KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Electrical Engineering, specializing in Systems, Control and Robotics

Second cycle, 30 credits

Control barrier function-enabled human-in-the-loop control for multi-robot systems

Centralized and distributed approaches

VICTOR NAN FERNANDEZ-AYALA

Stockholm, Sweden, 2022

Control barrier function-enabled human-in-the-loop control for multi-robot systems

Centralized and distributed approaches

VICTOR NAN FERNANDEZ-AYALA

Master's Programme, Aerospace Engineering, 120 credits
Date: July 1, 2022

Supervisor: Xiao Tan
Examiner: Dimos V. Dimarogonas
School of Electrical Engineering and Computer Science
Swedish title: Kontrollbarriärfunktion som möjliggör mänsklig kontroll i kretsloppet för flerrobotsystem
Swedish subtitle: Centraliserade och distribuerade tillvägagångssätt

Abstract

Autonomous multi-robot systems have found many real-world applications in factory settings, rescue tasks and light shows. Albeit these successful applications, the multi-robot system is usually pre-programmed with limited flexibility for online adaptation. Having a human-in-the-loop feature would provide additional flexibility such as handling unexpected situations, detecting and correcting bad behaviours and supporting the automated decision making. In addition, it would also allow for an extra level of cooperation between the robots and the human that facilitates certain real-world tasks, for example in the agricultural sector.

Control barrier functions (CBFs), as a convenient modular-design tool, will be mainly explored. CBFs have seen a lot of development in recent years and extending them to the field of multi-robot systems is still new. In particular, creating an original distributed approach, instead of a centralized one, will be one of the key topics of this Master's thesis project.

In this thesis work, several multi-robot coordination protocols and safety constraints will be identified and these constraints will be enforced using a control barrier function-enabled mixer module. This module will take in the commands from both the planner and the human operator, prioritizing the commands from the human operator as long as the safety constraints are not violated. Otherwise, the mixer module will filter the commands and send out a safe alternative. The underlying multi-robot tasks are expected to be achieved whenever feasible. Simulations in ROS, Python and MATLAB environments are developed to experimentally assess the safety and optimality of the system, and experiments with real robots in a lab are performed to show the applicability of this algorithm.

Finally, a distributed approach to the mixer module has been developed, based on previous research and extended to allow for more versatility. This is of key importance since it would allow each robot to compute its own controller based on local information, making the multi-robot system both more robust and flexible to be deployed on real-world applications.

Keywords

Multi-robot systems, Human-in-the-loop control, Control barrier functions, Safety constraints, ROS Implementation

Sammanfattning

Autonoma multirobotsystem har fått många verkliga tillämpningar i fabriksmiljöer, räddningsuppdrag och ljusshower. Trots dessa framgångsrika tillämpningar är multirobotsystemet vanligtvis förprogrammerat med begränsad flexibilitet för anpassning online. En människa i loopen skulle ge ytterligare flexibilitet, t.ex. när det gäller att hantera oväntade situationer, upptäcka och korrigera dåliga beteenden och stödja det automatiska beslutsfattandet. Dessutom skulle det också möjliggöra en extra samarbetsnivå mellan robotarna och människan som underlättar vissa verkliga uppgifter, till exempel inom jordbrukssektorn.

Kontrollbarriärfunktioner (CBF), som ett bekvämt verktyg för modulbaserad utformning, kommer huvudsakligen att undersökas. CBF:er har utvecklats mycket under de senaste åren och det är fortfarande nytt att utvidga dem till flerrobotsystem. Att skapa ett originellt distribuerat tillvägagångssätt i stället för ett centraliserat kommer att vara ett av de viktigaste ämnena i detta examensarbete.

I detta examensarbete kommer flera samordningsprotokoll och säkerhetsbegränsningar för flera robotar att identifieras och dessa begränsningar kommer att upprätthållas med hjälp av en mixermodul med kontrollbarriärfunktion. Denna modul kommer att ta emot kommandon från både planeraren och den mänskliga operatören och prioritera kommandon från den mänskliga operatören så länge säkerhetsbegränsningarna inte överträds. I annat fall kommer mixermodulen att filtrera kommandona och skicka ut ett säkert alternativ. De underliggande multirobotuppgifterna förväntas uppnås närhelst det är möjligt. Simuleringar i ROS-, Python- och MATLAB-miljöerna utvecklas för att experimentellt bedöma systemets säkerhet och optimalitet, och experiment med riktiga robotar i ett labb utförs för att visa algoritmens tillämpbarhet.

Slutligen har ett distribuerat tillvägagångssätt för mixermodulen utvecklats, baserat på tidigare forskning och utökat för att möjliggöra större mångsidighet. Detta är av central betydelse eftersom det skulle göra det möjligt för varje robot att beräkna sin egen styrning utifrån lokal information, vilket gör systemet med flera robotar både mer robust och flexibelt för att kunna användas i verkliga tillämpningar.

Nyckelord

Multirobotsystem, Human-in-the-loop-kontroll, Kontrollera barriärfunktionerna, Säkerhetsbegränsningar, ROS-implementering

Acknowledgments

I would like to thank Gil Serrano for the help in setting up and using both the ROS environment and the NEXUS robot experiments, my supervisor, Xiao Tan, for the guidance on the theoretical aspects of the thesis and my examiner, Dimos V. Dimarogonas, for reviewing my work.

Stockholm, July 2022
Victor Nan Fernandez-Ayala

Contents

List of Figures

List of Tables

List of acronyms and abbreviations

CA	Collision Avoidance
CBF	Control Barrier Function
CM	Connectivity Maintenance
DCS	Division of Decision and Control Systems
HIL	Human-In-the-Loop
KTH	KTH Royal Institute of Technology
MAS	Multi-Agent System
QP	Quadratic Program
ROS	Robot Operating System
SML	Smart Mobility Lab
STL	Signal Temporal Logic
UAV	Unmanned Aerial Vehicle

List of Symbols Used

The following symbols will be later used within the body of the thesis.

α Extended class $\mathcal{K}$ function, ... page 13

$\mathcal{G}$ The communication graph of a MAS, ... page 10

$\mathcal{I}$ The nodes of a graph, ... page 10

μ The parameter representing the constraints on the explicit solution of the QP, see equation (2.10), ... page 14

$\mathscr{C}$ The safety set, see equation (2.6), ... page 12

$\mathfrak{f}, \mathfrak{g}$ The system functions of a MAS, see equation (2.3), ... page 10

$\mathfrak{f}_i, \mathfrak{g}_i$ The system functions of an individual agent, see equation (2.1), page 9

$\boldsymbol{A}, \boldsymbol{b}$ The vectorized linear parameters of the CBF constraint, see equation (4.1), ... page 25

$\boldsymbol{a}_i, b_i$ The linear parameters of the CBF constraint, see equation (2.8), page 13

$\boldsymbol{c}$ Optimality variable for distributed approach, see equation (2.12), page 15

$\boldsymbol{u}$ The stacked control input of a MAS, see equation (2.3), ... page 10

$\boldsymbol{u}_i$ The control input of an individual agent, see equation (2.1), . . page 9

$\boldsymbol{u}_e$ The external element control input, see equation (4.4), ... page 27

$\boldsymbol{u}_{HIL}$ The human-in-the-loop control input, see equation (4.3), ... page 27

$\boldsymbol{u}_{nom,i}$ The nominal control input of an individual agent, see equation (2.5), page 11

$\boldsymbol{u}_{nom}$ The nominal control input of a MAS, see equation (2.4), ... page 10

$\boldsymbol{x}$ The stacked state of a MAS, see equation (2.3), ... page 10

$\boldsymbol{x}_d$ The stacked target state of a MAS, see equation (2.5), ... page 11

$\boldsymbol{x}_i$ The state of an individual agent, see equation (2.1), ... page 9

$\boldsymbol{x}_{loc,i}$ The locally available state for an individual agent, ... page 10

$\boldsymbol{y}$ Auxiliary decision vector for distributed approach, see equation (2.11), page 15

a The linear parameter for the α function, see equation (4.4), .. page 27

E The edges of a graph, ... page 10

h The control barrier function, see equation (2.6), ... page 12

$I_e(i)$ The indicator function for an agent in the distributed approach, see equation (4.22), ... page 34

k_0 The gain for $\boldsymbol{y}$ in the distributed approach, see equation (2.13), page 15

$L_{\mathfrak{f}}, L_{\mathfrak{g}}$ The Lie derivatives of the system functions, see equation (2.7), page 13

N The number of agents of a MAS, ... page 9

N_i The neighbor set for an individual agent, ... page 10

p The exponential parameter for the distributed approach, see equation (4.12), ... page 33

r The safety distance for collision avoidance and connectivity maintenance, see equation (4.5), ... page 28

v_{xe}, v_{ye} The maximum velocities of the extra robot/human, see equation (4.11), ... page 31

Chapter 1

Introduction

In this thesis work, we will study the combination of Human-In-the-Loop (HIL) control and the system safety for the field of autonomous multi-robot systems (sometimes also known as a Multi-Agent System (MAS)). For this reason, we will first study the existing multi-robot coordination protocols and their applications and then we will identify several safety constraints of high importance for these systems. To enforce the safety constraints, a Control Barrier Function (CBF)-enabled mixer module that takes in the commands from both the multi-robot planner and the human operator will be created. The mixer module will be designed to prioritize the commands from the human operator as long as the safety constraints are not violated. Otherwise, the module will filter the commands and send out a safe alternative to the multi-robot system. The underlying multi-robot tasks are expected to be achieved whenever feasible.

The algorithm will be developed using mathematical theory to prove that the safety conditions will always be fulfilled. Afterwards, simulations using environments with different levels of complexity will be perform to validate the results and lastly, experiments with real robots in a lab environment will be carried out to assess the robustness of the mixer module developed and show its applicability on real-world cases.

1.1 Background

Autonomous multi-robot systems are networks formed by two or more robots (typically a high number) that work together in a coordinated manner to fulfill certain tasks through the use of specific coordination protocols. These can include consensus, formation control, coverage, *etc* [1].

However, these systems are typically pre-programmed and therefore lack a lot of flexibility in real-world applications. Adding HIL control allows not only to drive the MAS in real-time through the use of human commands, but it can also add an extra level of cooperation between the robots and the human that facilitates certain real-world tasks, for example in collaborative manipulation [2].

Another issue of MAS, which has a renewed importance when HIL control is taken into consideration is safety. This can be expressed through constraints that cannot be violated by the system to ensure it is always inside what is considered the safe region of the state space. Collision avoidance between the robots and/or with the human are some of the most important constraints for MAS, but additional constraints like keeping the robots inside the specific lab arena or workspace will also be studied in this thesis. CBFs [3] have recently gained popularity in the field of control and autonomous systems. They are convenient modular tools to ensure the safety of systems and will be used in this work to create a specific algorithm that will guarantee the MAS always remain inside the safe region.

1.2 Problem

As explained before, safety is of special importance for multi-robot systems. Most coordination algorithms can work relatively well by themselves alone, but cannot ensure safety under all conditions. One very representative example is that of collisions between each of the agents of the network and in order to prevent this, additional controllers have to be implemented along side with the basic coordination algorithms. Another example is that of connectivity maintenance, since in a real-world environment most robots have a limited communication range and if one of the robots loses communication with their neighbors the coordination algorithms will stop working.

Adding a human to the problem increases even more the importance of safety, since collisions and random behaviours should always be avoided to guarantee the well-being of the operators. This is why MAS with human-in-the-loop features is a good scenario to implement safety-critical control, meaning that the conditions that are used to define safety are of utmost importance for the controller and should be fulfilled for all time.

1.2.1 Problem definition

The main objective of the project is to design a CBF-enabled mixer module to combine existing multi-robot coordination algorithms with human commands while making sure that the safety constraints are never violated, and validate the theoretical results. The following diagram in Figure 1.1 shows the expected developed module and its parts.

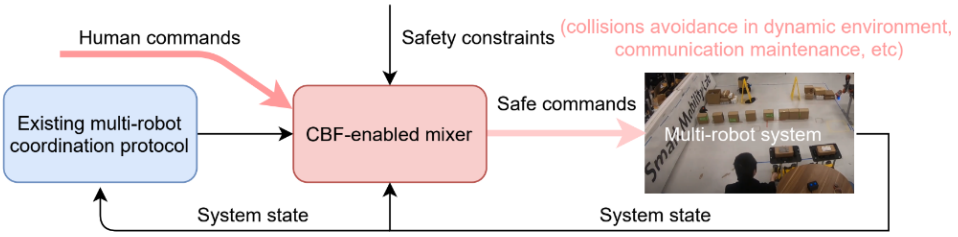


Figure 1.1: CBF-enabled mixer module to be developed.

