Tampere University

Anirudh Aynala

Coordination of Multi Robot Systems for Warehouse Automation

Faculty of Engineering and Natural Sciences (ENS)
Examiners: Assistant Professor Azwirman Gusrialdi
Associate Professor Reza Ghabcheloo
Master of Science thesis
December 2021

Abstract

Anirudh Aynala: Coordination of Multi Robot Systems for Warehouse Automation
Master of Science thesis
Tampere University
Degree Programme in Automation Engineering, MSc. (Tech)
December 2021

The recent advancements in robotics have enabled intelligent robotic systems to be deployed in an industrial environment to automate tasks and increase productivity. Mobile robots deployed in a warehouse are usually designed to do tasks individually and do not interact with one another. Although this increases efficiency in the performance of tasks over time, some tasks may be too complicated for a single robot system to accomplish. Multi-robot systems have tremendous potential in automating tasks in logistics and warehouses where robot teams can perform multiple tasks in parallel in less time compared to working individually. The prospect of these multi-robot systems autonomously working in the same environment allows tackling time constraints and dependence on human labour. But, in order to carry packages in a warehouse one of the challenges for the robots is to maintain a rigid formation shape and perform set tasks cooperatively while keeping the whole system safe from static and dynamic entities. In this thesis work, a real-time distributed control algorithm based on control barrier functions for cooperative control of multi-robot systems is presented. The main objective of the controller is to enable a team of robots to maintain a formation while avoiding obstacles and other robot formations present in their way and reaching their goal. The proposed controller has been verified and its performance evaluated through various scenarios in both simulations and physical mobile robots.

Keywords: Multi-robot systems, Control Barrier Functions, Cooperative control, Obstacle avoidance.

The originality of this thesis has been checked using the Turnitin Originality Check service.

PREFACE

I would like to sincerely thank my supervisors Asst. Prof. Azwirman Gusrialdi and Mr. Widhi Atman for their guidance during writing my thesis and for giving me the chance to work on this intriguing topic. I would like to specially thank Prof. Azwirman Gusrialdi for his mentorship through out the process.

Finally I would like to thank my parents for supporting me constantly and encouraging me to pursue my studies.

Tampere, 20th December 2021

Anirudh Aynala

Contents

List of Figures

List of Tables

List of Symbols

k_p	Gain of P controller
γ	Extended class $\mathcal{K}$ function
a	Length of major axis of Ellipse
b	Length of minor axis of Ellipse
v	Linear velocity of the robot
ω	Angular velocity of the robot
$\mathcal{G}$	Communication Graph
$\mathcal{A}$	Weighted adjacency matrix
h^{obs}	safety function for robot obstacle avoidance
h^{formg}	safety function for formation distance greater case
h^{forml}	safety function for formation distance lesser case
h^{Af}	safety function for avoiding other robot formations
Ds	Distance of robot from obstacle
Df	Distance between two robots in a formation
ϵ	Tolerance for a robot in maintaining formation distance

Abbreviations

AMR	Autonomous Mobile Robot
UAV	Unmanned Arial Vehicle
QP	Quadratic Program
PSO	particle swarm optimization
MOPSO	multi-object particle swarm optimization
ROS	Robot Operating system

1 Introduction

1.1 Motivation and Thesis Objective

In modern-day industries, there is increasing development towards automating warehouse distribution centres and logistics [18]. The author in [15] discusses the developments and challenges in deploying large numbers of Autonomous mobile robots(AMR) for warehouse automation

As Industry 4.0 aims to provide more flexible production systems through incorporating new technologies, robotics has been one of the key elements in striving towards it. The role of AMR's in production networks and how they can be utilized to develop the performance was discussed in [20].

Industrial warehouses and factories around the globe remain heavily dependant on human labour for strenuous and repetitive tasks which take quite a time to be finished. The operation of these warehouses is also limited by the time constraints of human labour as one cannot be working all day. In particular, during challenging times where the human workforce comes under-hit, the work in these industries would come to stop.

An example would be the effect of the COVID-19 pandemic where the production numbers did take quite a damage in trying to keep up with demand. Warehouses were short in man force to cope up with the surge in demand, the article in [34] discusses the impact of COVID on logistic systems and the food supply chain. This further encourages the need for automating hard labour.

This is where robots come into the scene, there are already well developed robotic systems for warehouse works such as Amazon's KIVA robot[1], Locus robot[9] from Locus Robotics, BOLT[11] from IAMRobotics, Vector and MAvek from Waypoint Robotics[24] are a few of the latest mobile robots being developed and deployed in the warehouses and distribution centres for a variety of tasks such as surveillance, security, package displacement etc.

Warehousing and logistics mainly involve human-run vehicles to pick and place the goods from one place to other. AMRs are being introduced in several Intralogistics operations mainly to ease the human labour involved in transferring the goods in an enclosed environment.

The main value in replacing manual labour is the efficiency at which the work can be done as these autonomous robots can work longer and more precise than their human counterparts. Intelligent warehouse systems allow us to tackle this issue of constant human supervision required to successfully run the day to day tasks in an industrial warehouse and automate it.

The prospect of using multi-robot systems in such scenarios where product moving vehicles can be replaced by a team of robots that are relatively smaller in size but together can perform the same task with minimal human intervention would potentially be one of the main features of a fully automated warehouse.

These multi-robot fleets offer more flexibility, speed and robustness in limited spaces or in presence of dynamic obstacles compared to their bigger counterparts. The advanced hardware and control software present in these robots allow safe autonomous operation in dynamic environments or situations i.e. avoiding obstacles and avoiding other robots.

One of the main ingredients to realize such a system is a scalable control algorithm implemented in each robot which guarantees safe operation of the multi-robot system.

The objective of this thesis work is to develop control algorithms allowing multiple mobile robots to cooperatively execute a task while ensuring their safe operation in a possibly dynamic environment.

As a use case, a scenario illustrated in Figure 1.1 will be considered in which a fleet of mobile robots move in a formation from one point to a pre-defined location while maintaining the predefined shape of the formation (.i.e, the smooth motion of the robots within the formation correspond to translation or rotation of the whole formation) necessary to cooperatively transport a load and at the same time avoiding collision with static/dynamic obstacles and other fleets of robots.

