users were not influenced by the fact of being asked to answer to two queries simultaneously we carried out an additional ANOVA adding the part of the experiment as independent variable. In this analysis, we neglected the trials with the time-varying coverage of the first part to avoid an unbalanced study. The null hypothesis for this study is:

H3 - Null hypothesis: *there is no statistical relationship between the two parts of the experiment and the legibility of the system, in terms of response time for the understanding of the spatial goal, namely the addition of the second answer did not affect the legibility.*

This additional analysis confirms **H3** (p-value 0.468 in the ANOVA). Hence, in

the second part, the users were not influenced by the additional question on the coordination objective. The survey confirms also this aspect: the users answered 1.714 ± 0.956 to the question *In the second part, do you think that choosing both the goal and the coordination objective distracted you?*.

Coverage Control

In the first part of the experiment, we implemented both the standard and the time-varying version of coverage control. In order to understand if there is some difference between the two versions, we evaluate the following hypothesis:

H4 - Null hypothesis: *there is no statistical relationship between the type of coverage control and the legibility of the system, in terms of response time for the understanding of the spatial goal.*

To investigate **H4** we carried out ANOVA only between the two types of coverage to understand whether the time-varying aspect can affect the legibility. From the ANOVA, no statistical relevance emerged (p-value 0.0874), but considering the mean (standard coverage 13.98 ± 6.65 s, time-varying coverage 16.69 ± 5.9 s) we can assert that using the standard coverage for the second part is legitimate, since standard coverage is slightly more legible. In fact, the time-varying coverage looks similar to the consensus because the robots converge to a point and then they move towards a goal, i.e., the density function moves towards the goal and, as a consequence, the robots. Therefore, we can not reject **H4**.

Coordination Objective

For the second ability of Definition 5.2.1, we study the ability of the multi-robot system to communicate its coordination objective. The null hypotheses are:

H5 - Null hypothesis: *there is no statistical relationship between the independent variables and the legibility of the system, in terms of correctness for the understanding of the coordination objective.*

H6 - Null hypothesis: *there is no statistical relationship between the independent variables and the legibility of the system, in terms of response time for the understanding of the coordination objective.*

For the analysis of the correctness of the recognition of the coordination objective in the second part of the experiment we proceeded as for the correctness of the goal. Tables 5.16-5.17 report the contingency tables for the coordination objective and for the number of robots, respectively. Given that the entries are sufficiently numerous, we carried out the Pearson's Chi-Square test and we discovered that both independent variables are significant (Table 5.18 reports the p-values). Hence, we can reject **H5**. It is worth noting that even the users perceived a good confidence in recognizing the coordination objective: the users answered 3.667 ± 0.796 to the question *Were you able to distinguish among the coordination objectives?*. In addition, the real percentage of correctness of this part was 68% (86 correct answers

Table 5.16: Contingency table of correctness of answers with respect to coordination objective in the recognition of the coordination objective.

Coordination objective \ Correctness	Wrong	Right	Total
Consensus	5	37	42
Formation	21	21	42
Coverage	14	18	42
Total	40	86	126

Table 5.17: Contingency table of correctness of answers with respect to the number of robots in the recognition of the coordination objective.

Number of robots \ Correctness	Wrong	Right	Total
5	27	36	63
15	13	50	63
Total	40	86	126

over 126 total), which means that the users were able to recognize the coordination objectives.

Table 5.18: p-value from the Pearson's Chi-Square on correctness for the coordination objective recognition.

Independent variable	p-value
Coordination objective	< 0.0001
Number of robots	0.0074

We removed the wrong answers to proceed with the analysis of the response time, as we did for testing **H2** in Section 5.5.3. We checked the balance of the remaining data, namely if some combination had too many or too few entries. Table 5.19 reports the p-values calculated with ANOVA, which allows us to reject **H6**. However, as already pointed out for the response time for the goal recognition in Section 5.5.3, the post-hoc test reported that the trial with formation and 5 robots is the one that differs most from the others, namely it was the most difficult to recognize.

About the survey of this last part we report that consensus was considered as the easiest coordination objective to recognize (100% of the users chose consensus in the question *Which coordination objective was the easiest to recognize?*). After the experiment, many users asserted that consensus was the simplest because they associated the small occupied area to that coordination objective, and, on the other side, formation and coverage occupied a similar area. Instead, coverage had been

Table 5.19: p-value from ANOVA on response time for the coordination objective recognition.

Independent variable	p-value
Coordination objective	< 0.0001
Number of robots	0.0154
Coordination objective-Number of robots	0.0045

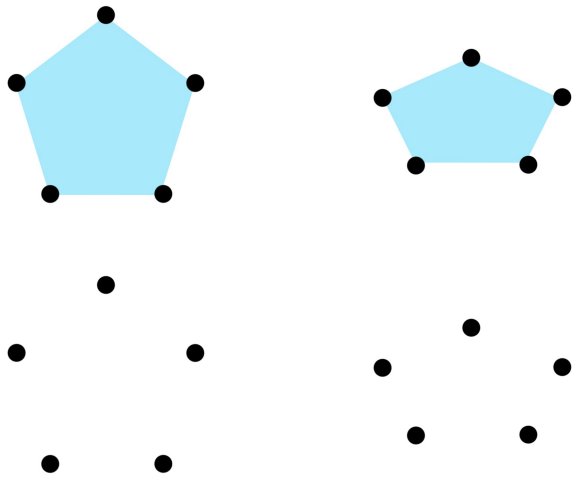


Figure 5.14: Issue on formation recognition caused by the point of view of the user. On the left, with the standard top-view it is easy to recognize and to understand the coordination objective. Instead, on the right, the view that the users had during the experiment: the formation is hard to recognize due to the viewing angle.

identified as the hardest to recognize (71.4% of the users answered coverage to the question *Which coordination objective was the hardest to recognize?*, and the remaining answered formation, confirming even more the preference on consensus).