1.2.2 Scientific and engineering issues

There are several issues that can limit the applicability of this module and that will be studied throughout this thesis. First of all, in order to ensure the safety under all conditions, the system has to be provably-safe from a mathematical point of view. This relies on mathematical theory and there might be several conditions limiting or complicating the proving process. Additionally, since MAS have a high number of variables that scale linearly with the number of robots, the computation time of the algorithms created needs to be taken into consideration so that the system can correctly compute the controller values at the required time spans, specially if the algorithm is to be distributed into several less powerful, on-board computers.

1.3 Purpose

Having a human interact with a robotic system in a safe manner has a lot of applications in different fields. Factory settings and warehouses in particular, where multi-robot systems are very common, have automated most of their systems in the last decades. Moving loads and packages, sorting food, printing and handing labels are some of the tasks that used to be done by people, but are slowly being substituted by robots.

This generates a lot of ethical and social issues, since one can argue that a high number of existing jobs have being lost and will continue to be lost the more automation is implemented, and the factory floors all over the world are slowly becoming less and less human-friendly. By incorporating the human into the MAS directly though HIL control we can solve these problems and take advantage of all the benefits of autonomous systems while doing it in a sustainable way. Taking the human into account from the start will also allow to create safety-critical systems which are better adapted to work with and around people.

Nevertheless, even with all of the recent advantages in intelligent systems, there are still certain tasks that humans are better at, like careful manipulation of objects in industries like the agricultural sector, so having a closer collaboration between human and robots would allow to complete specific tasks in the most efficient manner.

The purpose of this thesis work is to assess and study all of these advantages and work on an algorithm to show what this safe collaboration between human and multi-robot systems can achieve.

1.4 Goals

The goal of this project is to create the CBF-enabled mixer module shown previously. This has been divided into the following four sub-goals:

1. Get familiar with several classic multi-robot coordination algorithms, their respective assumptions and control barrier functions.

2. Perform a literature review on human-in-the-loop control for multi-robot systems.

3. Design the CBF-enabled mixer and analyze the closed-loop system performance through classical control methods and mathematical proofs.

4. Implement the multi-robot coordination algorithms and the designed CBF-enabled mixer module on multiple mobile robots, and test them both with simulations using Robot Operating System (ROS) and experiments at the Smart Mobility Lab (SML) at KTH Royal Institute of Technology (KTH).

1.5 Research Methodology

The main methods to be used throughout the thesis are firstly mathematical proofs, whenever general characteristics of the algorithm can be assessed directly from mathematics and are of special importance, like the concept of forward invariance for safety. Secondly, simulators will be used to test experimentally and safely with MATLAB and Python for point masses, and ROS for more realistic simulations using accurate representations of real-world robots. Finally, experiments on real robots will also be performed since the aim of the thesis is to create an algorithm that is applicable to real-world problems.

1.6 Delimitations

Due to time limitations on the thesis only one type of multi-robot coordination algorithm will be used, although several other ones will be studied to get familiar with all of them and select the more representative. Additionally, only a specific set of safety constraints will be selected, including the most important ones for the MAS with the selected coordination protocol. Lastly, although multi-robot systems can include a lot of different types of vehicles, for this thesis only ground vehicles (mainly for the experiments) and point-mass systems using simple dynamics will be considered.

1.7 Structure of the thesis

Chapter 2 presents relevant background information about multi-robot systems, human-in-the-loop and control barrier functions. Chapter 3 presents the methodology and methods used to solve the problem. Chapter 4 describes and explains both the centralized and distributed modules created in this thesis. Chapter 5 shows and studies the results of the simulations and experiments. Chapter 6 concludes the thesis and expands it with extra information about the future work.

Chapter 2

Background

This chapter describes and provides basic background information and applications, as well as the general formulation, for multi-robot systems and the coordination protocols used to drive them, human-in-the-loop control and control barrier functions.

2.1 Multi-robot systems

Autonomous MAS in comparison with a common autonomous robotic system, usually consist of a large number of agents that must work in a collaborative manner to fulfill a certain assignment or job. This of course has the major drawback of extra added complexity, but it has not stop them from becoming more and more prevalent in several industries, and even though they are still limited to pre-programmed tasks with limited flexibility there is still a lot of fields that can and will benefit from them. As explained before, some of these limitations can be reduced and the multi-robot systems can be made more reliable and worthwhile to use through things like decentralization.

One major example of an application of MAS is that of Unmanned Aerial Vehicle (UAV) swarms, which can be used to create amazing light shows through the use of specific RGB lights incorporated into each of drones in the swarm and by moving them into designated points in the night sky they can create shapes in the air like the one shown in Figure 2.1.

However, due to the high number of UAVs, the chance of collisions between them happening or any other accident increases. This in fact has happened in the past, like during the opening ceremony of the 2018 Winter Olympics where several quadcopters fell from the sky [5] and it is an inherent risk of the MAS.

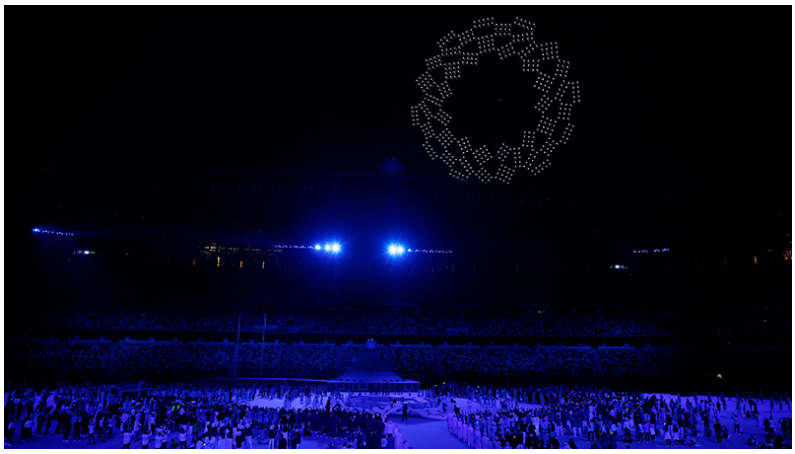


Figure 2.1: Intel light show during Tokyo Olympics [4].

Therefore, safety-critical control for these systems is of utmost importance and it is why there has been several works on safely controlling swarms of drones or UAVs. For example, in [6] where a network of micro flying robots had to sense, navigate and plan a trajectory around obstacles in a forest in real-time while maintaining their formation, which proved to be extremely complicated due to the high number of obstructions in the environment and their relatively small size.

Another characteristic example is that of satellite constellations like the one shown in Figure 2.2, which can be used to provide GPS or even internet all over the world. Again, safety is extremely important since collisions between each satellite will cost a large amount of money due to the the fact they cannot be repaired and new satellites will have to be launched, apart from the fact that the service they provide will be temporarily unavailable. Safety issues also occur when individual spacecraft docking between themselves or other objects [7], for which tools like CBF that will be explained later, are very important.

Lastly, ground robots is another big area that has a lot of applications, be that for moving items inside a warehouse, helping with work in agriculture or even autonomous cars. This specific type of multi-robot system is the one we will focus mostly in this thesis due to the fact that they are the ones we will use for the experiments. More about this will be explained in the next section.

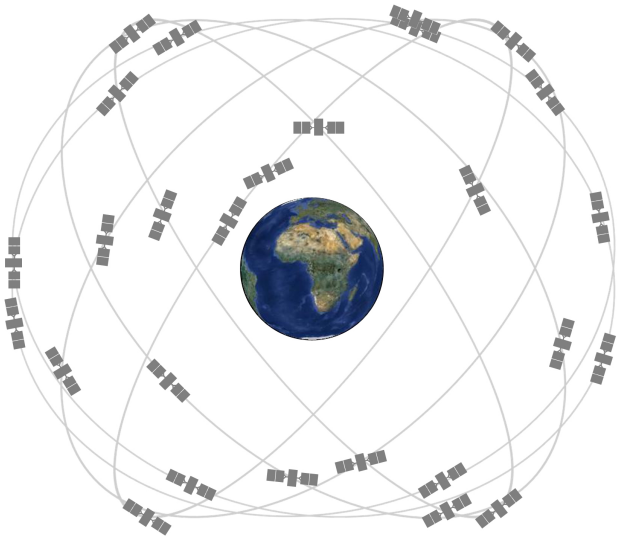


Figure 2.2: GPS satellite constellation [8].

2.1.1 Formulation

Consider a multi-robot system with N agents indexed by $\mathcal{I} = \{1, 2, ..., N\}$. Each agent of the system can be modelled in a generic way with their dynamics as

$$\dot{\boldsymbol{x}}_i = \mathfrak{f}_i(\boldsymbol{x}_i) + \mathfrak{g}_i(\boldsymbol{x}_i)\boldsymbol{u}_i, \tag{2.1}$$

where the state is $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$, and the control input is $\boldsymbol{u}_i \in \mathbb{R}^{m_i}$, $\mathfrak{f}_i(\boldsymbol{x}_i)$, $\mathfrak{g}_i(\boldsymbol{x}_i)$ are locally Lipschitz functions in $\boldsymbol{x}_i$.

Definition 1 (Lipschitz) *A function f such that*

$$|f(x) - f(y)| \leq C|x - y|, \tag{2.2}$$

for all x and y, where C is a constant independent of x and y, is called a Lipschitz function. For example, any function with a bounded first derivative must be Lipschitz.

We can also group the states of all the agents into one variable denoted as $\boldsymbol{x} := (\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, ..., \boldsymbol{x}_N^\top)^\top \in \mathbb{R}^n, n := \sum_{i\in\mathcal{I}} n_i$, the stacked control input would then be $\boldsymbol{u} := (\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top, ..., \boldsymbol{u}_N^\top)^\top \in \mathbb{R}^m, m = \sum_{i\in\mathcal{I}} m_i$, and the stacked vector fields $\mathfrak{f} = (\mathfrak{f}_1^\top, \mathfrak{f}_2^\top, ..., \mathfrak{f}_N^\top)^\top$ and $\mathfrak{g} = \text{blk}(\mathfrak{g}_1, \mathfrak{g}_2, ..., \mathfrak{g}_N)$. Thus, the stacked dynamics can be obtained as

$$\dot{\boldsymbol{x}} = \mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}. \tag{2.3}$$

The communication graph among the MAS is $\mathcal{G} = (\mathcal{I}, E)$, where $\mathcal{I}$ denotes the nodes or agents of the system and E are the edges connecting each of those nodes that share information between themselves. Figure 2.3 shows an arbitrary graph with the nodes and edges denoted. The associated Laplacian matrix [9] representing the connections is denoted as L.

In this report, 1-hop information share will be considered so each agent has access to its own state and the one of its direct neighbours denoted as $N_i := \{j \in \mathcal{I} : (i,j) \in E\}$. The locally obtainable state is $\boldsymbol{x}_{loc,i} := (\boldsymbol{x}_i^\top, \boldsymbol{x}_{j_1}^\top, ..., \boldsymbol{x}_{j_{|N_i|}}^\top)^\top, j_k \in N_i$, for $k \in \{1, 2, ..., |N_i|\}$, i.e., $\boldsymbol{x}_{loc,i}$ stores the states of agent i and all it neighboring agents $j \in N_i$. Throughout the thesis, this system will be originally controlled by a nominal controller assumed to use only the locally available state, which will simplify the latter work done in decentralizing the algorithm.

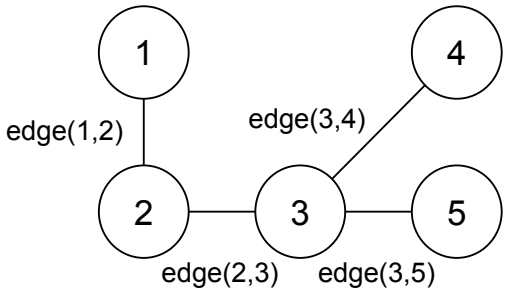


Figure 2.3: An arbitrary graph showing the nodes and edges.