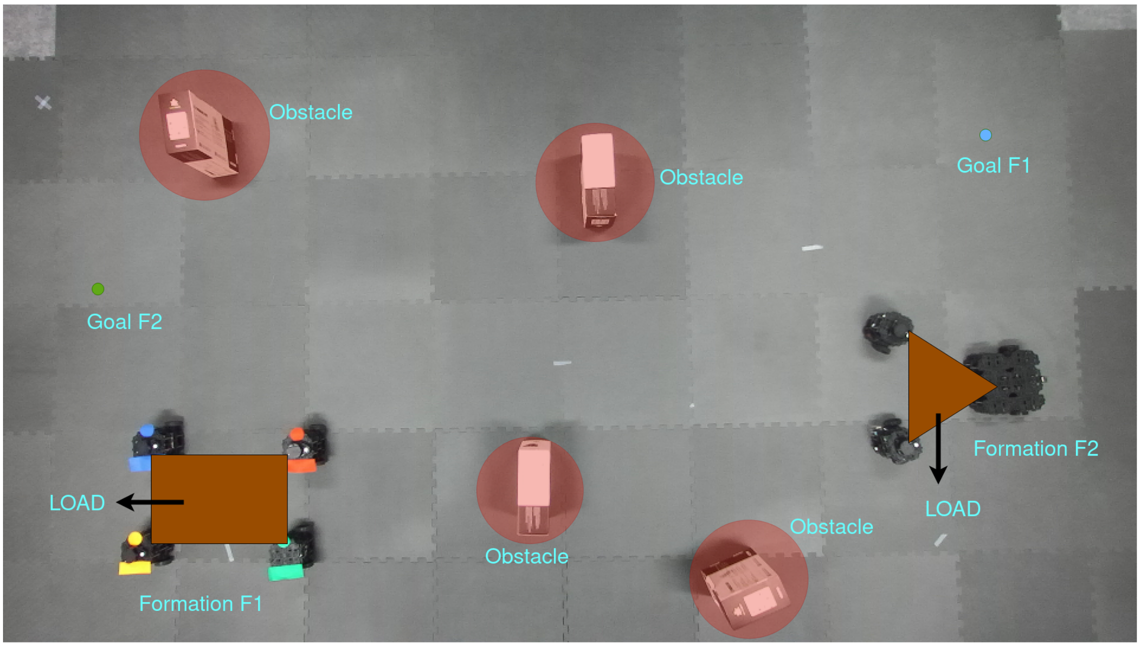


Figure 1.1 *A warehouse scenario of robot formations navigating through obstacles to reach desired goal positions.*

1.2 Research Questions

To achieve the previously mentioned objectives, this thesis aims at answering the following research questions.

1. How to design a real-time algorithm for controlling the movement of a group of mobile robots to accomplish a task with multiple and possibly conflicting control objectives? For example, in the scenario illustrated in Figure 1.1, the robots need to move in a formation to reach a predesignated goal location while avoiding collision and maintaining the shape of the formation.

2. How to implement the control algorithm in a distributed manner? This means that each robot should be able to compute its action by cooperating (i.e., exchanging information via a communication network) with several other robots and in the absence (or with very limited assistance) of a control centre. Distributed control algorithm is highly desirable as it provides several potential advantages such as scalability to the system's size and robustness concerning the failure of the individual system. In addition, this thesis also aims to answer the question of how the communication network topology between the robots needs to be designed to reduce the communication cost and thus allow the high-penetration of multi-robot systems in the warehouse.

1.3 Related Work

The following section talks about the existing work related to the objectives of this thesis work. There has been quite some research and development being carried out in the area of developing control schemes for a multi-robot system to perform required tasks cooperatively.

1.3.1 Formation Control of Multi-Robot Systems

This subsection presents related work to formation control of multi-robot systems where a formation shape is achieved by the robots by communicating with each other.

The main goal is to maintain certain formation and be able to execute assigned tasks such as passing through narrow passages etc. There has been extensive research on the formation control of multi-robot systems.

General approaches for formation maintenance in multi robot s involve the use of potential fields[38], behaviour based formation control techniques [10], Lyapunov functions [28], virtual leader based formation control [37] and Leader follower consensus [35].

The formation control problem has also been approached with different communication topology as in decentralised strategies as [3], optimal feedback with graph theory [16] and information flow based model [17]. The main aim of these control strategies is to establish a cooperative control between the multi-robot systems where the cost of communication is low as each other only performs communication with neighbouring robots.

The authors in [27] give an in-depth analysis of the requirements and objectives for formulating a set of control strategies based on the task defined constraints to have a cooperative motion planner with required formation control for stability and robustness.

The aforementioned methods so far neglect the discussion of rigid formations in their formation control strategies while navigating through an environment.Several discussions of rigid formations have been presented in [7],[22]. However, their capabilities haven't been tested in a warehouse environment facing dynamic entities while preserving their rigid formations.

1.3.2 Obstacle avoidance in Multi-robot Systems

This subsection presents related work to obstacle avoidance in robot systems to enable safe operation while performing tasks where there are several dynamic entities in the path of the robots.

The topic of obstacle avoidance has been well researched over the years and a wide range of methods can be used depending on the sensors deployed on the robot. The most common methods being used are potential fields [25] and Virtual force fields [12].

Some other methods where the constraints of the robot, both dynamic and kinematic are considered in [19]. In [32] obstacle avoidance for a swarm of robots is presented where a rotational potential field is applied to the mobile robots, it also discusses maintaining formation while avoiding the obstacles, which is one of the major aims of this thesis. The article [30] reviews several well-researched navigation methods for robots in a dynamic environment such as modelling a controller for navigation of swarm robots using particle swarm optimization (PSO). The authors in [2] proposed a navigation method based on PSO where the navigation problem is considered as an optimization problem where it searchers for a solution with the minimum value, whereas the authors in [39] have proposed a multi-object particle swarm optimization(MOPSO) which addresses the problem of finding the optimal path for the robots in an uncertain dynamic environment.

Control barrier certificates are one of the popular methods to enable proof of safety for dynamic systems. In [36] a safety feedback design is proposed using barrier certificates and control Lyapunov functions for control tasks that might change over time. In [13] a swarm safety control barrier certificates with an optimization-based controller to ensure forward invariance of safe operating set and generate collision-free operation with minimal impact on the underlying control laws is proposed and verified on swarm robots. In [31] a safety verification for hybrid systems to prove the trajectories of the system do not enter unsafe regions is proposed to provide proof for system safety.

1.3.3 Multi-robot formations while ensuring safety

This subsection presents related work to multi-robot formation control while ensuring the robot systems are safe.

In [26] a model is presented which tackles the problem of developing a controller for load transporting system in an industrial environment where, the model uses flocking algorithm and potential functions for formation control and obstacle avoidance for a fleet of mobile robots respectively.