Formation might seem as the easiest coordination objective to recognize, but it turned out to be the most difficult in this user study. This is due probably to the time needed to create the formation and the high dispersion of the robots when they perform this coordination objective. In addition the point of view is very different from the point of view with which formation control is usually presented, namely top-view (see Fig. 5.14). In fact, with the position of the camera inside the virtual environment (as reported in Fig. 5.2) and sharing the environment it is difficult to recognize any shape. Differently, consensus was easy to recognize because the spatial occupancy is easy to identify.

5.5.4 Conclusions

In this section we further investigated the legibility of multi-robot systems. We carried out a user study to answer to the question "Can a multi-robot system be legible, considering Definition 5.2.1?". In the experiment we investigated both the legibility with respect to the spatial goals and the coordination objectives.

About the ability of communicating the spatial goal, i.e., the first part of Definition 5.2.1, we can merge the information gathered in this section and in the previous Section 5.3 and 5.4. From all these results, we can state that, to be able to communicate the goal of a multi-robot system, it is important to use a straight trajectory and not to spread the robots. Spreading the robots in the environment could be due by high dispersion caused by strong artificial potential fields, or by a big variance of the Gaussian probability density function of the coverage control, or again by the initial procedure to create the shape in formation control. The more cohesive the group is along the path towards the goal, the more legible the motion is for the human observer.

Regarding the communication of a coordination objective, i.e., the second part of Definition 5.2.1, it is important to highlight the characteristics of the desired behavior. In fact, for consensus its inherent characteristic of agglomeration was easily identifiable. Thus, consensus was the easiest coordination objective to recognize. Instead, in the case of formation control, the major feature is the geometry of the formation. If this geometry does not emerge quickly enough, the identification of the coordination objective is significantly hindered. To improve the legibility of this coordination objective we would probably need to create an understandable formation even with the different point of view typical of proximal interaction. Further investigations are needed to understand the relationship between legibility and the shape of the formation. In addition, working towards the legibility of multi-robot systems executing complex behaviors will imply balancing the legibility aspect with the performance of the overall system.

From the results of the user studies reported in this chapter, we can establish that multi-robot systems are legible according to Definition 5.2.1. In fact, multi-robot systems can communicate effectively their spatial goal and their coordination objective.

Chapter 6

Main Findings and Future Works

6.1 Main Findings

The aim of this thesis was to take a step further multi-robot systems towards real world applications. Some bricks were added to the control structure that must be built to enable this type of systems to be deployed in real applications.

In Chapter 2 a novel connectivity maintenance method was proposed. The possibility to communicate is a crucial aspect for the deployment of these systems. We focused on the possibility of enabling the multi-robot system to maintain connectivity in a flexible way, and avoiding to introduce a huge restriction with respect to the main task that the system has to accomplish. We implemented this connectivity maintenance method along the lines of the global approach, maintaining the algebraic connectivity greater than zero. The proposed method was validated in simulations and on real robots. In the implementation on the hardware we faced the issue of the presence of delay in the communication network. We solved the issue introducing a heuristic to counterbalance the presence of delay. In addition, we dealt with the possibility of coming across some particular condition in which the proposed method was not well-defined. In fact, the proposed method was not well-defined when algebraic connectivity has multiplicity greater than one. Despite being an uncommon condition, it can happen and, hence, we investigated a solution that makes the proposed method robust also with respect to this condition. Also this improvement was tested in simulations and on real robots.

In Chapter 3 we aimed at removing a major limitations to one of the most common control strategies for multi-robot systems: coverage control. In fact, one of the biggest limitations of this control law, as presented in literature, is the fact that the robots are assumed to have an unlimited sensing range. Instead, we explicitly take into account the limited sensing range of the robots, and we do not require any communication network among the robots. The method was validated from a mathematical point of view, and tested on real robots.

In Chapter 4 we proposed a passivity-based control law that enables the multi-robot system to be robustly stable with respect to the uncertainty that it can encounter in real world applications. The method allows the system to be minimally modified, always ensuring passivity. We verified the proposed methodology both in

simulations and on real robots.

In Chapter 5 we investigated the possibility for multi-robot systems to communicate some information in an implicit manner. This possibility will enable these systems to be integrated in environments where the users can collaborate together with the robots and communicate in an efficient way. We introduced the definition of legibility for multi-robot systems, which is a quite unexplored topic in the literature. For legibility of a multi-robot system, we mean the ability of a group of robots to communicate to the user its spatial goal and/or its coordination objective. We investigated different control strategies, which modify the microscopic and macroscopic behavior of the system, in such a way to understand whether it was possible to communicate such information implicitly. For understanding this and to validate the proposed control strategies, we conducted a series of user study and we analyzed the data with the standard statistical tools.

6.2 Future Works

A lot of improvements and step forwards have yet to be done in order to fully take advantage of multi-robot systems in our every-day life. These systems will improve the efficiency of works and will help us in a lot of different tasks.

About the connectivity maintenance method proposed in Chapter 2, we will need to test its effectiveness also in the presence of noisy measurements and unstable communication. In fact, the real experiments we conducted so far are preliminary tests in controlled environment, e.g., with a motion capture system that provides the relative positions of the robots. In order to fully implement this method for a real world applications we need to test, and eventually to modify it, in more complex environments. In addition, it would be interesting to tune and to modify during the task both the threshold and the extended class $\mathcal{K}$ function, in such a way to obtain a flexible and even more optimized connectivity maintenance method. Another interesting development, from the control theory point of view, would be to consider a more complex model of the robots, e.g., to consider the non-holonomic constraints for wheeled robots, or to include the dynamic effects.

For the coverage control reported in Chapter 3, we think that it is a very powerful control law and that it can be further developed also for exploration tasks, even in the presence of unknown non-convex environments. In fact, being able to deploy a group of robots for different tasks using the same control law would be very effective. This would simplify a lot the tuning process and the required computing power of the single robot will be limited, which consequently will reduce the cost of the system. In addition, we will consider a group of robots characterized by heterogeneous limited sensing capabilities, which will enable even more complex tasks.