There are several conventional multi-agent coordination algorithms that can be used as a nominal controller, *e.g.* consensus (where all the agents seek to reach the same state variables), coverage (where all agents spread as far as possible, while satisfying some constraints, to cover as much as area as they can), leader-follower (where all the agents only communicate with one leader that distributes the tasks), formation control (where the agents seek to form a specific formation shape), *etc*. In particular, the formation controller would be used since it is one of the most commonly used for MASs and has a lot of applications as previously mentioned. It can be written in a stacked form as

$$\dot{\boldsymbol{x}} = \boldsymbol{u}_{nom} = -L(\boldsymbol{x} - \boldsymbol{x}_d), \tag{2.4}$$

and can also be written as the following

$$\boldsymbol{u}_{nom,i} = -\sum_{j \in N_i} (\boldsymbol{x}_i - \boldsymbol{x}_j - (\boldsymbol{x}_{d,i} - \boldsymbol{x}_{d,j})), \text{ for } i \in \mathcal{I}, \tag{2.5}$$

where $\boldsymbol{x}_d$ is the stacked target state vector for the desired formation shape.

2.2 Human-in-the-loop control

Another major background area for this thesis is the human-in-the-loop aspect. Having a human interact with autonomous systems not only allows to predict and avoid risks, but it also adds extra flexibility and collaboration as explained before.

HIL control is actually already in a lot of sectors, pilots in airplanes for example use the autopilot for most time of the travel. But this does not mean that pilots do nothing, on the contrary, the pilot is expected to monitor the system and modify its variables, like the flying altitude and speed, whenever needed. Drivers in autonomous cars nowadays are another example, since the autopilot systems are not advanced enough and need constant supervision. Another important application for autonomous systems where HIL has to be taken into account is in search and rescue missions [10], where a mixed team of robots and humans can more efficiently and rapidly locate victims, rather than each of them individually on their own.

However, most of these systems do not consider the human and the controller from a holistic point of view, but instead as separate systems. By contrast, if the human commands are taken into account, a more overall safe and efficient system can be created [11].

This is specially important for multi agent systems where the risks are high due to the number of agents, and general controllers that take the human into account from the start are needed, for example in the following reference where a swarm of robots is controlled through human commands that are pre-analyzed inside of the environment before execution to prevent collisions in case those commands are not safe and compute a feasible alternative [12].

Other interesting examples include the use of HIL control for assiting humans on complex tasks for systems that are typically very difficult to control. By filtering these human commands in real time and correcting ambiguous or incorrect inputs, these systems can be adequately managed [13, 14]. This is specially interesting for path planning of robotic arms [15], where a human can command a trajectory and the robot analyzes it to ensure it is achievable.

2.3 Control Barrier Functions

The last major aspect needed for this thesis and the one that will see the most work is that of Control Barrier Functions. They are extremely useful tools for safety and although they have existed for some time already, they are experiencing a lot of interest in recent years.

In fact, there has already being some work on CBFs to ensure safety on systems that receive human commands, like in [16] where the signals sent to a UAV are modified to avoid obstacles, and even for MASs where the nominal algorithm is modified through CBFs to avoid inter-collisions [17, 18, 19]. There has also been work on using formal methods like Signal Temporal Logic (STL) to describe tasks for these multi-robot systems and the control strategy is performed through the use of CBFs [20].

Lastly, something that has been used for distributed approach developed in this thesis is the work done on nonsmooth CBFs for MAS [21], where the max and min operators, which yields nonsmooth functions, are used to create Boolean compositions of several barrier functions into one.

2.3.1 Formulation

In order to properly explain how CBFs work, first the concept of a safety set $\mathscr{C}$, shown in Figure 2.4 needs to be introduced. This represents a specific constraint of our design defined by a differentiable function $h(\boldsymbol{x})$ that has to always be non-negative in order to consider the problem to be safe. This is represented as

$$\mathscr{C} = \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}. \tag{2.6}$$

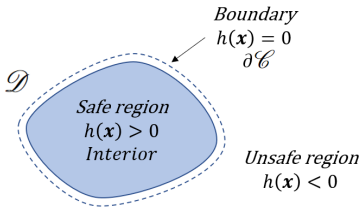


Figure 2.4: Geometric representation of the safe region $\mathscr{C}$.

Definition 2 (Extended class $\mathcal{K}$ function) *A continuous function* $\alpha : [0, a) \longrightarrow [0, \infty)$ *that is strictly increasing with* $\alpha(0) = 0$.

Definition 3 (CBF) *Let set $\mathscr{C}$ be defined by (2.6). $h(\boldsymbol{x})$ is a control barrier function (CBF) for the stacked system (2.3) if there exists a locally Lipschitz extended class $\mathcal{K}$ function α such that:*

$$\sup_{\boldsymbol{u}\in\mathbb{R}^m} [L_{\mathfrak{f}}h(\boldsymbol{x}) + L_{\mathfrak{g}}h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x}))] \geq 0,\ \forall \boldsymbol{x} \in \mathscr{D} \subset \mathscr{C}, \tag{2.7}$$

where $L_{\mathfrak{f}}h = \nabla h^\top \mathfrak{f}(\boldsymbol{x}) \in \mathbb{R}$ and $L_{\mathfrak{g}}h = \nabla h^\top \mathfrak{g}(\boldsymbol{x}) \in \mathbb{R}^{1\times m}$ are the Lie derivatives for the system and the operator $\nabla : C^1(\mathbb{R}^n) \to \mathbb{R}^n$ is defined as the gradient $\frac{\partial}{\partial x}$ of a scalar-valued differentiable function with respect to $\boldsymbol{x}$.

Additionally, it has been shown [22, 23] that any locally Lipschitz control input $\boldsymbol{u}$ that satisfies the CBF constraint (2.7) renders the set $\mathscr{C}$ forward invariant and, if $\mathscr{C}$ is compact, it is also asymptotically stable. This means that if the system ends up inside the safety set $\mathscr{C}$, it will never leave it, and if the initial condition starts outside, the trajectory will asymptotically converge to the inside of the set, respectively.

In general, the CBF condition shown in (2.7) can be represented in a simpler linear form

$$\sum_{i\in\mathcal{I}} \boldsymbol{a}_i^\top(\boldsymbol{x}_i)\boldsymbol{u}_i + \sum_{i\in\mathcal{I}} b_i(\boldsymbol{x}_i) \leq 0. \tag{2.8}$$

There are two types of constraints that satisfy the aforementioned assumption and that will be considered in this work:

1. Individual constraints $h_i(\boldsymbol{x}_i)$ that only depend on the state of one agent i and can be directly decentralized.

2. Collective constraints $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ which depend on the state of more than one agent and will be more complex to decentralize.

For the derivation of the main results, the collective constraints form will be used, but simulation and experiment results for both of them will be shown since the individual constraints form is simpler to derive.

Using the previous result, a general controller that would always satisfy the constraints can be created by solving the following Quadratic Program (QP)

$$\begin{aligned} &\min_{\boldsymbol{u}} \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|^2 \\ s.t.\ &\sum_{i\in\mathcal{I}} \boldsymbol{a}_i^\top(\boldsymbol{x}_i)\boldsymbol{u}_i + \sum_{i\in\mathcal{I}} b_i(\boldsymbol{x}_i) \leq 0. \\ &\text{for each of the constraints} \end{aligned} \tag{2.9}$$

where multiple constraints have been considered. If there is only one constraint, the QP controller can be simplified to

$$\begin{aligned} \boldsymbol{u}_i &= \boldsymbol{u}_{nom,i} - \mu \boldsymbol{a}_i \\ \text{where } \mu &= \max(0, (\boldsymbol{a}^\top \boldsymbol{u}_{nom} + \boldsymbol{b})/\|\boldsymbol{a}\|^2). \end{aligned} \tag{2.10}$$

2.3.2 Distributed implementation

However, the controller previously shown is based on a centralized approach and therefore needs more than just the locally available state. Decentralized approaches are more useful since each agent gets to compute its own controller based on the information that it has available and therefore it eliminates the need for a powerful computer.

This is due to the fact that if the network of agents is extremely big, each decentralized problem would become much easier to solve. Additionally, it gives more robustness to the problem since agents that are not neighbors would not be affected between each other if one stops working.

Going back to the controller shown in (2.7), in order to implement it in a distributed fashion, there exist different methods, for example dual decomposition [24] and ADMM [25], but the one proposed in [26] would be used due to continuity purposes and the advantages that it offers compared to the previous two methods as it can be seen in the following Table 2.1.

Table 2.1: Different distributed approaches for CBF.

Methods	Graph topology	Local info exchange	Satisfy constraint
Dual decomposition	Time-varying directed graph and $(\mathcal{I}, E_\infty)$ strongly connected	1-hop dual variable	Not always
ADMM	Undirected graph and agents sharing the same coupling constraints are neighbors	1-hop primal and dual variables	Not always
Distributed method for MAS	Connected and undirected	1-hop states and an auxiliary scalar	Always

In this method, instead of solving one centralized QP, the following equivalent QP is to be solved by each individual agent $i \in \mathcal{I}$ and when all of these individual QPs are summed up together it should be the same as computing the previous centralized QP

$$\begin{aligned} &\min_{\boldsymbol{u}_i \in \mathbb{R}^{m_i}} \|\boldsymbol{u}_i - \boldsymbol{u}_{nom,i}\|^2 \\ &s.t.\ \boldsymbol{a}_i^\top(\boldsymbol{x}_{loc,i})\boldsymbol{u}_i + \sum_{j \in N_i}(y_i - y_j) + b_i(\boldsymbol{x}_{loc,i}) \leq 0, \end{aligned} \tag{2.11}$$

where the CBF condition only uses the locally available information $\boldsymbol{x}_{loc,i}$ and $\boldsymbol{y} = (y_1, y_2, ..., y_N) \in \mathbb{R}^N$ is an auxiliary decision variable. If all the local QPs are feasible, the CBF condition (2.7) is fulfilled for any $\boldsymbol{y}$. The problem now is to find the optimal $\boldsymbol{y}$ that gives the optimal solution to the QP. For that, the following local variable is used

$$c_i = \frac{1}{\boldsymbol{a}_i^\top \boldsymbol{a}_i}(\boldsymbol{l}_i \boldsymbol{y} + \boldsymbol{a}_i^\top \boldsymbol{u}_{nom,i} + b_i), \tag{2.12}$$

and if $\boldsymbol{y}$ is chosen such that $c_i = c_j$ for any $i, j \in \mathcal{I}$, then $\boldsymbol{y}$ would be optimal. Particularly, if the following discontinuous adaptative law

$$\dot{\boldsymbol{y}} = -k_0 \text{sign}(L\boldsymbol{c}), \tag{2.13}$$

is used with the assumption that $\boldsymbol{a}_i, \boldsymbol{u}_{nom,i}$ and $\boldsymbol{b}_i$ are slowly time-varying and the gain

$$k_0 \geq a_{max}(2\delta_{max} D / a_{min} + \epsilon), \tag{2.14}$$

Then $\boldsymbol{c}$ reaches consensus within a finite time and therefore the optimal solution has been found. Additionally, instead of solving the local QP from (2.11), the following analytical solution can be used

$$\boldsymbol{u}_i = \boldsymbol{u}_{nom,i} - \max(0, c_i)\boldsymbol{a}_i, \ \forall i \in \mathcal{I}. \tag{2.15}$$

The following two types of constraints are identified in [26] that satisfied the assumption of using only locally available information

1. $h(\boldsymbol{x}) = \sum_{i \in \mathcal{I}} h_i(\boldsymbol{x}_i)$ for individual constraints.
2. $h(\boldsymbol{x}) = \sum_{e \in E} h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for collective constraints.

2.4 Summary

These three main background areas will be joint together to create the CBF-enabled mixer module that will satisfy the intended missions described in later chapters. The research previously done at KTH in the Division of Decision and Control Systems (DCS) for the decentralization of the CBF algorithm for multi-robot systems will be of particular importance and significant improvements are to be developed for it.

Chapter 3

Methods

The purpose of this chapter is to provide an overview of the research methods used in this thesis. Section 3.1 describes the research process. Section 3.2 focuses on the data collection techniques and experimental design used for this research. Section 3.3 explains the reliability and validity of the methods used, as well as an analysis of the data collected.

3.1 Research Process

The process that will be followed in this thesis is based on four main items. Starting with the centralized module, the following steps will be taken

Step 1 literature study and/or mathematical development & validation of algorithm,

Step 2 test algorithm in the simple environment simulators (Python & MATLAB),

Step 3 test algorithm in a more realistic simulator (ROS), and

Step 4 validate the algorithm through tests with real robots.

After the centralized module is finished, the same steps will be taken to create and validate the distributed implementation. This can also be seen in the diagram shown in Figure 3.1.