The objective for the multi-robot systems presented in the paper is a similar case to the objective of this thesis. However, the discussion regarding robots maintaining a rigid formation while navigating through the environment or while avoiding obstacles is missing.

The author in [23] addresses formulation for a model based on control barriers to navigate two UAVs in the presence of obstacles. However, the formation control law presented was limited to two UAVs navigating while avoiding static obstacles, the discussion of multi-robot formations avoiding dynamic obstacles and other formation of robots was missing.

The control algorithms discussed above for formation control, obstacle avoidance and robot formations avoiding obstacles, allow multi-robot systems to perform various tasks cooperatively. But, in a warehouse to transfer packages using small robots could be quite challenging as the robots should maintain the formation shape to not drop the package. There can be several robots operating in the warehouse and they need to perform their tasks coherently without interfering with one another.

1.4 Contributions of the thesis

The contributions of the thesis are summarized as follows:

- Real-time control algorithm based on control barrier function is proposed which allows individual robots to cooperatively accomplish a given task. The control barrier function-based approach serves as a unified framework to efficiently handle tasks with multiple and possibly control objectives as illustrated in Figure 1.1

- A method to reduce the required communication (information exchange) between the robots is proposed by the benefits offered by the control barrier function.

- Evaluation and demonstration of the proposed control algorithm in real-world experiments using mobile robots.

1.5 Thesis Outline

The remainder of the thesis is outlined as follows:

1. Chapter 2 provides a brief literature review of the techniques used for the cooperative control between multi-robot systems to perform the desired task. This chapter focuses on graph theory to model communication topology between the robots and the literature behind control barrier functions.

2. Chapter 3 focuses on establishing the problem statement on which the system is modelled, it also discusses the modelling of specifications for designing the control algorithm for cooperative control of multi robot systems.

3. Chapter 4 discusses the designing of a safety controller for multi-robot systems using desired specifications discussed in previous chapter.
4. In Chapter 5 the proposed controller in the previous chapter is evaluated on a simulator and the parameters which affect the performance of the system are analysed.
5. In Chapter 6 the implementation of the proposed controller on physical systems and the required framework to get it up and running are discussed.
6. Chapter 7 presents the conclusion of the thesis work and discusses the possible future work.

2 Theoretical Background

In this chapter a brief outline of graph theory and how it can be used to model the interactions (e.g., communication) between the robots and to address formation shape's maintenance are discussed. Furthermore, this section also provides a brief review of control barrier functions which will be a key ingredient in developing the control algorithms.

2.1 Graph Theory

Graph theory is defined as the study of graphs .i.e, mathematical models to establish pairwise relations between entities involved. Graphs can be categorised as undirected graphs and directed graphs, where undirected means there is no definite distinction between two endpoints of an edge .i.e. one cannot pinpoint the direction of information flow.

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ consists of a non empty node set $\mathcal{V} = \{1, 2, ..., n\}$, an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and a weighted adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$ with non negative entries corresponding to the weights of the edges.

The existence of a edge $(i, j) \in \mathcal{E}$ denotes that node j can directly obtain information from node i or vice versa. Each node of a graph has neighbours, the neighbours of node i can be defined as,

$$\mathcal{N}_i = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\} \tag{2.1}$$

In a multi-robot system represented by a graph the nodes represent robots and the edges represent the communication link between them, thus graphs can be used to model how multi-robot systems interact with one another.

In terms of formation maintenance, we opt for an undirected graph because of the tendency of the robots as vertices of a graph edge to maintain the distance constraint between them.

2.1.1 Rigid Graph

There has been a lot of study on the rigidity of graphs, The authors in [21] introduces rigidity theory and its respective application to networks. Rigid graphs are important to analyse the concept of rigid formations for a set of robots arranged in a graph schema.

Let us consider the edges of a graph as bars connecting the nodes. Moreover, it is assumed that the nodes are free to move continuously under the constraint that the length of the bars do not change. A graph is considered rigid if every motion of the nodes preserves the distances of all pairs of nodes. The Figure 2.1 shows an undirected rigid graph.

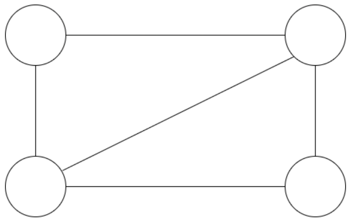


Figure 2.1 *An illustration of an undirected rigid graph*

In terms of the formation of mobile robots, the formation is called rigid if the only smooth motion of the robots within the formation are those corresponding to translation and rotation of the formation. In [8] a theory for rigid graph control for both undirected and directed graph architecture is provided for modelling of control formations for autonomous robots.

2.2 Control Barrier Functions

In control systems, the most important objective is to keep the system safe, mainly in AMRs the priority to safely execute the tasks is demanded.

Control barrier functions allow us to incorporate this inherent safety priority into the nominal controller. The authors in [4] and [6] introduces us to control barrier functions and how they can be used to enforce safety properties on to the behaviour of our system. These safety barrier certificates guarantee forward set invariance .i.e the system always stays in the safe set if it starts in the safe set.

Let us consider a nonlinear affine control system assumed of form:

$$\dot{x} = f(x) + g(x)u \tag{2.2}$$

with f and g locally Lipschitz, $x \in D \subset \mathbb{R}^n$ and u $\in U \subset \mathbb{R}^m$ where D and U are open sets. As discussed in [6] stability is considered as to enforcing invariance of set, where a set $\mathcal{C}$ is defined as a super level set of continuously differentiable function $h : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ which gives us,

$$\begin{aligned} \mathcal{C} &= \{x \in D \subset \mathbb{R}^n : h(x) \geq 0\} \\ \partial\mathcal{C} &= \{x \in D \subset \mathbb{R}^n : h(x) = 0\} \end{aligned} \tag{2.3}$$

For the system to be safe the forward invariance of set $\mathcal{C}$ is must and from [6] the safety is defined for any initial condition $x_0 \in D$ there exists a maximum interval of existence $I(x_0) = [0,\tau_{max}]$ such x(t) is unique solution to (2.2) on $I(x_0)$. In this case safety is defined by:

Definition 2.2.1 *[6] The set $\mathcal{C}$ is forward invariant if for every $x_0 \in \mathcal{C}$, x(t) $\in \mathcal{C}$ for x(0) $= x_0$ and all $t \in I(x_0)$. The system (2.2) is safe with respect to the set $\mathcal{C}$ if the set $\mathcal{C}$ is forward invariant*

In order for h to be a control barrier function it should render $\mathcal{C}$ invariant but not its sublevel sets as stated in [6], Let us note an extended class $\mathcal{K}_\infty$ function is a function $\gamma : \mathbb{R} \rightarrow \mathbb{R}$ that is strictly increasing and with $\gamma(0) = 0$;

Definition 2.2.2 *[6] Let $\mathcal{C} \subset D \subset \mathbb{R}^n$ be the superlevel set of a continuously differentiable function h: D$\rightarrow \mathbb{R}$, then h is a control barrier function(CBF) if there exists an extended class $\mathcal{K}_\infty$ function γ such that for the control system (2.2):*

$$\sup_{u \in U}[L_f h(x) + L_g h(x)u] \geq -\gamma(h(x)). \tag{2.4}$$

for all $x \in D$.