Along the lines of the future works for the connectivity maintenance method, also for the passivity-based control presented in Chapter 4 we will investigate the possibility of tuning the extended class $\mathcal{K}$ function based on how the energy inside the tank should be used. This will allow the system to be more flexible and to manage the energy in a wiser manner.

About the study on legibility of multi-robot systems presented in Chapter 5, a lot of improvements should be done. For example, it would be interesting to analyze if it is possible to infer a general model that maps the variables of a given control law to the legibility of the system. We presented a preliminary work on this aspect in [146]. Another important development will be to explore if explicit communication outperforms implicit communication, and by how much. Moreover, to take into account the perspective of real world collaboration, we will need to test the effectiveness of this communication during a real collaboration task, and to test whether it is more effective alone or in combination with explicit communication. This additional works will increase the ability of multi-robot systems to communicate with users, hence, they will become more user-friendly and accessible to everyone.

Bibliography

[1] Robotics: Latest trends & growth areas in the post pandemic world. https://analyticsindiamag.com/robotics-latest-trends-growth-areas-in-the-post-pandemic-world/. [Online; accessed 22-10-2021].

[2] Industry trends and market potential – what's next? https://www.automate.org/industry-insights/industry-trends-and-market-potential-what-s-next. [Online; accessed 22-10-2021].

[3] Robotics outlook 2030: How intelligence and mobility will shape the future. https://www.bcg.com/publications/2021/how-intelligence-and-mobility-will-shape-the-future-of-the-robotics-industry. [Online; accessed 22-10-2021].

[4] Henrik Christensen, Nancy Amato, Holly Yanco, Maja Mataric, Howie Choset, Ann Drobnis, Ken Goldberg, Jessy Grizzle, Gregory Hager, John Hollerbach, et al. A roadmap for us robotics–from internet to robotics 2020 edition. *Foundations and Trends® in Robotics*, 8(4):307–424, 2021.

[5] Mordor Intelligence. *Global Robotics Market*, volume 1. Mordord Intelligence, 2020.

[6] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008.

[7] Y Uny Cao, Andrew B Kahng, and Alex S Fukunaga. Cooperative mobile robotics: Antecedents and directions. In *Robot colonies*, pages 7–27. Springer, 1997.

[8] Zhi Yan, Nicolas Jouandeau, and Arab Ali Cherif. A survey and analysis of multi-robot coordination. *Int. Journal of Advanced Robotic Systems*, 10(12):399, 2013.

[9] Wolfram Burgard, Mark Moors, Cyrill Stachniss, and Frank E Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386, 2005.

[10] Jorge Cortés. Coverage optimization and spatial load balancing by robotic sensor networks. *IEEE Transactions on Automatic Control*, 55(3):749–754, 2010.

[11] Matthew Dunbabin and Lino Marques. Robots for environmental monitoring: Significant advancements and applications. *IEEE Robotics & Automation Magazine*, 19(1):24–39, 2012.

[12] Jingjin Yu, Mac Schwager, and Daniela Rus. Correlated orienteering problem and its application to persistent monitoring tasks. *IEEE Transactions on Robotics*, 32(5):1106–1118, 2016.

[13] David Portugal and Rui Rocha. A survey on multi-robot patrolling algorithms. In *Doctoral conference on computing, electrical and industrial systems*, pages 139–146. Springer, 2011.

[14] Fabio Pasqualetti, Antonio Franchi, and Francesco Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE Transactions on Robotics*, 28(3):592–606, 2012.

[15] Valerio Digani, Lorenzo Sabattini, Cristian Secchi, and Cesare Fantuzzi. Ensemble coordination approach in multi-agv systems applied to industrial warehouses. *IEEE Transactions on Automation Science and Engineering*, 12(3):922–934, 2015.

[16] Raffaello D'Andrea. Guest editorial: A revolution in the warehouse: A retrospective on kiva systems and the grand challenges ahead. *IEEE Transactions on Automation Science and Engineering*, 9(4):638–639, 2012.

[17] Samira Hayat, Evşen Yanmaz, Timothy X Brown, and Christian Bettstetter. Multi-objective uav path planning for search and rescue. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 5569–5574. IEEE, 2017.

[18] Joseph L Baxter, EK Burke, Jonathan M Garibaldi, and Mark Norman. Multi-robot search and rescue: A potential field based approach. In *Autonomous Robots and Agents*, pages 9–16. Springer, 2007.

[19] Antonio Barrientos, Julian Colorado, Jaime del Cerro, Alexander Martinez, Claudio Rossi, David Sanz, and Joao Valente. Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28(5):667–689, 2011.

[20] Vikram Puri, Anand Nayyar, and Linesh Raja. Agriculture drones: A modern breakthrough in precision agriculture. *Journal of Statistics and Management Systems*, 20(4):507–518, 2017.

[21] Pedro Santana, José Barata, Hildebrando Cruz, António Mestre, Joao Lisboa, and Luís Flores. A multi-robot system for landmine detection. In *Proc. IEEE Conf. on Emerging Technologies and Factory Automation*, volume 1, pages 8–pp. IEEE, 2005.

[22] Charles Lesire, Guillaume Infantes, Thibault Gateau, and Magali Barbier. A distributed architecture for supervision of autonomous multi-robot missions. *Autonomous Robots*, 40(7):1343–1362, 2016.

[23] Gautham P Das, Thomas M McGinnity, Sonya A Coleman, and Laxmidhar Behera. A distributed task allocation algorithm for a multi-robot system in healthcare facilities. *Journal of Intelligent & Robotic Systems*, 80(1):33–58, 2015.

[24] Michael B Cohn, Lara S Crawford, Jeffrey M Wendlandt, and S Shartkar Sastry. Surgical applications of milli-robots. *Journal of Robotic Systems*, 12(6):401–416, 1995.

[25] R. Soukieh, I. Shames, and B. Fidan. Obstacle avoidance of non-holonomic unicycle robots based on fluid mechanical modeling. In *Proc. European Control Conf. (ECC)*, 2009.