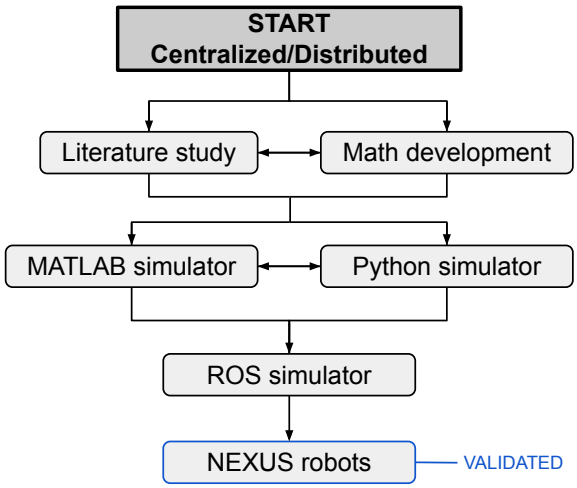


Figure 3.1: Diagram showing the research process.

3.2 Data Collection and Experimental design

The last three steps of the research process will be studied in this section as they are the ones that will provide experimental data and therefore their configurations must be explained.

3.2.1 MATLAB and Python simulator

These are fundamentally two different simulators, but due to the similarity of the code structure and that they will use the same model they are incorporated together into the same subsection.

MATLAB is a programming platform that uses its own matrix-based language that works really well for computational mathematics. Because of this, it would be used as the starting step to test the algorithms developed, particularly for the distributed implementation. With it, simple systems dynamics will be implemented, based on point-mass robots or agents such that $\dot{\boldsymbol{x}} = \boldsymbol{u}$. The simulator computes the nominal and final controllers based on system parameters. Figure 3.2 shows how the output of the simulator looks like for an arbitrary case. The asterisks represent the starting point of the agents and the circle the final point. The dashed lines represent the communication topology of the network at the final time instant.

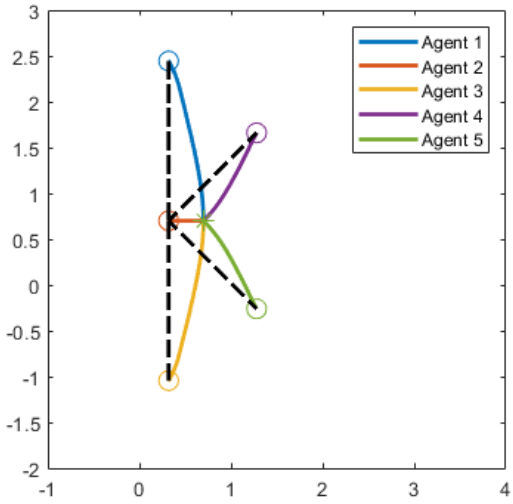


Figure 3.2: MATLAB simulator output.

Additionally, this simulator also creates several plots for both the general CBF function and each individual one, as well as the CBF condition and a comparison with the nominal and centralized controllers.

Python is an interpreted, object-oriented, high-level programming language that is adapted for several different applications and fields thanks to its wide range of modules and support for different systems and platforms. For this thesis work, the Python simulator created will use the same system dynamics and model as the MATLAB one. The main reason for its creation is the high speed of computation compared with MATLAB which allows to compute cases with a higher number of robots, and the possibility to get videos of the evolution of the MAS. Lastly, for the centralized cases this simulator works much better since the Python modules for computing QP problems (mainly SciPy) are much faster than those for MATLAB. The following Figure 3.3 shows how the Python simulator videos look like, with the blue dots representing the agents and the lines representing the edges of the graph. The red rectangle represents the size of the arena, which will be used for some cases in the centralized implementation.

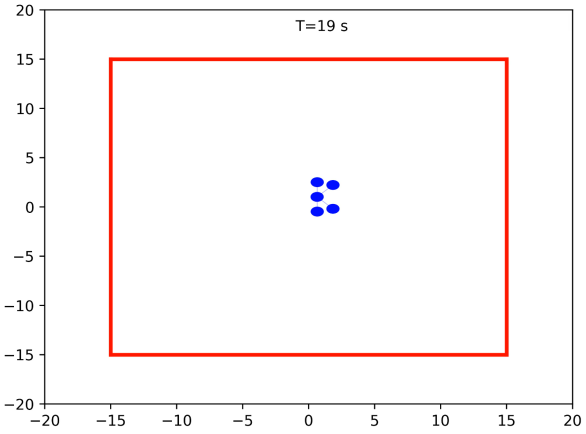


Figure 3.3: Python simulator output.

3.2.2 ROS

The Robot Operating System (ROS) is a set of software libraries and tools with the aim of helping people build robot applications [27]. It also provides a communication platform to share information between different agents in the network through topics and publishers/subscribers, as well as services. In addition, it has an ample selection of high fidelity simulators. For this thesis, we will use Gazebo [28] with realistic models of the NEXUS robots to be used for the real-world experiments. The system dynamics used would be the same as before $\dot{\boldsymbol{x}} = \boldsymbol{u}$, but the output would be handled in a different manner through a separate controller associated with each robot.

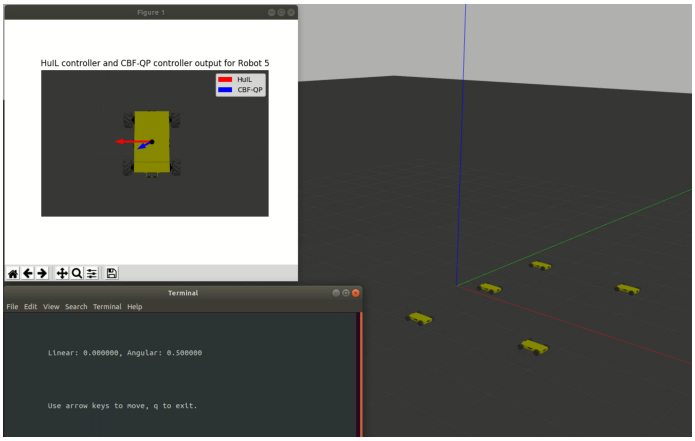


Figure 3.4: ROS simulator output.

Figure 3.4 shows how the Gazebo simulator looks like with ROS, where five yellow NEXUS robots have been spawned and for the one that is controlled through HIL a specific window displays the human and final input. There is also a small command panel where the human can send linear and angular commands to the HIL robot.

3.2.3 NEXUS 4WD robots

After the algorithm has been extensively tested in both simple and more complex simulators, real-world experiments can be performed. For this, the NEXUS 4WD Mecanum Wheel Mobile Arduino Robotics car [29], shown in Figure 3.5, will be used, in particular five of them to form a network of agents. Each robot counts with four mecanum wheels driven each one by a different motor with encoders which gives it flexibility to move in every direction possible, an aluminum alloy body with suspension to ensure that the four wheels adhere to ground and ultrasonic sensors to measure distance to obstacles. All of this can be accessed through an Arduino micro-controller.

Figure 3.5: NEXUS 4WD Mecanum Wheel Mobile Arduino Robotics car.

In order to utilize ROS with the NEXUS robots, they have been fitted with NVIDIA Jetson Nano or Intel NUC on-board computers running an Ubuntu Distro. This would allow us to utilize the code developed for the ROS simulator directly with very minor modifications.

For localization, the Motion Capture System from Qualisys [30] that is already installed in the SML will be used. This allow us to obtain the position of each NEXUS in real-time by placing several markers on each of the robots and letting the cameras of the system record their positions.

3.3 Assessing reliability and validity of the data collected

The most important question to ask now is how reliable are the steps shown in the research process to develop the required capabilities for the stated goals in Chapter 1. For this, we have to look first at the methods and then at the data collected from them. Table 3.1 shows the main concerns affecting the reliability and validity of the data and the methods used.

Table 3.1: Main concerns for the reliability and validity of data and methods.

	Multi-robot system	**HIL**	**CBF**
Methods	Using a system that is generic enough	How representative are the HIL elements chosen?	Are the safety constraints representative?
Data	Make sure that the data is correctly distributed	Is the human input being sent/processed correctly?	Are the constraints always satisfied for the REAL data?

3.3.1 Validity and reliability of method

On of the main concerns of the methods proposed is the fact that the module might only work for a specific set of multi-robot systems, particularly those that have been used to developed the module, like the NEXUS robots and the point-mass system. However, since all the mathematics have been developed using the generic system dynamics shown in (2.3) and not the simplified ones, the proposed algorithms should work for any system that can be model with the general equations.

Additionally, ROS can easily work with more complicated systems like quadcopters, so even though only ground robots have been considered, the methodology applied should work satisfactory for any other type of system that works with ROS.

3.3.2 Data validity and reliability

The other main concern is that the data obtained from these methods is not valid, specially for the real-world robots since it might happen that the data collected is either not enough or corrupted. However, since we are recording the same data that we are using to compute the controller values from the algorithm, it can be determined that as long as the algorithm works, the data should be reliable enough to draw conclusions from it.

The last thing to consider is which data will be created with these methods and how will it be analyzed. Taking into account that safety is the most important characteristic of the system, the CBF functions and its secondary variables will be the main focus point to study. Therefore, not only will they be recorded and plotted, but the data required to compute them as well.

The second main thing to study is the nominal task of the controller, which in this case is the formation controller and for this the time evolution plots of the system will be extremely useful to determine if the desired shape is not only achieved for the final instant, but also for individual time-stamps during the development of the system.

The HIL inputs will also be analyzed to make sure that the robots are receiving the correct information and acting on it. For this, plotting the normed difference between the nominal and final controller will easily show if the human inputs are being sent correctly.

Chapter 4

CBF-enabled mixer module

In this chapter, the module developed during the thesis will be explained. Both the centralized and the distributed implementation will be presented in the following sections.

4.1 Centralized module

Starting from the centralized QP controller presented in (2.9), the controller for all the agents is calculated as one stacked problem. Note that every edge is represented by the tuple (i, j), where the communication is bidirectional $(i, j) = (j, i)$, with the following convention $i \leq j$.

The general formula for the controller using the vectorized form is the following:

$$\begin{aligned} &\min_{\boldsymbol{u}} \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|^2 \\ &s.t.\ \boldsymbol{A}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{b}(\boldsymbol{x}) \geq 0. \\ &\text{for each of the constraints} \end{aligned} \tag{4.1}$$

where the nominal controller is the formation controller previously shown and the constraints represent the safety parameters. The system dynamics used are the stacked ones from (2.3) and the linear constraints are created using the CBF condition from (2.7).

The Laplacian matrix is needed for the nominal formation controller. For all of the experiments and simulations five robots will be used forming a K-letter shape like the one shown in Figure 4.1. The Laplacian matrix for this example is

$$L = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} \text{Created using the following} \\ \text{if } i = j \Rightarrow L_{i,j} = |N_i| \\ \text{if } i \neq j \text{ and } j \in N_i \Rightarrow L_{i,j} = -1 \\ \text{else } \Rightarrow L_{i,j} = 0 \end{cases} \tag{4.2}$$

where i represents the rows and j the columns. However, since our problem is in 2D, it is needed to extend this Laplacian for two components. For that, the Kronecker product is used such that $L = L \otimes I_{2\times 2}$. If it was in 3D, this would become $L = L \otimes I_{3\times 3}$ and similar for higher dimensions.

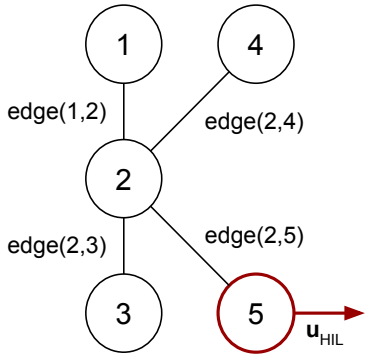


Figure 4.1: Network of five robots to be used for the experiments.

It is important to mention that this method works for any shape and number of dimensions and the rest of the mathematics are developed in a generic way without substituting L, this is just an example to show how it would work for a specific configuration.

4.1.1 Adding HIL

There are several ways to add Human-In-the-Loop features to the controller, but in this paper only two cases will be considered. The first one is the most simple one, where one of the agents is controlled to some extent by a human operator. However, since this agent is still part of the network its nominal controller will be defined as a combination of the main nominal controller used (like a formation controller) and the HIL input

$$\boldsymbol{u}_{nom,i}^{HIL} = \boldsymbol{u}_{HIL} + \boldsymbol{u}_{nom,i}, \tag{4.3}$$

which will then be passed through the QP to obtain the sactual control input.

In the second case, the HIL element will be a human in the presence of the workspace. For example, a person could enter into the experiment's arena and interact with the agents in some way, like herding/chasing the robots. In this case, the human behaviour can be considered as an external agent fully controlled by the human

$$\boldsymbol{u}_e = \boldsymbol{u}_{HIL}. \tag{4.4}$$

4.1.2 Safety constraints

The safety constraints to be considered are the most representative for MASs. The following Figure 4.2 shows the main constraints of arena size, wedge size and inter-collision avoidance and connectivity maintenance.