Here the extended class $\mathcal{K}_\infty$ function is used to control how fast our system approaches the perimeter of $\mathcal{C}$ function. This allows us to regulate it using the function of choice. The set consisting of all control values that render $\mathcal{C}$ safe are given by:

$$K_{\mathrm{CBF}}(x) = \{u \in U : L_f h(x) + L_g h(x)u + \gamma(h(x)) \geq 0\}. \tag{2.5}$$

As defined in [6], the existence of control barrier function implies that the control system is safe:

Theorem 1 *[6] Let $\mathcal{C} \subset \mathbb{R}^n$ be a set defined as the superlevel set of a continuously differentiable $h : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$. If h is a control barrier function on D and $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$,then any Lipschitz continuous controller $u(x) \in K_{CBF}(x)$ for the system (2.2) renders the set $\mathcal{C}$ safe and asymptotically stable in D.*

Therefore, Let us note that control barrier functions provide maximum probability for necessary safety as state in [5] and [6]

Theorem 2 *[6] Let $\mathcal{C}$ be a compact set that is the superlevel set of a continuously differentiable function h: D$\rightarrow \mathbb{R}$ with the property $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$. If there exists a control law $u = k(x)$ that renders $\mathcal{C}$ safe, i.e., $\mathcal{C}$ is forward invariant with respect to (2.2).*

In [29] the notion of function h that defines the safe set $\mathcal{C}$ explicitly on time is considered. Let us consider the control affine form (2.2) and to ensure the forward invariance of time-varying set $\mathcal{C}(t) \subseteq D \subset \mathbb{R}^n$ defined by the superlevel set of function $h : D \to \mathbb{R}$

$$\mathcal{C}(t) = \{x \in D \subset \mathbb{R}^n : h(x,t) \geq 0\} \tag{2.6}$$

The Time varying control barrier functions which ensure the forward invariance of time-varying set $\mathcal{C}(t) \subseteq \mathbb{R}^n$ as given in [14] as:

Definition 2.2.3 *[14] Given dynamic system* (2.2) *and set* $\mathcal{C}(t)$ *in* (2.6), *function h is time-varying control barrier function defined on D with $\mathcal{C}(t) \subseteq D \subset \mathbb{R}^n$, if there exists a locally Lipschitz extended class $\mathcal{K} function$ γ such that $\forall x \in D$ and* $\forall t \in [t_0, t_1]$,

$$\sup_{u \in U} \left[\frac{\partial h}{\partial t} + L_f h(x,t) + L_g h(x,t)u + \gamma(h(x,t)) \right] \geq 0. \tag{2.7}$$

holds. From condition (2.7) the set which consists of all the control values that render $\mathcal{C}(t)$ safe can be given by:

$$K_{\text{CBF}}(x,t) = \left\{ u \in U : \frac{\partial h}{\partial t} + L_f h(x,t) + L_g h(x,t)u + \gamma(h(x,t)) \geq 0 \right\}. \tag{2.8}$$

Therefore, the above-reviewed concepts of control barrier functions can be utilized such that, to achieve the desired objectives, they need to be formulated as control barrier functions and the safe set obtained can be used as a constraint for designing the controller.

3 Problem Statement

The main problem considered to be solved in thesis work is to navigate a team of mobile robots through a warehouse environment while maintaining a rigid formation with a set tolerance and avoiding obstacles and other robot formations in their way. In this section, we first discuss the kinematic model of the mobile robot followed by formally formulating the problem.

For control design purpose a simple model with single integrator dynamics is considered. Let us consider a robot i with motion dynamics given by:

$$\dot{p_i} = u_i \tag{3.1}$$

where $p_i \in \mathbb{R}^2$ is the planar position of robot, and u_i is its input velocity.

Consider a warehouse consisting of n mobile robots and static obstacles. The robots are divided into c groups of robots where each group consists of c_i robots. For the sake of simplicity and with no loss of generality, this thesis considers the case where c_i is either equal to two or four.

Furthermore, let us assign a rigid graph $\mathcal{G}^k = \{\mathcal{V}^k, \mathcal{E}^k, \mathcal{A}^k\}$ to the kth group of robots, where the entries of the weighted adjacency matrix $\mathcal{A}^k$ represent distance that needs to be maintained between pair of robots representing an edge $(i, j) \in \mathcal{E}^k$ given by $Df_{i,j}$.

The kinematics of each robot is given by a single integrator(3.1). Assuming that each robot knows its location the objective is to design cooperative control,

$$u_i = q_i(p_j), \quad j \in i \bigcup \mathcal{N}_i \tag{3.2}$$

where $\mathcal{N}_i$(to be defined later) represents a neighbouring set of robot i from/to which robot can receive/send information so that all the robots reach their goal position while satisfying the following specifications:

1. The robots within each group need to maintain the designated shape. This constraint can be mathematically formulated as the following inequality:

$$-\epsilon^2 \leq \left\|p_i - p_j\right\|^2 - Df_{ij}^2 \leq \epsilon^2 \quad \forall \quad (i, j) \in \mathcal{E}^k \tag{3.3}$$

   where ϵ is the tolerance introduced to add flexibility to the system (for example, due to the flexibility of the robot's gripper/manipulator).

2. The robots in the warehouse need to avoid collisions with static obstacles,

which can be mathematically formulated as the following inequality.

For O number of static obstacles with each obstacle's centre denoted by $p_j^{obs} \in \mathbb{R}^2$. Assuming the obstacles can be approximated by a circle or sphere, at anytime t, the robot needs to satisfy the constraint

$$\left\|p_i(t) - p_j^{obs}\right\| > D_{sj} \quad \forall i \in \mathcal{V}^k,\ \forall j \in \{1, \ldots, O\} \tag{3.4}$$

where D_{sj} is the distance the robot needs to maintain from the obstacle j.

3. The robots in one formation group need to avoid collisions with robots from another formation group. This constraint can be mathematically formulated as the following inequality.