[26] D. Lee, A. Franchi, H.I. Son, C. Ha, H.H. Bulthoff, and P. Robuffo Giordano. Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles. *IEEE/ASME Transactions on Mechatronics*, 18(4):1334–1345, Aug 2013.

[27] Jacopo Banfi, Alberto Quattrini Li, Ioannis Rekleitis, Francesco Amigoni, and Nicola Basilico. Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Autonomous Robots*, 42(4):875–894, 2018.

[28] Yiannis Kantaros and Michael M Zavlanos. Distributed intermittent connectivity control of mobile robot networks. *IEEE Transactions on Automatic Control*, 62(7):3109–3121, 2016.

[29] Lorenzo Sabattini, Beatrice Capelli, Cesare Fantuzzi, and Cristian Secchi. Teleoperation of multi-robot systems to relax topological constraints. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 4558–4564. IEEE, 2020.

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

[31] Dimos V Dimarogonas and Karl H Johansson. Decentralized connectivity maintenance in mobile networks with bounded inputs. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1507–1512. IEEE, 2008.

[32] Amir Ajorlou, Ahmadreza Momeni, and Amir G Aghdam. A class of bounded distributed control strategies for connectivity preservation in multi-agent systems. *IEEE Transactions on Automatic Control*, 55(12):2828–2833, 2010.

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

[34] Lorenzo Sabattini, Nikhil Chopra, and Cristian Secchi. Decentralized connectivity maintenance for cooperative control of mobile robotic systems. *The Int. Journal of Robotics Research*, 32(12):1411–1423, 2013.

[35] Xiangpeng Li, Dong Sun, and Jie Yang. A bounded controller for multirobot navigation while maintaining network connectivity in the presence of obstacles. *Automatica*, 49(1):285–292, 2013.

[36] Koresh Khateri, Mahdi Pourgholi, Mohsen Montazeri, and Lorenzo Sabattini. A comparison between decentralized local and global methods for connectivity maintenance of multi-robot networks. *IEEE Robotics and Automation Letters*, 4(2):633–640, 2019.

[37] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *Proc. 53rd IEEE Conf. on Decision and Control (CDC)*, pages 6271–6278. IEEE, 2014.

[38] Xiangru Xu, Thomas Waters, Daniel Pickem, Paul Glotfelter, Magnus Egerstedt, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds. In *Proc. IEEE Conf. on Control Technology and Applications (CCTA)*, pages 1769–1775. IEEE, 2017.

[39] Chiara Talignani Landi, Federica Ferraguti, Silvia Costi, Marcello Bonfè, and Cristian Secchi. Safety barrier functions for human-robot interaction with industrial manipulators. In *Proc. 18th European Control Conf. (ECC)*, pages 2565–2570. IEEE, 2019.

[40] Li Wang, Aaron D Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[41] Magnus Egerstedt, Jonathan N Pauli, Gennaro Notomista, and Seth Hutchinson. Robot ecology: Constraint-based control design for long duration autonomy. *Annual Reviews in Control*, 2018.

[42] Wenhao Luo and Katia Sycara. Voronoi-based coverage control with connectivity maintenance for robotic sensor networks. In *Proc. Int. Symp. on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2019.

[43] Luca Siligardi, Jacopo Panerati, Marcel Kaufmann, Marco Minelli, Cinara Ghedini, Giovanni Beltrame, and Lorenzo Sabattini. Robust area coverage with connectivity maintenance. In *Proc. Int. Conf. on Robotics and Automation (ICRA)*, pages 2202–2208. IEEE, 2019.

[44] Francesco Bullo, Jorge Cortés, and Sonia Martinez. *Distributed control of robotic networks: a mathematical approach to motion coordination algorithms*. Princeton University Press, 2009.

[45] Norman Biggs, Norman Linstead Biggs, and Biggs Norman. *Algebraic graph theory*. Cambridge university press, 1993.

[46] Chris Godsil and Gordon F Royle. *Algebraic graph theory*, volume 207. Springer Science & Business Media, 2001.

[47] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.

[48] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[49] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[50] A. D. Ames, S Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada. Control barrier functions: Theory and applications. In *Proc. European Control Conf. (ECC)*, pages 3420–3431, June 2019.

[51] Hassan K Khalil. Nonlinear systems. *Upper Saddle River*, 2002.

[52] Beatrice Capelli and Lorenzo Sabattini. Connectivity maintenance: Global and optimized approach through control barrier functions. In *Proc. Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2020.

[53] A. Gasparri, L. Sabattini, and G. Ulivi. Bounded control law for global connectivity maintenance in cooperative multi-robot systems. *IEEE Trans. on Robotics*, 33(3):700–717, June 2017.

[54] Camille Poignard, Tiago Pereira, and Jan Philipp Pade. Spectra of laplacian matrices of weighted graphs: structural genericity properties. *SIAM Journal on Applied Mathematics*, 78(1):372–394, 2018.

[55] Jan R Magnus. On differentiating eigenvalues and eigenvectors. *Econometric Theory*, 1(2):179–191, 1985.

[56] P. Glotfelter, J. Cortés, and M. Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE Control Systems Letters*, 1(2):310–315, 2017.

[57] Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[58] María Santos, Yancy Diaz-Mercado, and Magnus Egerstedt. Coverage control for multirobot teams with heterogeneous sensing capabilities. *IEEE Robotics and Automation Letters*, 3(2):919–925, 2018.

[59] Sung G Lee, Yancy Diaz-Mercado, and Magnus Egerstedt. Multirobot control using time-varying density functions. *IEEE Transactions on Robotics*, 31(2):489–493, 2015.

[60] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[61] B. Capelli, H. Fouad, G. Beltrame, and L. Sabattini. Decentralized connectivity maintenance with time delays using control barrier functions. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021.

[62] Cristian Secchi, Stefano Stramigioli, and Cesare Fantuzzi. *Control of interactive robotic interfaces: A port-Hamiltonian approach*, volume 29. Springer Science & Business Media, 2007.