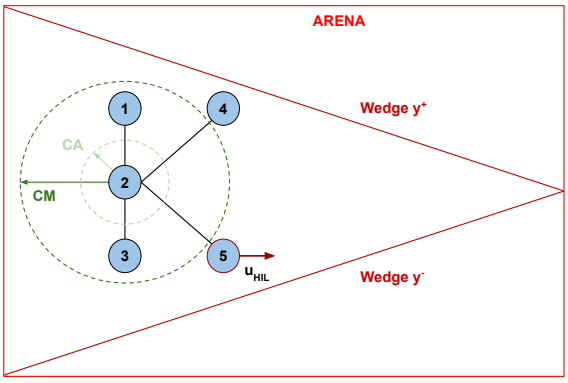


Figure 4.2: Virtual arena showing the safety constraints.

To show how the constraints work, examples will be shown with either 3 or 5 robots in 2D with simple dynamics $\dot{\boldsymbol{x}} = \boldsymbol{u}$ and with $\alpha(h(\boldsymbol{x})) = a \cdot h(\boldsymbol{x})$.

4.1.2.1 Inter-collision avoidance and connectivity maintenance

The edges of the formation determine which robots are neighbours and each edge will create one constraint for the collective constraints, like Collision Avoidance (CA) and Connectivity Maintenance (CM). Taken into account that this is the centralized version, this constraint will be divided in two gradients/matrix elements of the vectorized constraint, since there are two agents per edge and one control input per agent.

Assuming only connectivity maintenance is used the CBF function becomes

$$h(\boldsymbol{x}_i, \boldsymbol{x}_j) = r^2 - ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \geq 0, \tag{4.5}$$

where $\forall(i,j) \in E$ and $||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = (x_i - x_j)^2 + (y_i - y_j)^2$ in the 2D case for ground vehicles where $\boldsymbol{x}_i = (x_i, y_i)$.

For vector $\boldsymbol{b}$, taking into account that $h(\boldsymbol{x}_i, \boldsymbol{x}_j) = h(\boldsymbol{x}_j, \boldsymbol{x}_i)$. Considering the five agent network shown in Figure 4.1 as an example this would become

$$\boldsymbol{b} = \alpha(h) = a \cdot h = a \cdot \begin{bmatrix} h(\boldsymbol{x}_1, \boldsymbol{x}_2) \\ h(\boldsymbol{x}_2, \boldsymbol{x}_3) \\ h(\boldsymbol{x}_2, \boldsymbol{x}_4) \\ h(\boldsymbol{x}_2, \boldsymbol{x}_5) \end{bmatrix} \tag{4.6}$$

For matrix $\boldsymbol{A}$, the partial derivatives of the CBF function h are needed. Assuming again the 2D case for the ground vehicles with the simplified dynamics they become

$$\begin{array}{l} \partial h/\partial x_i = -2(x_i - x_j) \\ \partial h/\partial y_i = -2(y_i - y_j) \\ \partial h/\partial x_j = 2(x_i - x_j) \\ \partial h/\partial y_j = 2(y_i - y_j) \end{array} \Rightarrow \partial h/\partial x_i = -\partial h/\partial x_j \text{ and } \partial h/\partial y_i = -\partial h/\partial y_j \tag{4.7}$$

for edge (i,j), the safety constraint is calculated as

$$\left.\begin{array}{l} h(\boldsymbol{x}_i, \boldsymbol{x}_j) = r^2 - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2 \\ \partial h/\partial x_i = -2(x_i - x_j) \\ \partial h/\partial y_i = -2(y_i - y_j) \\ \partial h/\partial x_j = 2(x_j - x_i) \\ \partial h/\partial y_j = 2(y_j - y_i) \end{array}\right\} \text{CBF constraint: } \nabla h^\top \boldsymbol{u} + \alpha(h) \geq 0$$

$$[\partial h/\partial x_i \quad \partial h/\partial y_i \quad \partial h/\partial x_j \quad \partial h/\partial y_j] \begin{bmatrix} \boldsymbol{u}_i \\ \boldsymbol{u}_j \end{bmatrix} + \alpha(h(\boldsymbol{x}_i, \boldsymbol{x}_j)) \geq 0$$

$$-2 \begin{bmatrix} x_i - x_j & y_i - y_j & x_j - x_i & y_j - y_i \end{bmatrix} \begin{bmatrix} u_{ix} \\ u_{iy} \\ u_{jx} \\ u_{jy} \end{bmatrix} + \alpha(h(\boldsymbol{x}_i, \boldsymbol{x}_j)) \geq 0$$

extrapolating this for all the edges, the final matrix $\boldsymbol{A}$ is obtained as

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial h_{12}}{\partial x_1} & \frac{\partial h_{12}}{\partial y_1} & -\frac{\partial h_{12}}{\partial x_1} & -\frac{\partial h_{12}}{\partial y_1} & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & \frac{\partial h_{23}}{\partial x_2} & \frac{\partial h_{23}}{\partial y_2} & -\frac{\partial h_{23}}{\partial x_2} & -\frac{\partial h_{23}}{\partial y_2} & 0 & 0 & \dots \\ 0 & 0 & \frac{\partial h_{24}}{\partial x_2} & \frac{\partial h_{24}}{\partial y_2} & 0 & 0 & -\frac{\partial h_{24}}{\partial x_2} & -\frac{\partial h_{24}}{\partial y_2} & \dots \\ 0 & 0 & \frac{\partial h_{25}}{\partial x_2} & \frac{\partial h_{25}}{\partial y_2} & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \tag{4.8}$$

where $h_{ij} = h(\boldsymbol{x}_i, \boldsymbol{x}_j)$. All of this has been coded in a modular way such that the number of robots, formation shape and neighbour configuration can be modify at will and the controller variables and QP problem will be updated accordingly. As explained before, this is the particular example used for the experiments and simulations, but the algorithm can work for any generic number of dimensions and different network configurations.

4.1.2.2 Arena safety constraint

The current size limits of the arena at the SML are: $x_{max} = 2$ m, $x_{min} = -2$ m, $y_{max} = 2$ m and $y_{min} = -3$ m. This means that for each robot there will be 4 safety constraints for the 4 rectangular sides represented as

$$h_i^{x_{\text{max}}} = x_{\text{max}} - x_i$$
$$h_i^{x_{\text{min}}} = x_i - x_{\text{min}}$$

$$h_i^{y_{\text{max}}} = y_{\text{max}} - y_i$$
$$h_i^{y_{\text{max}}} = y_i - y_{\text{min}}$$

For each agent, the CBF constraint is $\nabla h_i^T \boldsymbol{u}_i + \alpha(h_i) \geq 0$, where

$$\nabla h_i = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \text{ and } \boldsymbol{b}_i = \alpha(\boldsymbol{h}_i) = a \begin{bmatrix} x_{\text{max}} - x_i \\ x_i - x_{\text{min}} \\ y_{\text{max}} - y_i \\ y_i - y_{\text{min}} \end{bmatrix}.$$

Assuming 3 agents, the constraints can be joined together in the form of $\boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} \geq 0$, where the $\boldsymbol{A}$ matrix and $\boldsymbol{u}$ controller are shown below and the vector $\boldsymbol{b}$ is just an extension of the one shown for individual agents and therefore it is skipped.

$$\boldsymbol{u} = \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \end{bmatrix} \Rightarrow \boldsymbol{A} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

This matrix can be constructed using $\boldsymbol{A} = I_{3x3} \otimes \nabla h_i^T$.

4.1.2.3 Wedge shape safety constraint

This constraint is similar to the one used for the arena and it is also shown in Figure 4.2, but only two constraints are needed to make sure that the robot does not go outside of the following wedge shape (defined using the available space at the SML). The CBF functions for each agent are

$$\begin{aligned} h_i^{\max} &= y^+ - y_i = -\frac{y_{\max}}{x_{\max} + x_{\min}} x_i + y_{\max}/2 - y_i \\ h_i^{\min} &= y_i - y^- = y_i - \frac{y_{\max}}{x_{\max} + x_{\min}} x_i + y_{\max}/2 \end{aligned} \tag{4.9}$$

so that $\nabla h_i = \begin{bmatrix} -\frac{y_{\max}}{x_{\max}+x_{\min}} & -\frac{y_{\max}}{x_{\max}+x_{\min}} \\ -1 & 1 \end{bmatrix}$ and $\boldsymbol{b}_i = a \begin{bmatrix} y^+ - y_i \\ y_i - y^- \end{bmatrix}$.

For the centralized problem assuming 3 agents this becomes

$$\boldsymbol{A} = \begin{bmatrix} -\frac{y_{\max}}{x_{\max}+x_{\min}} & -1 & 0 & 0 & 0 & 0 \\ -\frac{y_{\max}}{x_{\max}+x_{\min}} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{y_{\max}}{x_{\max}+x_{\min}} & -1 & 0 & 0 \\ 0 & 0 & -\frac{y_{\max}}{x_{\max}+x_{\min}} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{y_{\max}}{x_{\max}+x_{\min}} & -1 \\ 0 & 0 & 0 & 0 & -\frac{y_{\max}}{x_{\max}+x_{\min}} & 1 \end{bmatrix} \tag{4.10}$$

This matrix can again be constructed using $\boldsymbol{A} = I_{3x3} \otimes \nabla h_i^T$. With $\boldsymbol{b}$ being an extension for one agent and $\boldsymbol{u}$ being the same as for the arena case.

4.1.2.4 Extra Human-In-The-Loop agent

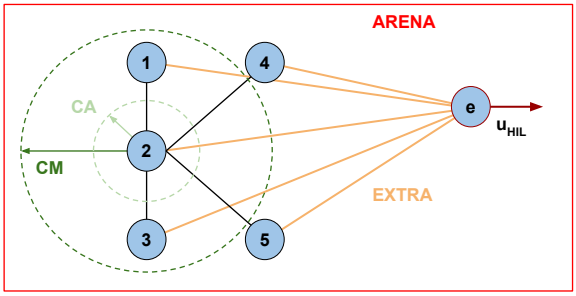


Figure 4.3: Virtual arena showing the safety constraints for the extra robot.

The last constraint to study is collision avoidance with an extra robot that is fully controlled by a human (or maybe a human itself) and it is not taken into account in the network. Each of the robots in the formation would have to make sure they have enough distance with this extra robot/human to avoid collisions, like shown in Figure 4.3. For one agent this becomes

$$\begin{array}{ll} \text{For the agent } i\text{:} & \text{For extra-HIL:} \\ \boldsymbol{u}_i = \begin{bmatrix} u_{ix} \\ u_{iy} \end{bmatrix} & \boldsymbol{u}_e = \begin{bmatrix} u_{ex} \\ u_{ey} \end{bmatrix} \end{array} \tag{4.11}$$

$$\begin{aligned} & h\left(\boldsymbol{x}_i, \boldsymbol{x}_e\right) = \left\|\boldsymbol{x}_i - \boldsymbol{x}_e\right\|^2 - r^2 \\ & \text{where } \left\|\boldsymbol{x}_i - \boldsymbol{x}_e\right\|^2 = \left(x_i - x_e\right)^2 + \left(y_i - y_e\right)^2 \\ & \nabla h\left(\boldsymbol{x}_i, \boldsymbol{x}_e\right) = \begin{bmatrix} \partial h/\partial x_i \\ \partial h/\partial y_i \\ \partial h/\partial x_e \\ \partial h/\partial y_e \end{bmatrix} = \begin{bmatrix} 2\left(x_i - x_e\right) \\ 2\left(y_i - y_e\right) \\ -2\left(y_i - y_e\right) \\ -2\left(y_i - y_e\right) \end{bmatrix} \\ & \boldsymbol{A}_i = 2\begin{bmatrix} x_i - x_e & y_i - y_e \end{bmatrix} \\ & b_i = \partial h/\partial x_e v_{xe} + \partial h/\partial y_e v_{ye} + \alpha(h) = \\ & \partial h/\partial x_e v_{xe} + \partial h/\partial y_e v_{ye} + a \cdot \left(\left\|\boldsymbol{x}_i - \boldsymbol{x}_e\right\|^2 - r^2\right) \end{aligned}$$

where v_{xe} and v_{ye} are the absolute maximum velocities of the extra robot/human in the x and y axis respectively. This way safety is ensured for all possible speeds while being more conservative for speeds lower than the maximum one.