   Consider two formations m and k with c_i robots, for any robot i from group m to avoid any robot j from group k, the constraint can be given by,

$$\left\|p_i - p_j\right\|^2 > D_{Af} \quad \forall i \in \mathcal{V}^m, \forall j \in \mathcal{V}^k, \quad m \neq k \tag{3.5}$$

   where D_{Af} is the distance assumed such that each robot in formation m would be able to avoid all the robots of the formation k.

The information regarding the obstacle's position is available to all of the robots in the formation. Finally, it is assumed that at the initial positions $p_i(0),\ \forall I \in \mathcal{V}^k$, all the above constraints are satisfied for the robots.

4 Design of cooperative control with safety guarantee

In this chapter, the formulation of the control architecture is discussed. In the following sections, a controller based on barrier certificates and formulation of the constraints that the robots need to satisfy is presented. This optimal control input from the safety controller for the robot systems is such that the robots always operate in a safe region.

4.1 QP-based Controller

The objective is that the controller needs to needs to be minimally invasive and modify the nominal controller as little as possible unless safety conditions are being violated. In this work, a Quadratic Programming-based controller is opted.

Quadratic programming is optimizing a quadratic function based on linear equality and inequality constraints. Since we model our specifications for the robot systems as linear inequalities, we use a QP controller which can find minimal control input subject to the constraints imposed, this results in the case where the controller tries to keep both nominal and actual control command as near as possible unless safety is under threat.

Therefore for ensuring safety the QP controller is given by,

$$\begin{aligned} \mathbf{u}_i^* = \underset{\mathbf{u} \in \mathbb{R}^2}{\operatorname{argmin}} \quad & J(\mathbf{u}_i) = \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 \\ \text{s.t. specifications (3.3), (3.4), (3.5) are satisfied.} \end{aligned} \tag{4.1}$$

where $\hat{\mathbf{u}}_i$ is the control input from the nominal controller, $\mathbf{u}_i^*$ is the actual control input which is fed to the robots. The control input from $\mathbf{u}_i^*$ would remain the same as the nominal controller .i.e, $\hat{\mathbf{u}}_i$ when the system is safe and only alters it when safety is at risk.

4.1.1 Design of Nominal Controller

A nominal controller is used to drive the robots to their goal positions. A P-controller is used as the nominal controller where the objective of the controller is to provide control input which is proportional to the distance between the robot i's position and corresponding goal position.

$$u_i = k_p(G_i - p_i), \qquad k_p > 0 \tag{4.2}$$

where k_p is the proportional gain and G_i is the desired goal position for the robot.

4.2 Formulation of constraints for QP solver using Control barrier functions

The QP controller proposed in (4.1) allows the system to be safe and executes the required objectives given to the nominal controller. This safety can be achieved by incorporating the specifications as defined in chapter 3 into the constraints in (4.1).

The following section discusses how to couple defined constraints with the robot's kinematic model. The constraints are modelled using control barrier functions discussed in section 2.2 and are incorporated into the controller.

Maintaining Formation

Let us start formulating constraints for formation maintenance defined in earlier chapter for the multi robot systems as control barrier functions.

The equation(3.3) gives rise to two inequality distance formulations between two robots in the formation. The safety barrier functions can given by

$$h_{ij}^{formg}(p_i) = \left\|p_i - p_j\right\|^2 - (Df_{ij} - \epsilon)^2 \geq 0 \quad \forall (i,j) \in \mathcal{E}^k \tag{4.3}$$

$$h_{ij}^{forml}(p_i) = (Df_{ij} + \epsilon)^2 - \left\|p_i - p_j\right\|^2 \geq 0 \quad \forall (i,j) \in \mathcal{E}^k \tag{4.4}$$

Obstacle avoidance

In a dynamic environment, for the robots to avoid collisions with O number of obstacles in its way and keep the system(package) safe, from (3.4) a safety function for robot i can be defined as an inequality as,

$$h_i^{obsj}(p_i) = \left\|p_i - p_{obsj}\right\|^2 - D_s^2 \geq 0 \quad \forall i \in \mathcal{V}^k,\ \forall j \in \{1, \ldots, O\} \tag{4.5}$$

Avoiding other formations with elliptical safety function

Let us consider the specification (3.5) defined for avoiding collision between two formations k and m. For every robot from formation k to avoid every robot from formation m, the robots of both formations can individually try to avoid the other formation's robots. However, the cost of communication and processing would be much higher if it approached that way.

Instead, the idea is:

- As the formation maintenance specification ensures that each formation can only translate or rotate as a whole. Therefore, the formation's shape can be approximated by a circle or an ellipse.
- If information regarding the formation's centroid, orientation and dimensions of it's shape are known to each robot in the opposite formation, they can avoid the ellipse while moving and as a result of this collision between all robots in any two formations can be avoided.
- This could be achieved with a reduced communication cost where the leaders from each formation exchange information between them and share it with robots in their respective group. The robot which can communicate with all other robots in the formation is chosen as the leader. It would enable one to model the constraint of avoiding formation of robots as a safety function.

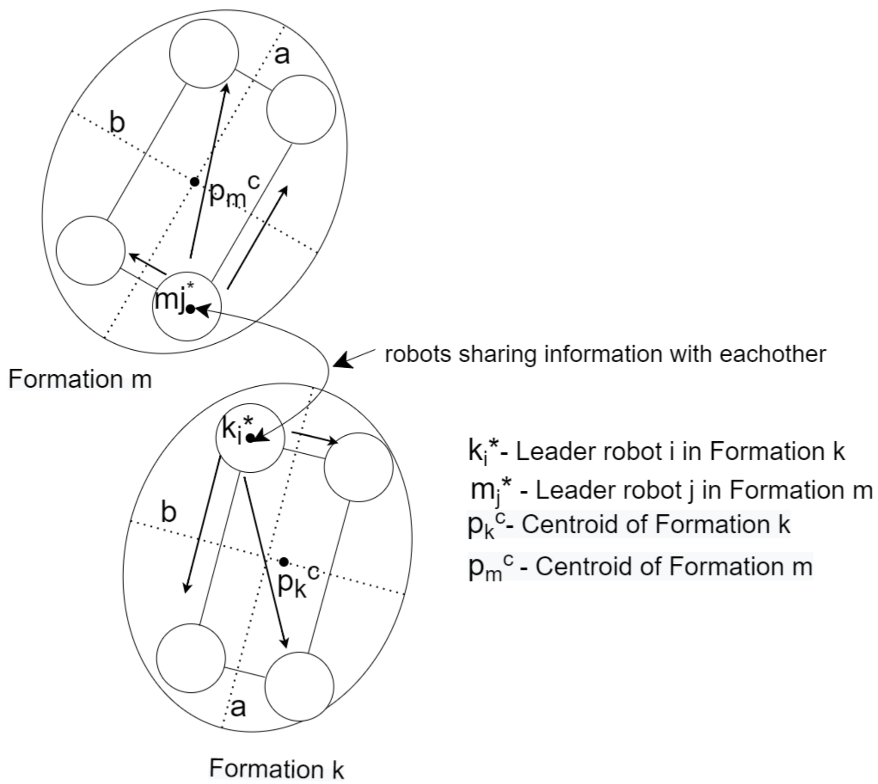


Figure 4.1 *Illustration of two four-robot rectangular formations k and m with similar shapes trying to avoid each other. The formations are engulfed by elliptical barrier approximated through information exchange between leaders. a,b are the major and minor axis approximated from the dimensions of formation shape.*