[63] Mrdjan Jankovic. Control barrier functions for constrained control of linear systems with input delay. In *Proc. American Control Conf. (ACC)*, pages 3316–3321. IEEE, 2018.

[64] Gábor Orosz and Aaron D Ames. Safety functionals for time delay systems. In *Proc. American Control Conf. (ACC)*, pages 4374–4379. IEEE, 2019.

[65] Carlo Pinciroli and Giovanni Beltrame. Swarm-oriented programming of distributed robot networks. *Computer*, 49(12):32–41, 2016.

[66] Carlo Pinciroli, Adam Lee-Brown, and Giovanni Beltrame. A tuple space for data sharing in robot swarms. In *Proc. 9th EAI Int. Conf. on Bio-inspired Information and Communication Technologies (formerly BIONETICS)*, pages 287–294, 2016.

[67] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, et al. Argos: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, 2012.

[68] Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, and Giuseppe Oriolo. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[69] Steven Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2(1-4):153, 1987.

[70] P. Ong, B. Capelli, L. Sabattini, and J. Cortés. Network connectivity maintenance via nonsmooth control barrier functions. In *Proc. IEEE Int. Conf. on Decision and Control (CDC)*, 2021.

[71] P. Ong, B. Capelli, L. Sabattini, and J. Cortés. Nonsmooth control barrier function design of continuous constraints for network connectivity maintenance. *Submitted to SAGE Int. Journal of Robotics Research (IJRR)*, 2021.

[72] M. Franceschelli, S. Martini, M. Egerstedt, A. Bicchi, and A. Giua. Observability and controllability verification in multi-agent systems through decentralized laplacian spectrum estimation. In *Proc. IEEE Conf. on Decision and Control (CDC)*, pages 5775–5780. IEEE, December 2010.

[73] M. D. Schuresko and J. Cortés. Distributed motion constraints for algebraic connectivity of robotic networks. *Journal of Intelligent and Robotic Systems*, 56(1):99–126, 2009.

[74] B. J. Morris, M. J. Powell, and A. D. Ames. Continuity and smoothness properties of nonlinear optimization-based feedback controllers. In *Proc. IEEE Conf. on Decision and Control (CDC)*, pages 151–158, Osaka, Japan, Dec 2015.

[75] C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Studies in Economic Theory. Springer, 1999.

[76] R. A. Freeman and P. V. Kototovic. *Robust Nonlinear Control Design: State-space and Lyapunov Techniques*. Birkhauser Boston Inc., Cambridge, MA, USA, 1996.

[77] Daniel Shevitz and Brad Paden. Lyapunov stability theory of nonsmooth systems. *IEEE Transactions on Automatic Control*, 39(9):1910–1914, 1994.

[78] F. H. Clarke. *Optimization and Nonsmooth Analysis*. Canadian Mathematical Society Series of Monographs and Advanced Texts. Wiley, 1983.

[79] Ashish Cherukuri, Bahman Gharesifard, and Jorge Cortes. Saddle-point dynamics: conditions for asymptotic stability of saddle points. *SIAM Journal on Control and Optimization*, 55(1):486–511, 2017.

[80] Jorge Cortés and Simon K Niederländer. Distributed coordination for nonsmooth convex optimization via saddle-point dynamics. *Journal of Nonlinear Science*, 29(4):1247–1272, 2019.

[81] G. Oriolo, A. De Luca, and M. Vendittelli. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, 10(6):835–852, 2002.

[82] F. Pratissoli, B. Capelli, and L. Sabattini. On coverage control for limited range multi-robot systems. In *Submitted to IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021.

[83] Jorge Cortés and Magnus Egerstedt. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6):495–503, 2017.

[84] Jorge Cortes, Sonia Martinez, and Francesco Bullo. Spatially-distributed coverage optimization and control with limited-range interactions. *ESAIM: Control, Optimisation and Calculus of Variations*, 11(4):691–719, 2005.

[85] Qiang Du, Maria Emelianenko, and Lili Ju. Convergence of the lloyd algorithm for computing centroidal voronoi tessellations. *SIAM Journal on Numerical Analysis*, 44(1):102–119, 2006.

[86] Enrique Teruel, Rosario Aragues, and Gonzalo López-Nicolás. A practical method to cover evenly a dynamic region with a swarm. *IEEE Robotics and Automation Letters*, 6(2):1359–1366, 2021.

[87] Mac Schwager, James McLurkin, and Daniela Rus. Distributed coverage control with sensory feedback for networked robots. In *Robotics: Science and Systems*, pages 49–56, 2006.

[88] KR Guruprasad and Prithviraj Dasgupta. Distributed voronoi partitioning for multi-robot systems with limited range sensors. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3546–3552. IEEE, 2012.

[89] Boulat A Bash and Peter J Desnoyers. Exact distributed voronoi cell computation in sensor networks. In *Proc. 6th Int. Conf. on Information Processing in Sensor Networks*, pages 236–243, 2007.

[90] Waleed Alsalih, Kamrul Islam, Yurai Núñez Rodríguez, and Henry Xiao. Distributed voronoi diagram computation in wireless sensor networks. In *SPAA*, page 364, 2008.

[91] Qun Li and Daniela Rus. Navigation protocols in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 1(1):3–35, 2005.

[92] Min Cao and Christoforos Hadjicostis. Distributed algorithms for voronoi diagrams and application in ad-hoc networks. *UIUC Coordinated Science Laboratory, Tech. Rep. UILU-ENG-03-2222, DC-210*, 2003.

[93] Stephanie Kemna, John G Rogers, Carlos Nieto-Granda, Stuart Young, and Gaurav S Sukhatme. Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2124–2130. IEEE, 2017.

[94] Chenlong He, Zuren Feng, and Zhigang Ren. Distributed algorithm for voronoi partition of wireless sensor networks with a limited sensing range. *Sensors*, 18(2):446, 2018.