For the centralized problem assuming 3 agents the constraints can be expressed as $\boldsymbol{A}\boldsymbol{u} + \boldsymbol{b} \geq 0$, where

$$\boldsymbol{u} = \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \end{bmatrix}, \quad \boldsymbol{b} = a \begin{bmatrix} \|\boldsymbol{x}_1 - \boldsymbol{x}_e\|^2 - r^2 - 2(x_1 - x_e)v_{xe} - 2(y_1 - y_e)v_{ye} \\ \|\boldsymbol{x}_2 - \boldsymbol{x}_e\|^2 - r^2 - 2(x_1 - x_e)v_{xe} - 2(y_1 - y_e)v_{ye} \\ \|\boldsymbol{x}_3 - \boldsymbol{x}_e\|^2 - r^2 - 2(x_1 - x_e)v_{xe} - 2(y_1 - y_e)v_{ye} \end{bmatrix}$$

$$\boldsymbol{A} = \begin{bmatrix} 2\left(x_1 - x_e\right) & 2\left(y_1 - y_e\right) & 0 & 0 & \dots \\ 0 & 0 & 2\left(x_2 - x_e\right) & 2\left(y_2 - y_e\right) & \dots \\ 0 & 0 & 0 & 0 & \dots \end{bmatrix}$$

4.2 Distributed module

Starting from the distributed method shown in (2.11), this only works when one constraint is taken into account. In order to be able to satisfy more than one constraint, the solution needs to be adapted. One of the possible ways to extend the solution to more constraints is through the use of the minimum function.

4.2.1 Minimum function approximation

In order to be inside the safety set shown in (2.6), all the CBF functions have to be non-negative. Therefore, to group all of these conditions together into one equation, the minimum function can be used such that the minimum of all of the CBF functions has to be non-negative, that is $\min_{e \in E} h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$. However, this does not follow the constraints format from the literature [26] where the general CBF function is expressed as the summation of several constraints. This is solved by using the following minimum approximation

$$\min_{e \in E} h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq -\frac{1}{p} \log \left(\sum_{e \in E} e^{-p h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right) \geq 0, \tag{4.12}$$

for $(i, j) \in e$, and adapting it to the required format by moving the inequality sign

$$-\frac{1}{p}\log\left(\sum_{e\in E} e^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j)}\right) \geq 0 \Rightarrow \log\left(\sum_{e\in E} e^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j)}\right) \leq 0, \tag{4.13}$$

using the exponential on both sides

$$\sum_{e\in E} e^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j)} \leq 1 \Rightarrow 1 - \sum_{e\in E} e^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j)} \geq 0, \tag{4.14}$$

and finally, incorporating everything inside the summation sign

$$h(\boldsymbol{x}) = \sum_{e\in E} \tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{e\in E}\left(\frac{1}{|E|} - e^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j)}\right) \geq 0, \tag{4.15}$$

where $|E|$ denotes the cardinality of the set E. Additionally, the gradient of the CBF will be needed for the CBF condition (2.7)

$$\nabla h(\boldsymbol{x}) = \sum_{e\in E} \nabla\tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{e\in E} pe^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j))}\nabla h_e(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{4.16}$$

This result works for both collective and individual constraints and it is more conservative than just using the minimum function due to the approximation in (4.12).

Implementing this solution into (2.7) gives us

$$\nabla h^T(\boldsymbol{x})(\mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}) + \alpha(h(\boldsymbol{x})) \geq 0, \tag{4.17}$$

which can further be developed as

$$\sum_{e\in E} \nabla\tilde{h}_e^\top(\boldsymbol{x}_i, \boldsymbol{x}_j)\mathfrak{g}\boldsymbol{u} + \sum_{e\in E}\left(\nabla\tilde{h}_e^\top(\boldsymbol{x}_i, \boldsymbol{x}_j)\mathfrak{f} + \alpha(\tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j))\right) \geq 0, \tag{4.18}$$

The final step is to derive a formula for each $\boldsymbol{a}_i$ and b_i in (2.8) for each agent $i \in \mathcal{I}$.

For both collective constraints and individual constraints, the function $\alpha(h(\boldsymbol{x}))$ will be assumed linear with a constant parameter a, such that

$$\alpha(h(\boldsymbol{x})) = a \cdot h(\boldsymbol{x}). \tag{4.19}$$

4.2.2 Final formulation - Collective constraints

$$\boldsymbol{a}_i = -\sum_{e \in E} I_e(i) \nabla_{\boldsymbol{x}_i} \tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j) \mathfrak{g}_i^\top, \tag{4.20}$$

$$b_i = -\sum_{e \in E} I_e(i) \left(\nabla_{\boldsymbol{x}_i} \tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j)^\top \mathfrak{f}_i + \frac{a}{2} \left(\frac{1}{|E|} - e^{-p h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)} \right) \right), \tag{4.21}$$

where the indicator function $I_e(i)$ is

$$I_e(i) = \begin{cases} 1, & \text{if edge } e \text{ contains } i \\ 0, & \text{if edge } e \text{ does not contain } i \end{cases} \tag{4.22}$$

and $\nabla_{\boldsymbol{x}_i} h$ is the gradient of h with respect to the variables of the state vector of $\boldsymbol{x}_i$. For example, if $\boldsymbol{x}_i = (x_i, y_i)$ this becomes

$$\nabla_{\boldsymbol{x}_i} h = \begin{bmatrix} \frac{\partial h}{\partial x_i} \\ \frac{\partial h}{\partial y_i} \end{bmatrix} \tag{4.23}$$

Thus, we have shown that $\boldsymbol{a}_i$ and b_i only use local information of agent i and its neighbours $j \in N_i$.

4.2.3 Final formulation - Individual constraints

$$\boldsymbol{a}_i = -\sum_{k \in K} \nabla \tilde{h}_k(\boldsymbol{x}_i) \mathfrak{g}_i^\top, \tag{4.24}$$

$$b_i = -\sum_{k \in K} \left(\nabla \tilde{h}_k(\boldsymbol{x}_i)^\top \mathfrak{f}_i + a \left(\frac{1}{|\mathcal{I}||K|} - e^{-p h_k(\boldsymbol{x}_i)} \right) \right), \tag{4.25}$$

where K is the number of individual constraints on every agent $i \in \mathcal{I}$. For example, in a 2D problem where a certain polygon is to be avoided, each of the sides of the polygon would be an individual constraint.

4.2.4 Characterizing the error region

Since the equivalent QPs in (2.11) always satisfy the CBF condition, this new formulation will always fulfill it as well. However, due to the fact that now there is a summation of several terms for each $\boldsymbol{a}_i$ and b_i, this means that there are certain regions where $\boldsymbol{a}_i = 0$ and $\sum_{j \in N_i} (y_i - y_j) + b_i(\boldsymbol{x}_{loc,i}) > 0$.

Therefore, for those cases, the CBF conditions would be violated. Nevertheless, this region, which would be denoted as the "error" region has zero volume as it is only composed of individual points in the problem space. Therefore, these points are almost never reached, except of course when one initializes the problem in those points. Denoting this region as

$$\mathcal{R} = \{\boldsymbol{x} : \boldsymbol{a}_i(\boldsymbol{x}_{loc,i}) = 0 \text{ for some } i \in \mathcal{I}\} \tag{4.26}$$

There are different possible ways to solve this, the one used in this thesis work is the following

- Simple engineering fix: although this method does not ensure the safety of the system at all times, in practice, due to the zero volume characteristic of $\mathcal{R}$, it has worked for all of the simulations and experiments done for this paper. The idea is to set $y_i = 0$ when in $\mathcal{R}$ and let $\boldsymbol{u}_{nom,i}$ drive the agent i away from $\mathcal{R}$.

4.2.5 Experiments and simulations

For the distributed method, all experiments and simulations will have a HIL control element like the one shown in (4.3), a network system with 5 agents forming a K-formation as shown in Figure 4.1 and in 2D with $\boldsymbol{x}_i = (x_i, y_i)$.

For simplification purposes, only the CM constraint will be used, which implies the CBF function will be the following

$$h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) = r^2 - ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \geq 0, \tag{4.27}$$

where $\forall(i,j) \in E = \{(1,2),(2,3),(2,4),(2,5)\}$ and $||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = (x_i - x_j)^2 + (y_i - y_j)^2$.

The solution shown in (2.15) will be used, with the approximation shown previously in (4.15) for the five collective constraints. The gradient shown in (4.16) will also be used, with the specific gradient of the CM condition being

$$\nabla h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{bmatrix} \frac{\partial h_e}{\partial x_i} \\ \frac{\partial h_e}{\partial y_i} \\ \frac{\partial h_e}{\partial x_j} \\ \frac{\partial h_e}{\partial y_j} \end{bmatrix} = \begin{bmatrix} -2(x_i - x_j) \\ -2(y_i - y_j) \\ 2(x_i - x_j) \\ 2(y_i - y_j) \end{bmatrix} \tag{4.28}$$

for each collective constraint pair. As shown in the gradient formulation only two dimensions will be considered for this problem.

The simplified system dynamics considered are

$$\dot{\boldsymbol{x}} = \mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{u}. \tag{4.29}$$

In order to see the effect of the CM constraints over the formation controller used as the nominal, the safety distance will be smaller than the distance commanded by the nominal controller for agent 4 and 5. Therefore, the actual final formation, if safety is ensured, should look liked the constrained K shown in Figure 4.4.

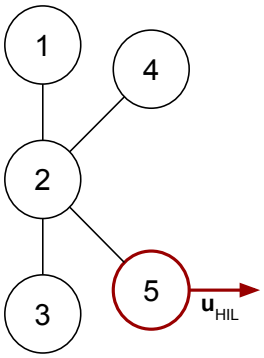


Figure 4.4: Constrained network system created by 5 agents.

Chapter 5

Results and Analysis

In this chapter, we present the results of the CBF-enabled mixer module described in the previous Chapter 4 and analyze and discuss them.

5.1 Major results

As before, both cases will be discussed. The centralized one with ROS simulations and experiments with real-robots and the distributed one with simpler simulations considering only one type of constraint (either one type of individual or one type of collective constraint) due to time limitations.

5.1.1 Centralized module

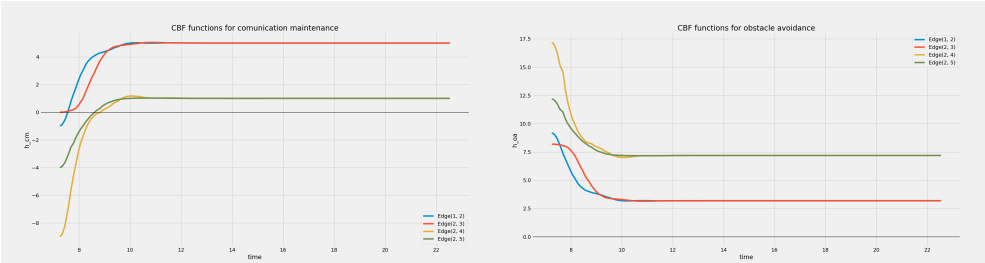
Four main simulations were performed to asses if the algorithm developed has achieved the desired properties. Attached to each of this there is a YouTube video link showing the evolution of the system with ROS.

- A normal K-shape formation with a wide enough CM and CA constraints to not modify the formation controller.

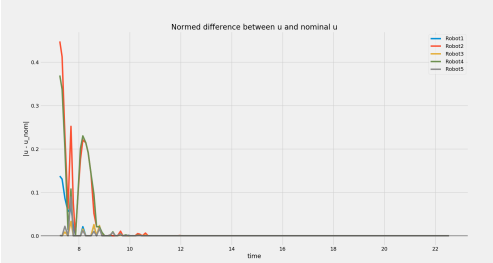
  As expected, both h_{CM} and h_{CA} are all greater than zero once the initial positions have been corrected. The normed difference between the nominal and final controller also converges to zero due to the constraints being satisfied by the nominal controller alone.

  The evolution of the system can be seen in the following video `https://youtu.be/pjV-o5fWWIo`.

- A constrained K-shape formation by using a smaller CM and OA constraints that will force the robots to converge to a smaller shape.