In this thesis work, this safety function is modelled to be an ellipse with its centre at the other formation's centroid, the major and minor axis of the ellipse are approximated from the dimensions of formation's shape. The orientation of the formation is also used in formulating the ellipse. This would create a control barrier certificate enclosing the entire formation such that robots cannot enter the barrier enclosing opposite formation as shown in Figure 4.1.

The motivation behind choosing an ellipse safety function is that the approximation using ellipse provides more flexibility in maneuvering compared to a circle as it has an additional degree of freedom .i.e, rotation.

Figure 4.1 also shows an illustration of how information is exchanged between two robot formations through leader robots k_i^* and m_j^*, these leader robots share information regarding the opposite formations with other members in the group through topologies given by $\mathcal{G}^k = \{\mathcal{V}^k, \mathcal{E}^k, \mathcal{A}^k\}$ and $\mathcal{G}^m = \{\mathcal{V}^m, \mathcal{E}^m, \mathcal{A}^m\}$ respectively.

A safety function for robot i from formation k to avoid robots from formation m can be given as,

$$\begin{aligned} h_i^{Afc}(p_i) = &\left(\frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}\right)(p_{ix} - x_m^c)^2 + \left(\frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}\right)\left(p_{iy} - y_m^c\right)^2 \\ &+2\left(p_{ix} - x_m^c\right)\left(p_{iy} - y_m^c\right)\cos\alpha\sin\alpha\left(\frac{1}{a^2} - \frac{1}{b^2}\right) - 1 \geq 0 \quad \forall i \in \mathcal{V}^k,\ m \neq k \end{aligned} \tag{4.6}$$

where $p_i = [p_{ix}, p_{iy}]$ is the position of robot i and $p_m^c = [x_m^c, y_m^c]$ is the centre of formation m in robot i's path. Here a and b are major and minor axis of the ellipse whose orientation is given by α respectively.

The centroid of formation m can be formulated as,

$$p_m^c = \frac{1}{|\mathcal{V}^m|}\sum_{j=1}^{|\mathcal{V}^m|} p_j \tag{4.7}$$

Complete constraints for QP Solver

As proposed in section 2.2 after defining the safety functions, the control space in which the multi-robot system can stay safe is given by (2.2), accordingly using the same procedure, all the constraints mentioned above are only withheld if u_i is chosen such that,

$$\frac{\partial h_i^{obsj}(p_i)}{\partial p_i}u_i \geq -\gamma(h_i^{obsj}(p_i))\ \forall i \in \mathcal{V}^k,\ \forall j \in \{1, \ldots, o\} \tag{4.8}$$

$$\frac{\partial h_{ij}^{formg}(p_i)}{\partial p_i}u_i \geq -\gamma(h_{ij}^{formg}(p_i)) \quad \forall (i,j) \in \mathcal{E}^k \tag{4.9}$$

$$\frac{\partial h_{ij}^{forml}(p_i)}{\partial p_i}u_i \geq -\gamma(h_{ij}^{forml}(p_i)) \quad \forall (i,j) \in \mathcal{E}^k \tag{4.10}$$

$$\frac{\partial h_i^{Afc}(p_i)}{\partial p_i}u_i \geq -\gamma(h_i^{Afc}(p_i)) \quad \forall i \in \mathcal{V}^k,\ m \neq k \tag{4.11}$$

are held with a locally Lipschitz extended class $\mathcal{K}$ function γ.

Therefore QP controller with constraints which allow robots to obtain above mentioned objectives can be given by:

$$\mathbf{u}_i^* = \underset{\mathbf{u}\in\mathbb{R}^2}{\text{argmin}} \quad J(\mathbf{u}_i) = \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 \tag{4.12a}$$

$$\text{s.t.} \ \frac{\partial h_i^{obsj}(p_i)}{\partial p_i}u_i \geq -\gamma(h_i^{obsj}(p_i)) \ \forall i \in \mathcal{V}^k \quad \forall j \in \{1,\dots,o\} \tag{4.12b}$$

$$\frac{\partial h_{ij}^{formg}(p_i)}{\partial p_i}u_i \geq -\gamma(h_{ij}^{formg}(p_i)) \quad \forall (i,j) \in \mathcal{E}^k \tag{4.12c}$$

$$\frac{\partial h_{ij}^{forml}(p_i)}{\partial p_i}u_i \geq -\gamma(h_{ij}^{forml}(p_i)) \quad \forall (i,j) \in \mathcal{E}^k \tag{4.12d}$$

$$\frac{\partial h_i^{Afc}(p_i)}{\partial p_i}u_i \geq -\gamma(h_i^{Afc}(p_i)) \quad \forall i \in \mathcal{V}^k,\ m \neq k \tag{4.12e}$$

Here different extended class $\mathcal{K}$ functions can be used for each inequality constraint but for the sake of simplicity it's chosen as γ, a positive gain.

Thus a cooperative controller with a safety guarantee and desired specifications is successfully designed.

5 Evaluation Using Numerical Simulations

In this chapter proposed algorithm from the previous section is implemented in a simulated environment. Various scenarios described below allow extensive testing and analysis of the performance of the algorithm on multi-robot systems.

5.1 Formation avoiding obstacles

In this section, a four robot formation is made to maneuver through static and dynamic obstacles and reach a predefined goal position in a simulated environment.