[95] Farsam Farzadpour, Xuebo Zhang, Xiang Chen, and Tong Zhang. On performance measurement for a heterogeneous planar field sensor network. In *Proc. IEEE Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, pages 166–171. IEEE, 2017.

[96] Max Rudolph, Sean Wilson, and Magnus Egerstedt. Range limited coverage control using air-ground multi-robot teams. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, 2021.

[97] Katie Laventall and Jorge Cortés. Coverage control by multi-robot networks with limited-range anisotropic sensory. *Int. Journal of Control*, 82(6):1113–1121, 2009.

[98] Atsuyuki Okabe. Spatial tessellations. *International Encyclopedia of Geography: People, the Earth, Environment and Technology: People, the Earth, Environment and Technology*, pages 1–11, 2016.

[99] Naoki Hayashi, Kohei Segawa, and Shigemasa Takai. 2d voronoi coverage control with gaussian density functions by line integration. *SICE Journal of Control, Measurement, and System Integration*, 10(2):110–116, 2017.

[100] B. Capelli, C. Secchi, and L. Sabattini. Passivity and control barrier functions: Optimizing the use of energy. *IEEE Robotics and Automation Letters (RA-L)*, 2021.

[101] Romeo Ortega, Julio Antonio Loría Perez, Per Johan Nicklasson, and Hebertt J Sira-Ramirez. *Passivity-based control of Euler-Lagrange systems: mechanical, electrical and electromechanical applications*. Springer Science & Business Media, 2013.

[102] Arjan J Van der Schaft and AJ Van Der Schaft. *L2-gain and passivity techniques in nonlinear control*, volume 2. Springer, 2000.

[103] Gianluca Antonelli, Filippo Arrichiello, Fabrizio Caccavale, and Alessandro Marino. Decentralized time-varying formation control for multi-robot systems. *The Int. Journal of Robotics Research*, 33(7):1029–1043, 2014.

[104] F. Ferraguti, C. Talignani Landi, L. Sabattini, M. Bonfé, C. Fantuzzi, and C. Secchi. A variable admittance control strategy for stable physical human-robot interaction. *The Int. Journal of Robotics Research (SAGE)*, 38(6):747–765, May 2019.

[105] Jee-Hwan Ryu, Dong-Soo Kwon, and B. Hannaford. Control of a flexible manipulator with noncollocated feedback: time-domain passivity approach. *IEEE Transactions on Robotics*, 20(4):776–780, 2004.

[106] Jee-Hwan Ryu, Jordi Artigas, and Carsten Preusche. A passive bilateral control scheme for a teleoperator with time-varying communication delay. *Mechatronics*, 20(7):812 – 823, 2010. Special Issue on Design and Control Methodologies in Telerobotics.

[107] M. De Stefano, R. Balachandran, and C. Secchi. A passivity-based approach for simulating satellite dynamics with robots: Discrete-time integration and time-delay compensation. *IEEE Transactions on Robotics*, 36(1):189–203, 2020.

[108] D. Lee and K. Huang. Passive-set-position-modulation framework for interactive robotic systems. *IEEE Transactions on Robotics*, 26(2):354–369, 2010.

[109] Michel Franken, Stefano Stramigioli, Sarthak Misra, Cristian Secchi, and Alessandro Macchelli. Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency. *IEEE Transactions on Robotics*, 27(4):741–756, 2011.

[110] Erfan Shahriari, Aljaž Kramberger, Andrej Gams, Aleš Ude, and Sami Haddadin. Adapting to contacts: Energy tanks and task energy for passivity-based dynamic movement primitives. In *Proc. IEEE-RAS 17th Int. Conf. on Humanoid Robotics (Humanoids)*, pages 136–142. IEEE, 2017.

[111] Federica Ferraguti, Nicola Preda, Auralius Manurung, Marcello Bonfe, Olivier Lambercy, Roger Gassert, Riccardo Muradore, Paolo Fiorini, and Cristian Secchi. An energy tank-based interactive control architecture for autonomous and teleoperated robotic surgery. *IEEE Transactions on Robotics*, 31(5):1073–1088, 2015.

[112] Paolo Robuffo Giordano, Antonio Franchi, Cristian Secchi, and Heinrich H Bülthoff. A passivity-based decentralized strategy for generalized connectivity maintenance. *The Int. Journal of Robotics Research*, 32(3):299–323, 2013.

[113] Lorenzo Sabattini, Cristian Secchi, Beatrice Capelli, and Cesare Fantuzzi. Passivity preserving force scaling for enhanced teleoperation of multirobot systems. *IEEE Robotics and Automation Letters*, 3(3):1925–1932, 2018.

[114] Giuseppe Riggio, Cesare Fantuzzi, and Cristian Secchi. On the use of energy tanks for multi-robot interconnection. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3738–3743. IEEE, 2018.

[115] Cristian Secchi and Federica Ferraguti. Energy optimization for a robust and flexible interaction control. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1919–1925. IEEE, 2019.

[116] Federico Benzi and Cristian Secchi. An optimization approach for a robust and flexible control in collaborative applications. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2021.

[117] Gennaro Notomista, Xiaoyi Cai, Junya Yamauchi, and Magnus Egerstedt. Passivity-based decentralized control of multi-robot systems with delays using control barrier functions. In *Proc. Int. Symp. on Multi-Robot and Multi-Agent Systems (MRS)*, pages 231–237. IEEE, 2019.

[118] Naomi Ehrich Leonard and Edward Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proc. 40th IEEE Conf. on Decision and Control (CDC)*, volume 3, pages 2968–2973. IEEE, 2001.

[119] Dimos V Dimarogonas and Kostas J Kyriakopoulos. Connectedness preserving distributed swarm aggregation for multiple kinematic robots. *IEEE Transactions on Robotics*, 24(5):1213–1223, 2008.

[120] L. Sabattini, C. Secchi, and C. Fantuzzi. Arbitrarily shaped formations of mobile robots: artificial potential fields and coordinate transformation. *Autonomous Robots (Springer)*, 30(4):385–397, may 2011.