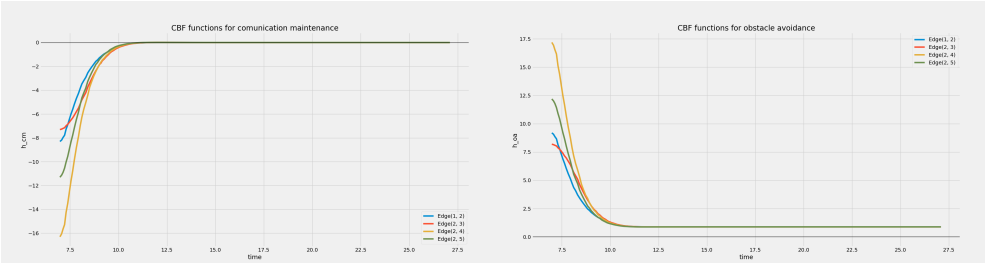


(a) CBF CM (b) CBF OA

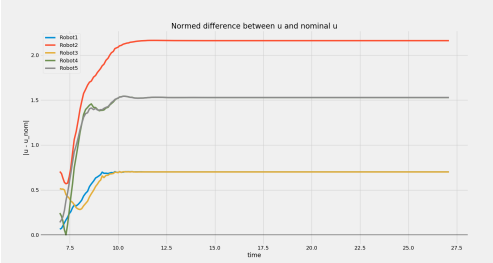


(c) Normed difference

Figure 5.1: Evolution of the CBF functions and normed difference for K with CM and CA.



(a) CBF CM (b) CBF OA



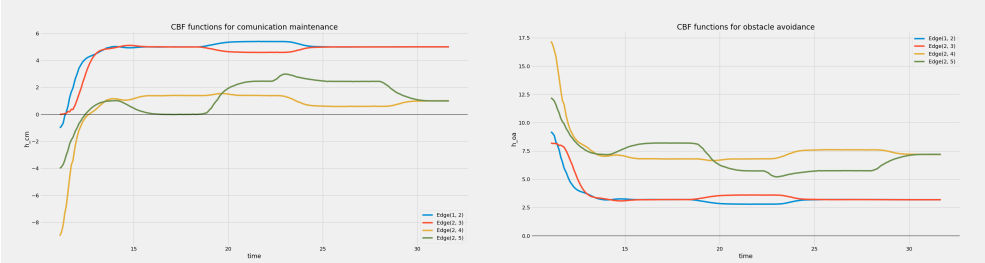
(c) Normed difference

Figure 5.2: Evolution of the CBF functions and normed difference for K-constrained with CM and CA.

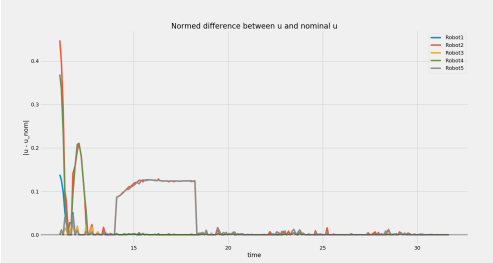
In this case, since the nominal controller alone does not satisfy the safety constraint, the final controller will always have a positive normed difference with the nominal one, as long as the CBF functions are non-negative.

The evolution of the system can be seen in the following video https://youtu.be/28yuAXJqv1c.

- The normal K with a HIL robot with CM and CA constraints.



(a) CBF CM (b) CBF OA



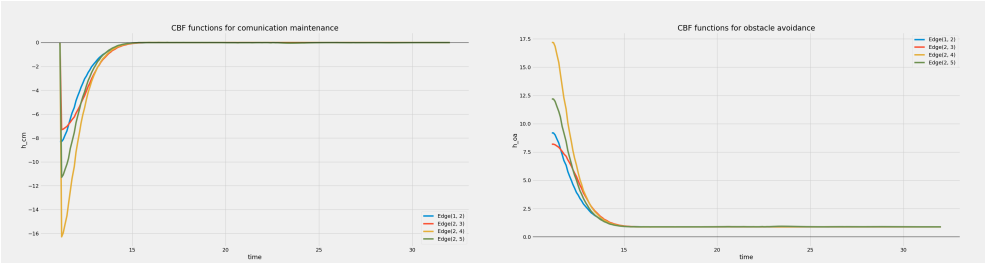
(c) Normed difference

Figure 5.3: Evolution of the CBF functions and normed difference for K with HIL and with CM and CA.

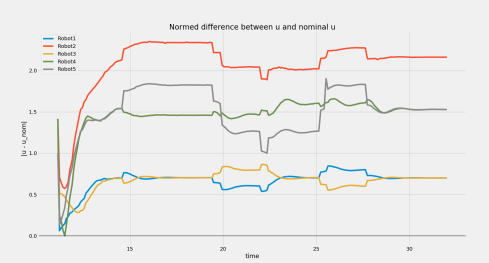
Only the case with the constraints is shown since the one without it was only use as a baseline for comparison. The main difference it has with the previous ones is that there are spikes and sections in the CBF functions were they go closer to zero, but they never cross the zero-line since the constraint have to always be satisfied. The normed difference plot shows the main region where the HIL was non-zero with the step and the latter smaller non-zero human commands of smaller time span.

The evolution of the system can be seen in the following video https://youtu.be/hLN_PQqiSMo.

- A constrained K with a HIL robot with smaller CM and CA constraints.



(a) CBF CM (b) CBF OA



(c) Normed difference

Figure 5.4: Evolution of the CBF functions and normed difference for K-constrained with HIL and with CM and CA

The same thing happens as before, but the normed difference never goes to zero as it was explained in the previous cases.

The evolution of the system can be seen in the following video https://youtu.be/43mtOdq0aIU.

Lastly, three experiments with real robots were performed. Again, each of them has a YouTube video link attached showing the evolution of the system recorded at the SML.

- 5 robots in a K-formation with one controlled by HIL.

  In this experiment the network of robots shown in Figure 4.1 will have the CM and CA constraints between each neighbour as well as the arena size constraint for each robot. The agent 5 is controlled by a human through the HIL shown in (4.3).

  The evolution of the system can be seen in the following video https://youtu.be/4USvQttIvHk.

- 5 robots in a K-formation with one controlled by HIL and a wedge shape constraint.

In this experiment the network of robots shown in Figure 4.1 with the wedge triangular shape constraint shown in Figure 4.2 will have the CM and CA constraints between each neighbour as well as the arena size constraint for each robot. The agent 5 is controlled by a human through the HIL shown in (4.3).

The evolution of the system can be seen in the following video https://youtu.be/nOUCDp-sQ7w.

- 5 robots in a K-formation avoiding a Human.

  In this experiment the network of robots shown in Figure 4.1 will have the CM and CA constraints between each neighbour as well as the arena size constraint for each robot. Each agent will also have the extra constraint shown in Figure 4.3 with the human actions defined by (4.4).

  The evolution of the system can be seen in the following video https://youtu.be/UFtF162iOEc.

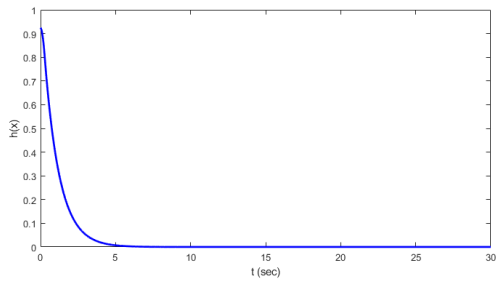
5.1.2 Distributed module

Only simulations with the MATLAB and Python simulators will be performed, in particular four different simulations to compare with and without HIL and with and without the simple fix.

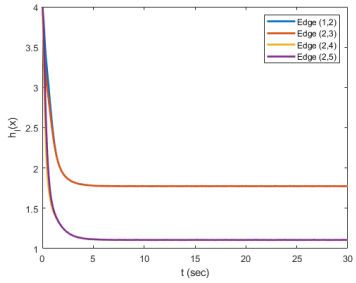
5.1.3 Connectivity maintenance for a K-formation with 5 agents

In this example, a network system with 5 agents forming a K-formation will be considered as shown in Figure 4.1. In addition, the agent 5 will be partially controlled by a human.

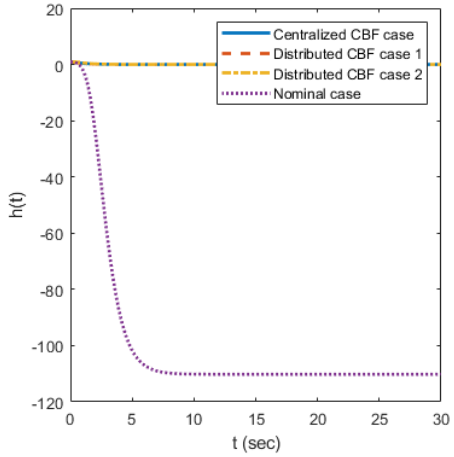
The results for this case can be seen in Figure 5.5. The combined CBF function $h(\boldsymbol{x})$ is always greater or equal than zero which proves the safety of the system, and this can be compared with the nominal case shown in the Subfigure 5.5 (c) which does not fulfill this safety condition. Additionally, when comparing the summation of all the constraints $h(\boldsymbol{x})$ to each individual one, the conservative behaviour of this method can be seen since the individual CBFs are higher than zero.



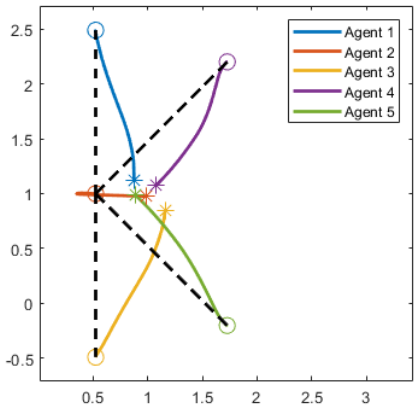
(a) Time evolution of $h(\boldsymbol{x})$.



(b) Time evolution of each $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$.



(c) Comparison of $h(\boldsymbol{x})$ for different methods.



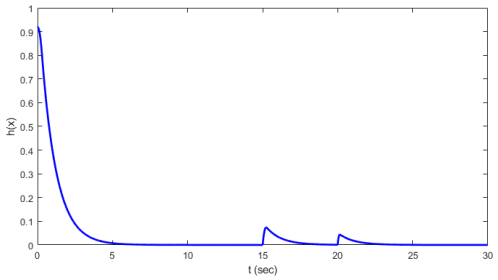
(d) Evolution of the MAS with time in the problem space.

Figure 5.5: Numerical results for 5 agents in a K-formation and a connectivity maintenance collective constraint between each neighbour.

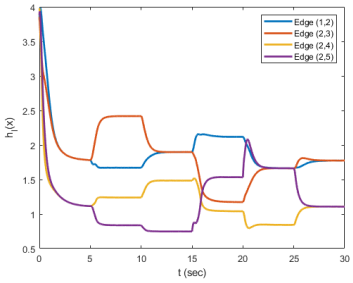
5.1.4 Adding HIL to the problem

In this case, one of the agents (shown in Figure 4.1 in red) will have a HIL element as shown in (4.3). The human will command a speed pattern such that the trajectory of agent 5 forms a rectangle. The results for the CBF functions and the time evolution can be seen in Figure 5.6.

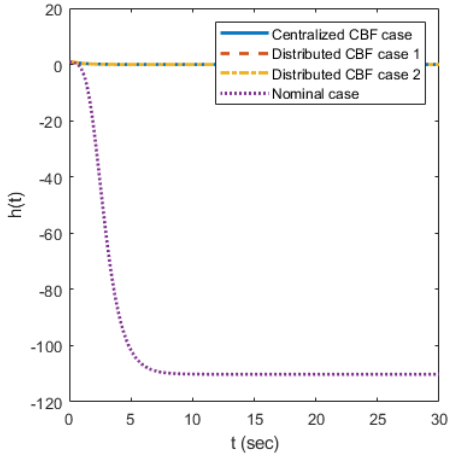
In order to visualize it properly, the following video shows the evolution of the whole system with time `https://youtu.be/l6IuXK7RytI`. Additionally, a simulation with 21 agents following the same formation and HIL pattern was also created, the main difference being that collision avoidance between all the agents has been used instead of CM, this can be seen in the following video `https://youtu.be/jTHEBdl4R4o`.



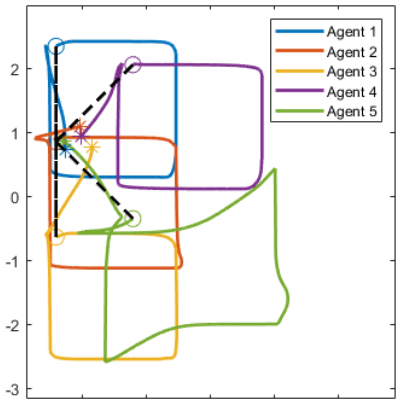
(a) Time evolution of $h(\boldsymbol{x})$.



(b) Time evolution of each $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$.



(c) Comparison of $h(\boldsymbol{x})$ for different methods.