5.1.1 Four robot formation avoiding static obstacles

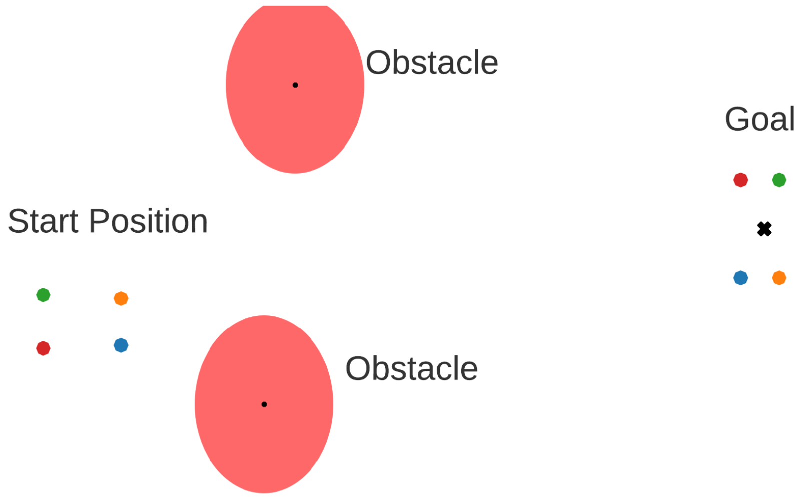


Figure 5.1 *Illustration of scenario for a four robot(Blue- 1,Orange- 2,,Green -3, Red - 4) formation with obstacles, start and goal positions.*

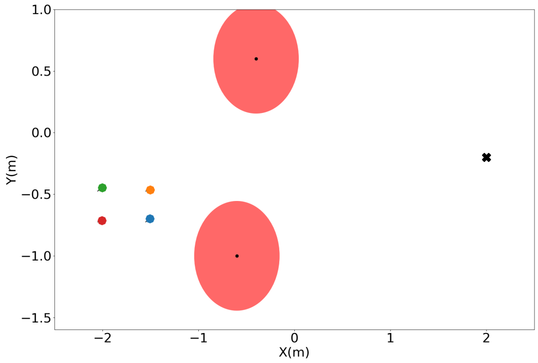
Table 5.1 *parameter settings for four robot formation to avoid obstacles.*

k_p	γ	$D_s(m)$	L(m)	B(m)
2	1	0.3	0.5	0.25

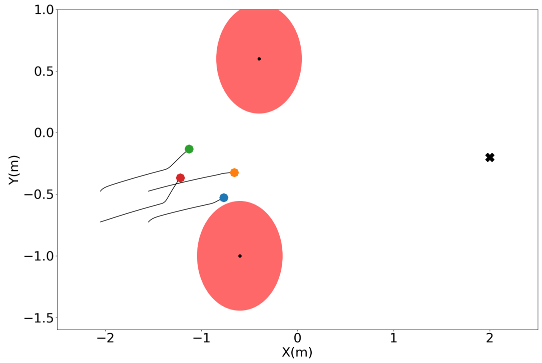
In Figure (5.1) a four robot formation with a rectangular shape is considered. The simulation is modelled out such that the controller would be thoroughly tested to uphold the specifications defined in section **??** for formation maintenance and obstacle avoidance. The parameters related to the control input calculation are given in Table 5.1

The weighted adjacency matrix used for the four robot formation shown in Figure 5.1 with dimensions considered in Table 5.1 is given by:

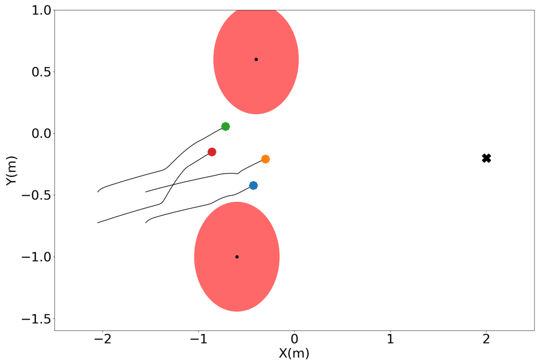
$$\mathcal{A} = \begin{bmatrix} 0 & 0.5 & 0 & 0.25 \\ 0.5 & 0 & 0.25 & 0.559 \\ 0 & 0.25 & 0 & 0.5 \\ 0.25 & 0.559 & 0.5 & 0 \end{bmatrix} \tag{5.1}$$