[121] Jiayi Sun, Jun Tang, and Songyang Lao. Collision avoidance for cooperative uavs with optimized artificial potential field algorithm. *IEEE Access*, 5:18382–18390, 2017.

[122] Melanie Schranz, Martina Umlauft, Micha Sende, and Wilfried Elmenreich. Swarm robotic behaviors and current applications. *Frontiers in Robotics and AI*, 7:36, 2020.

[123] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *Proc. 8th Annu. ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pages 301–308. IEEE Press, 2013.

[124] Antonio Franchi, Paolo Robuffo Giordano, Cristian Secchi, Hyoung Il Son, and Heinrich H Bülthoff. A passivity-based decentralized approach for the bilateral teleoperation of a group of uavs with switching topology. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 898–905. IEEE, 2011.

[125] Jawad Nagi, Alessandro Giusti, Luca M Gambardella, and Gianni A Di Caro. Human-swarm interaction using spatial gestures. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 3834–3841. IEEE, 2014.

[126] Andreas Kolling, Phillip Walker, Nilanjan Chakraborty, Katia Sycara, and Michael Lewis. Human interaction with robot swarms: A survey. *IEEE Transactions on Human-Machine Systems*, 46(1):9–26, 2016.

[127] Günter Niemeyer, Carsten Preusche, Stefano Stramigioli, and Dongjun Lee. Telerobotics. In *Springer handbook of robotics*, pages 1085–1108. Springer, 2016.

[128] Robin R Murphy. Trial by fire [rescue robots]. *IEEE Robotics & Automation Magazine*, 11(3):50–61, 2004.

[129] Gerd Hirzinger, Klaus Landzettel, Bernhard Brunner, Max Fischer, Carsten Preusche, Detlef Reintsema, Alin Albu-Schäffer, Günter Schreiber, and Bernhard-Michael Steinmetz. Dlr's robotics technologies for on-orbit servicing. *Advanced Robotics*, 18(2):139–174, 2004.

[130] Jean Scholtz, Mary Theofanos, and Brian Antonishek. Development of a test bed for evaluating human-robot performance for explosive ordnance disposal robots. In *Proc. 1st ACM SIGCHI/SIGART Conf. on Human-Robot Interaction*, pages 10–17, 2006.

[131] Marco Minelli, Federica Ferraguti, Nicola Piccinelli, Riccardo Muradore, and Cristian Secchi. An energy-shared two-layer approach for multi-master-multi-slave bilateral teleoperation systems. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 423–429. IEEE, 2019.

[132] Federica Ferraguti, Chiara Talignani Landi, Silvia Costi, Marcello Bonfè, Saverio Farsoni, Cristian Secchi, and Cesare Fantuzzi. Safety barrier functions and multi-camera tracking for human–robot shared environment. *Robotics and Autonomous Systems*, 124:103388, 2020.

[133] Przemyslaw A Lasota, Terrence Fong, Julie A Shah, et al. A survey of methods for safe human-robot interaction. *Foundations and Trends® in Robotics*, 5(4):261–349, 2017.

[134] Daniel Szafir, Bilge Mutlu, and Terry Fong. Communicating directionality in flying robots. In *Proc. 10th Annu. ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pages 19–26. ACM, 2015.

[135] Alyxander David May, Christian Dondrup, and Marc Hanheide. Show me your moves! conveying navigation intention of a mobile robot to humans. In *Proc. Eur. Conf. on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2015.

[136] Aaron St Clair and Maja Mataric. How robot verbal feedback can improve team performance in human-robot task collaborations. In *Proc. 10th Annu. ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pages 213–220. ACM, 2015.

[137] Fabrizio Ghiringhelli, Jérôme Guzzi, Gianni A Di Caro, Vincenzo Caglioti, Luca M Gambardella, and Alessandro Giusti. Interactive augmented reality for understanding and analyzing multi-robot systems. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1195–1201. IEEE, 2014.

[138] Jayam Patel, Yicong Xu, and Carlo Pinciroli. Mixed-granularity human-swarm interaction. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1059–1065. IEEE, 2019.

[139] Tathagata Chakraborti, Sarath Sreedharan, Anagha Kulkarni, and Subbarao Kambhampati. Projection-aware task planning and execution for human-in-the-loop operation of robots in a mixed-reality workspace. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4476–4482. IEEE, 2018.

[140] Maartje MA De Graaf and Bertram F Malle. People's explanations of robot behavior subtly reveal mental state inferences. In *Proc. 14th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pages 239–248. IEEE, 2019.

[141] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communication of intent in assistive free flyers. In *Proc. ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pages 358–365. ACM, 2014.

[142] Anca Dragan and Siddhartha Srinivasa. Generating legible motion. In *Proc. 9th Annu. Robotics: Science and Systems (RSS)*, 2013.

[143] Anca D Dragan, Shira Bauman, Jodi Forlizzi, and Siddhartha S Srinivasa. Effects of robot motion on human-robot collaboration. In *Proc. 10th ACM/IEEE Int. Conf. on Human-Robot Interaction (HRI)*, pages 51–58. IEEE, 2015.

[144] Beatrice Capelli, Valeria Villani, Cristian Secchi, and Lorenzo Sabattini. Understanding multi-robot systems: on the concept of legibility. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 7355–7361. IEEE, 2019.

[145] Beatrice Capelli, Cristian Secchi, and Lorenzo Sabattini. Communication through motion: legibility of multi-robot systems. In *Proc. Int. Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 126–132. IEEE, 2019.

[146] Beatrice Capelli and Lorenzo Sabattini. Human-friendly multi-robot systems: Legibility analysis. In *Proc. Int. Workshop on Human-Friendly Robotics (HFR)*, pages 15–28. Springer, 2019.

[147] Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explicability and predictability for robot task planning. In *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1313–1320. IEEE, 2017.