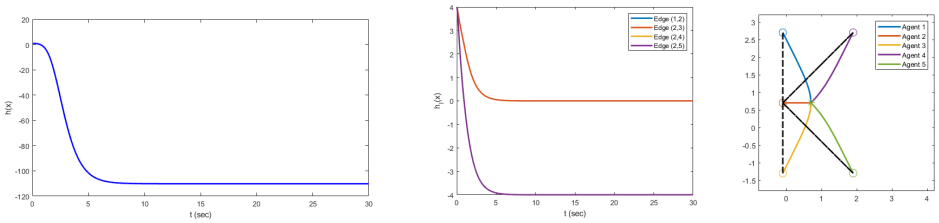
(d) Evolution of the MAS with time in the problem space.

Figure 5.6: Numerical results for 5 agents in a K-formation and a connectivity maintenance collective constraint between each neighbour when one has a HIL element.

5.1.5 Special case when $\boldsymbol{a}_i = 0$

To test what happens when the system is in the "error" region $\mathcal{R}$, in this simulation it will be initialized in one of those individual points where $\boldsymbol{a}_i = 0$. As it can be seen in Fig. 5.7, the CBF functions become negative and the MAS forms the non-constrained K, which is not inside the safe region.

However, this problem can be solved using the simple fix previously stated as it can be seen in Figure 5.8, where all CBF functions remain non-negative.

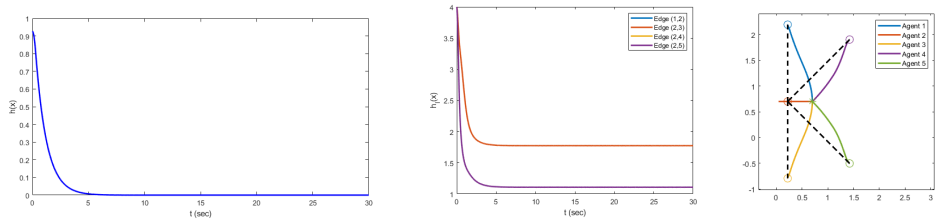


(a) Time evolution of $h(\boldsymbol{x})$.

(b) Time evolution of each $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

(c) Evolution of the MAS with time in the problem space.

Figure 5.7: Numerical results for 5 agents in a K-formation and a connectivity maintenance collective constraint between each neighbour initialized in $\mathcal{R}$.



(a) Time evolution of $h(\boldsymbol{x})$.

(b) Time evolution of each $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$.

(c) Evolution of the MAS with time in the problem space.

Figure 5.8: Numerical results for 5 agents in a K-formation and a connectivity maintenance collective constraint between each neighbour initialized in $\mathcal{R}$ and with the distributed fix.

5.1.6 Comments on implementation when $a_i \to \infty$

The other case when the QP controller will not work experimentally is due to the exponential used to compute $\boldsymbol{a}_i$ and b_i in (4.20), (4.24), (4.21) and (4.25). Since if $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is negative and $|h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)|$ is big enough, the limitations in the software used to compute the exponential would approximate it as infinity relatively quick. This can be solved by tuning the value of the parameters a and, more importantly, p to lower the argument of the exponential.

For example, for the simple collective constraint case shown before, if the agents are to be inside an arena where the largest dimension is D and the connectivity maintenance safety distance is d, the maximum negative value of the CBF function is $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) = d^2 - ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = d^2 - D^2$. If we know that the software used to compute the QP has a maximum limit for the argument of the exponential J, then in order to assure that the QP would work at any point inside the arena, the parameter p can be computed as $p \leq \frac{J}{d^2 - D^2}$.

5.2 Reliability and Validity Analysis

The last thing to consider is the reliability of the data and the analysis done as discussed in Chapter 3. Since all the CBFs encountered behaved as expected (that is asymptotically converging to the safe region and continuing as such for the rest of the time evolution of the system), we can consider the methods used to develop the algorithm to be reliable. The data can be consider reliable as well since the data obtained and used to compute the CBFs is also the same data used to compute the controllers, so even if there are small variations in the real-life system, the algorithm should still work with the data that it is given.

Chapter 6

Conclusions and Future work

6.1 Conclusions

In this work, a CBF-enabled mixer module for MASs with HIL control has been developed. Several multi-robot coordination algorithms and types of systems have been studied, but only the most representative of those has been used to show the performance of the module. Afterwards, MATLAB, Python and ROS simulations, as well as experiments with the real-world NEXUS robots have been carried out to show the correct operation of the algorithm.

Additionally, an existing distributed implementation for CBF-induced QPs for multi-agent systems was adapted to allow for more than one constraint through the use of the minimum function approximation. This allows for each agent to solve its local QP while the MAS still remains safe. Additionally, the regions where the algorithm might fail were analyzed and some corrections to fix this problem where proposed. Several simulations to show the behaviour of this new controller together with the simple fixes are demonstrated to prove the performance and safety of the proposed implementation.

6.2 Limitations

The main limitation has been the time constraint, which restricted the study of different multi-robot coordination protocols, types of system and the safety constraints used.

The distributed implementation has also been somewhat limited due to time, since more testing with the ROS simulator and specially the real-world robots needs to be completed.

To really execute a decentralized implementation the code must be distributed to each of the NEXUS robots and this implies more work on modifying the existing ROS packages for NEXUS.

Lastly, more complicated HIL implementations could have been considered, like collaborative manipulation, but again the time constraints as well as the scope of the thesis did not allow for it.

To all of this is added the limitations in the computational power of the processor used, which did not allow for extensive realistic simulations with a high number of robots.

Nevertheless, even though the results only reflect the performance of the module for a subset of all the possible situations, the generality of the methods used is still comprehensive enough to assess the validity of the module developed.

6.3 Future work

As explained before, due to the breadth of the problem and its alternatives, some things have been left to finish. In general all the goals have been met, but some possible future work still exist. In this section we will focus on some of the remaining issues that should be addressed in the continuing work. The general summary is presented in Table 6.1.

6.3.1 What has been left undone?

Specifically, the goal one about the study of MAS and goal three about the specific design of the CBF-enabled mixer could have seen more work done on them.

6.3.1.1 Quadcopters & satellites

The time limitations did not allow to consider other types of vehicles than the NEXUS robot, but the SML has capability to perform experiments with quadcopters apart from ground vehicles. Therefore, the mixer module could have been tested with UAV swarms to show the generality of the results.

In addition to this, satellite constellations could have also being considered. Initially the spacecraft dynamics were studied and a CBF module to keep a certain number of satellites as far away as possible while still visible between each other through a visibility function was created, but again due to time constraints it was not finished.

6.3.1.2 Multi-robot coordination protocols

A future research effort is needed to test the algorithm when different nominal controller are used. Rather than formation controllers, coverage controllers can be tested since they have a lot of synergies with Connectivity Maintenance CBF-induced constraints.

6.3.1.3 More HIL cases

HIL can be done in a lot of ways and one of the most interesting ones is when humans collaborate with the robot on tasks they are better at, like manipulation. More work could also have been done on coming up with new collaborative schemes.

6.3.2 Improvements on the distributed approach

In particular, the author of this thesis wishes to point out that the distributed approach only works for one specific type of constraint at a time and even though more than one constraint can be used, the computations done to solve the minimum approximation function quickly grow out of bounds. Solving this problem and maybe even founding a new approach is the next thing that should be done.

Additionally, experiments with the real-world NEXUS robots need to be performed in the same manner that have been completed for the centralized module.

Table 6.1: Major improvements left to do for each background area.

	Multi-robot system	**HIL**	**CBF**
Improvements	- Quadcopters & satellites - Trying coverage controller	- Design new collaborative tasks	- Improve distributed approach for more constraints

Lastly a new provably-safe fix to the "error" region needs to be studied. This can be done through the use of higher-order CBFs [31], for example

$$h_1(\boldsymbol{x}) = \nabla h^T \mathbf{f} + \alpha_1(h), \tag{6.1}$$

when in $\mathcal{R}$, together with a special selection of α

$$\alpha_1(h(\boldsymbol{x})) = \boldsymbol{x}^T \boldsymbol{x} h, \tag{6.2}$$

to break recursivity of the higher-order CBFs. Several higher-orders might be needed to ensure all the points of $\mathcal{R}$ are taken into consideration in the corrections. In this way, the original CBF condition of the multi-agent system would always be fulfilled, even when $\boldsymbol{a}_i = 0$.

6.3.3 Codes and scripts

The codes developed for the simulators, ROS pakages and the mixer module itself would eventually be published in the author personal GitHub `https://github.com/ViktorNfa`. For now, the code will remain private since a publication is being prepared with this thesis work and some future work.

6.4 Reflections

As mentioned before, one of the most important results of this thesis is the improvements added to the distributed method from [26], as well as the real-world experiments discussed in Chapter 5, which show what the module mixer is capable of doing. More specifically, this research has a lot of synergies with the CANOPIES project [32], partly done by the DCS at KTH, where collaborative tasks between human workers and multi-robot teams are done for precision agriculture.

References

[1] S. Martínez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *Control Systems, IEEE*, vol. 27, pp. 75 – 88, 09 2007.

[2] C. R. d. Cos and D. V. Dimarogonas, "Adaptive cooperative control for human-robot load manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5623–5630, 2022.

[3] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[4] "Drone light shows powered by intel," https://www.intel.co.uk/content/www/uk/en/technology-innovation/intel-drone-light-shows.html, accessed: 2022-06-13.

[5] "A record-breaking drone show ended with quadcopters falling from the sky," https://gizmodo.com/a-record-breaking-drone-show-ended-with-quadcopters-fal-1825745673, accessed: 2022-06-13.

[6] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu, and F. Gao, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, 2022.

[7] J. Breeden and D. Panagou, "Guaranteed safe spacecraft docking with control barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 2000–2005, 2022.

[8] "Global positioning system," https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS.html, accessed: 2022-06-13.

[9] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.

[10] I. Nourbakhsh, K. Sycara, M. Koes, M. Yong, M. Lewis, and S. Burion, "Human-robot teaming for search and rescue," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 72–79, 2005.

[11] W. Li, D. Sadigh, S. Sastry, and S. Seshia, "Synthesis for human-in-the-loop control systems," vol. 8413, 04 2014. ISBN 978-3-642-54861-1 pp. 470–484.

[12] D. Zhou and M. Schwager, "Assistive collision avoidance for quadrotor swarm teleoperation," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1249–1254.

[13] K. Fitzsimons, E. Tzorakoleftherakis, and T. D. Murphey, "Optimal human-in-the-loop interfaces based on Maxwell's Demon," in *2016 American Control Conference (ACC)*, 2016, pp. 4397–4402.

[14] A. Broad, T. Murphey, and B. Argall, *Operation and Imitation Under Safety-Aware Shared Control*, 05 2020, pp. 905–920. ISBN 978-3-030-44050-3

[15] P. Long, T. Keleştemur, A. . Önol, and T. Padir, "Optimization-based human-in-the-loop manipulation using joint space polytopes," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 204–210.

[16] B. Xu and K. Sreenath, "Safe teleoperation of dynamic UAVs through control barrier functions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7848–7855.

[17] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015, analysis and Design of Hybrid Systems ADHS.

[18] M. Machida and M. Ichien, "Consensus-based control barrier function for swarm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8623–8628.

[19] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[20] L. Lindemann and D. V. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic

tasks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1916–1928, 2020.

[21] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems," *IEEE Control Systems Letters*, vol. 1, no. 2, pp. 310–315, 2017.

[22] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015, analysis and Design of Hybrid Systems ADHS.

[23] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.

[24] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.

[25] Y. Chen, M. Santillo, M. Jankovic, and A. D. Ames, "Online decentralized decision making with inequality constraints: An admm approach," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2156–2161, 2021.

[26] X. Tan and D. V. Dimarogonas, "Distributed implementation of control barrier functions for multi-agent systems," *IEEE Control Systems Letters*, vol. 6, pp. 1879–1884, 2022.

[27] "Ros - robot operating system," https://www.ros.org/, accessed: 2022-06-14.

[28] "Gazebo - simulate before you build," https://gazebosim.org/home, accessed: 2022-06-14.

[29] "4wd mecanum wheel mobile arduino robotics car 10011," https://www.nexusrobot.com/product/4wd-mecanum-wheel-mobile-arduino-robotics-car-10011.html, accessed: 2022-06-14.

[30] "Qualisys motion capture," https://www.qualisys.com/, accessed: 2022-06-14.

[31] X. Tan, W. Shaw Cortez, and D. V. Dimarogonas, "High-order barrier functions: robustness, safety and performance-critical control," *IEEE Transactions on Automatic Control*, Early access, 2021.

[32] "Canopies," https://www.canopies-project.eu/, accessed: 2022-06-17.

TRITA-EECS-EX- 2022:00