[148] Anagha Kulkarni, Siddharth Srivastava, and Subbarao Kambhampati. A unified framework for planning in adversarial and cooperative environments. In *Proc. AAAI Conf. on Artificial Intelligence*, volume 33, pages 2479–2487, 2019.

[149] Tathagata Chakraborti, Anagha Kulkarni, Sarath Sreedharan, David E Smith, and Subbarao Kambhampati. Explicability? legibility? predictability? transparency? privacy? security? the emerging landscape of interpretable agent behavior. In *Proc. 29th Int. Conf. on Automated Planning and Scheduling*, volume 29, pages 86–96, 2019.

[150] Cynthia Breazeal. Toward sociable robots. *Robotics and Autonomous Systems*, 42(3):167–175, 2003. Socially Interactive Robots.

[151] Griffin Dietz, Jane L. E, Peter Washington, Lawrence H. Kim, and Sean Follmer. Human perception of swarm robot motion. In *Proc. CHI Conf. Extended Abstracts on Human Factors in Computing Systems*, pages 2520–2527, Denver, Colorado, 2017.

[152] F. Levillain, D. St-Onge, E. Zibetti, and G. Beltrame. More than the sum of its parts: Assessing the coherence and expressivity of a robotic swarm. In *Proc. IEEE Int. Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 583–588, Aug 2018.

[153] María Santos and Magnus Egerstedt. From motions to emotions: Can the fundamental emotions be expressed in a robot swarm? *Int. Journal of Social Robotics*, pages 1–14, 2020.

[154] David St-Onge, Ulysse Côté-Allard, Kyrre Glette, Benoit Gosselin, and Giovanni Beltrame. Engaging with robotic swarms: Commands from expressive motion. *ACM Transactions on Human-Robot Interaction (THRI)*, 8(2):1–26, 2019.

[155] Lawrence H Kim and Sean Follmer. Generating legible and glanceable swarm robot motion through trajectory, collective behavior, and pre-attentive processing features. *ACM Transactions on Human-Robot Interaction (THRI)*, 10(3):1–25, 2021.

[156] Frank Papenmeier, Meike Uhrig, and Alexandra Kirsch. Human understanding of robot motion: The role of velocity and orientation. *Int. Journal of Social Robotics*, Oct 2018.

[157] Valeria Villani, Beatrice Capelli, and Lorenzo Sabattini. Use of virtual reality for the evaluation of human-robot interaction systems in complex scenarios. In *Proc. 27th IEEE Int. Symp. on Robot and Human Interactive Communication (RO-MAN)*, pages 422–427. IEEE, 2018.

[158] Douglas C Montgomery. *Design and analysis of experiments*. John wiley & sons, 2017.

[159] Behrang Keshavarz, Heiko Hecht, and Ben D Lawson. Visually induced motion sickness: causes, characteristics, and countermeasures. In *Handbook of Virtual Environments*, pages 652–703. CRC Press, 2014.

[160] Tamar Flash and Neville Hogan. The coordination of arm movements: an experimentally confirmed mathematical model. *Journal of Neuroscience*, 5(7):1688–1703, 1985.

[161] Stefan Glasauer, Markus Huber, Patrizia Basili, Alois Knoll, and Thomas Brandt. Interacting in time and space: Investigating human-human and human-robot joint action. In *Proc. 19th Annu. IEEE Int. Symp. in Robot and Human Interactive Communication (RO-MAN)*, pages 252–257. IEEE, 2010.

[162] John Lasseter. Principles of traditional animation applied to 3d computer animation. In *Proc. ACM Siggraph Computer Graphics*, volume 21, pages 35–44. ACM, 1987.

[163] Neil J Salkind. *Encyclopedia of research design*, volume 1. Sage, 2010.

[164] K Anders Ericsson and Herbert A Simon. Verbal reports as data. *Psychological review*, 87(3):215, 1980.

[165] K Anders Ericsson and Herbert A Simon. *Protocol analysis: Verbal reports as data.* the MIT Press, 1984.

[166] Emrah Akin Sisbot, Luis F Marin-Urias, Rachid Alami, and Thierry Simeon. A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883, 2007.

[167] Baptiste Busch, Jonathan Grizou, Manuel Lopes, and Freek Stulp. Learning legible motion from human-robot interactions. *Int. Journal of Social Robotics*, 9(5):765–779, 2017.

[168] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: a review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, March 2013.

[169] John Klingner, Anshul Kanakia, Nicholas Farrow, Dustin Reishus, and Nikolaus Correll. A stick-slip omnidirectional powertrain for low-cost swarm robotics: Mechanism, calibration, and control. In *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 846–851, 2014.

[170] Sean Wilson, Ruben Gameros, Michael Sheely, Matthew Lin, Kathryn Dover, Robert Gevorkyan, Matt Haberland, Andrea Bertozzi, and Spring Berman. Pheeno, a versatile swarm robotic research and education platform. *IEEE Robotics and Automation Letters*, 1(2):884–891, 2016.

[171] Sean Wilson, Paul Glotfelter, Li Wang, Siddharth Mayya, Gennaro Notomista, Mark Mote, and Magnus Egerstedt. The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems. *IEEE Control Systems Magazine*, 40(1):26–44, 2020.

[172] Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stéphane Magnenat, Jean christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proc. 9th Conf. on Autonomous Robot Systems and Competitions*, pages 59–65, 2009.

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

[174] J. Cortes, S. Martinez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006.

[175] J. Lin, A. S. Morse, and B. D. O. Anderson. The multi-agent rendezvous problem. part 1: The synchronous case. *SIAM Journal on Control and Optimization*, 46(6):2096–2119, December 2007.

[176] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[177] Changbin Yu, Brian D. O. Anderson, Soura Dasgupta, and Bariş Fidan. Control of minimally persistent formations in the plane. *SIAM Journal on Control and Optimization*, 48(1):206–233, 2009.

[178] Mehran Mesbahi and Magnus Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton Series in Applied Mathematics. Princeton University Press, 2010.

[179] Clayton Lewis. *Using the "thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY, 1982.