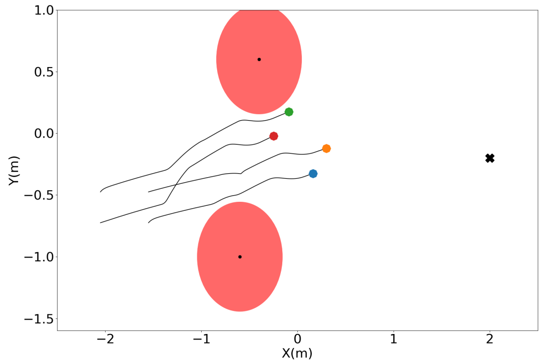
(a) t = 1s



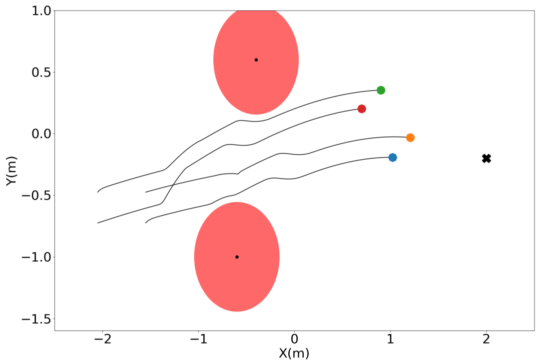
(b) t = 25s



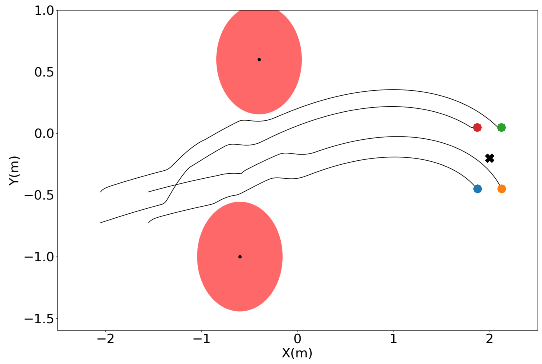
(c) t = 40s



(d) t = 65s



(e) t = 90s



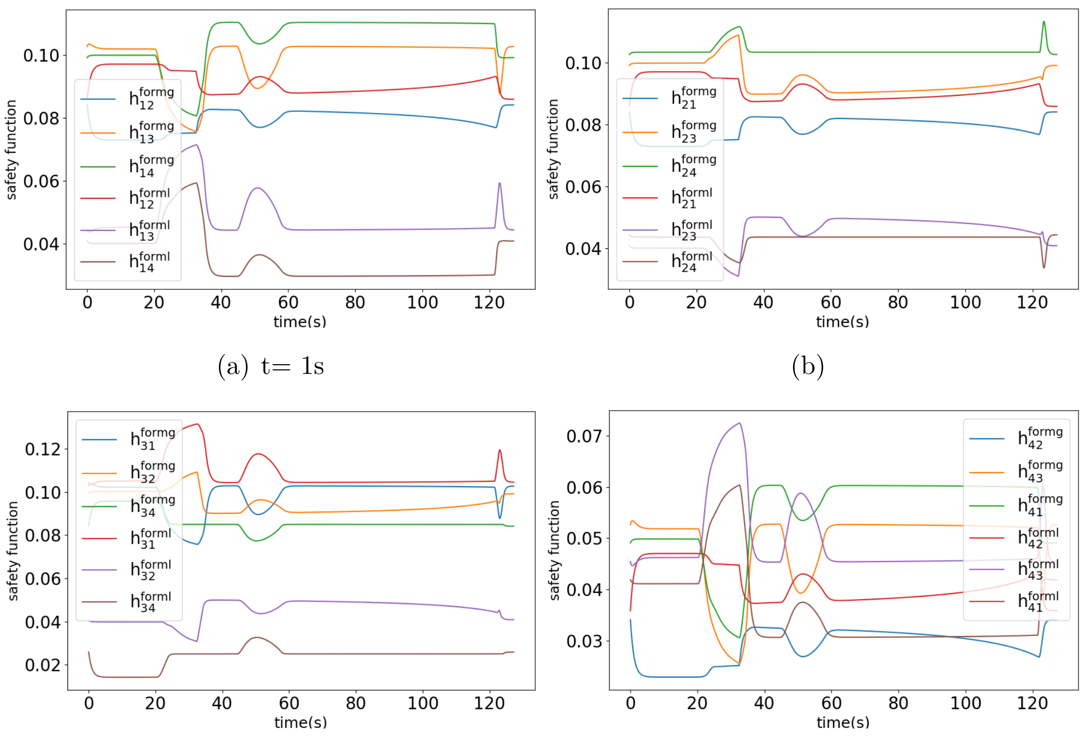
(f) t = 120s

Figure 5.2 *Snapshots of a four robot formation avoiding two static obstacles and reaching goal position.*

In Figure 5.2 a simulated result of formation of four robots successfully avoiding two static obstacles and reaching its goal position is shown, here about t = 25s and t = 65s the four robots maneuver such that they keep the formation intact and the controller's control input to the robots enabled them to avoid the obstacles.

The performance of the controller can be evaluated by analysing the plots of the safety function over time. The following safety functions defined in section 4.2 need to stay positive through out the simulation.

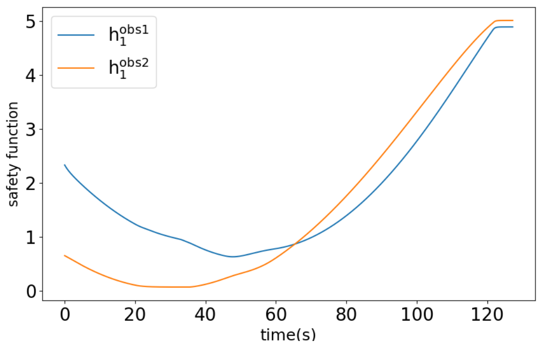
$$\begin{bmatrix} h_i^{obsj}(p_i) \\ h_{ij}^{formg}(p_i) \\ h_{ij}^{forml}(p_i) \end{bmatrix} \geq 0 \tag{5.2}$$



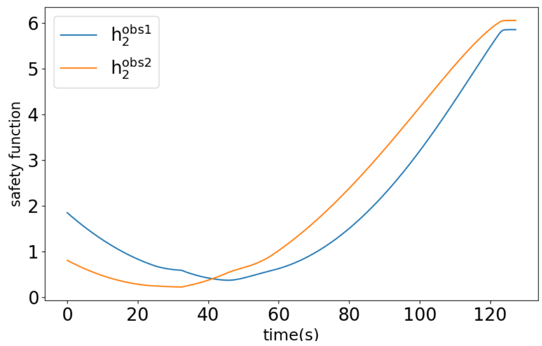
(a) t= 1s (b)

(c) (d)

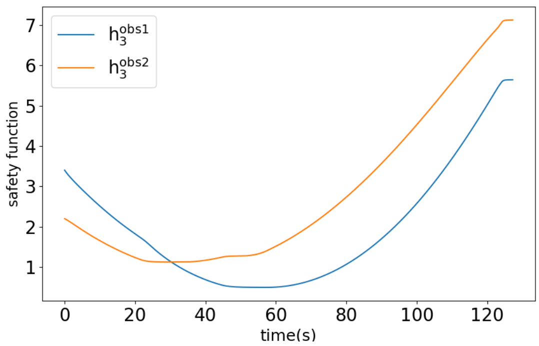
Figure 5.3 *Plots of safety function for formation maintenance for each robot.*



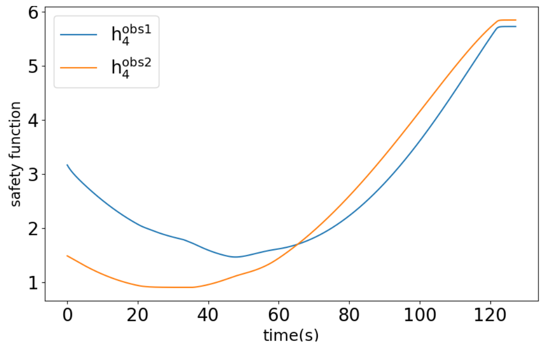
(a) t= 1s



(b)



(c)



(d)

Figure 5.4 *Plots of safety function for obstacle avoidance for each robot.*

In Figure 5.4 a plot consisting of the safety functions related to the obstacle avoidance performed by the Four-robot formation. The safety functions at t = 40s to 60s get close to zero as the formation moves closer to the obstacle, the control input given by the controller is such that it allows the robot formation to maneuver such that they stay positive which is desired behaviour.

The formation was able to successfully pass through the obstacles while maintaining its formation in tact and reach the goal position as shown in the figure. 5.1.

5.1.2 Four robot formation avoiding dynamic obstacles

In this scenario a four robot formation is made to go to a goal position while avoiding a moving obstacle as shown in the Figure. 5.5. The obstacle is moving with a velocity of 5×10^{-2} m/sec. The four robot formation needs to maneuver such that while maintaining the formation the robot needs to avoid the dynamic obstacle in its way.

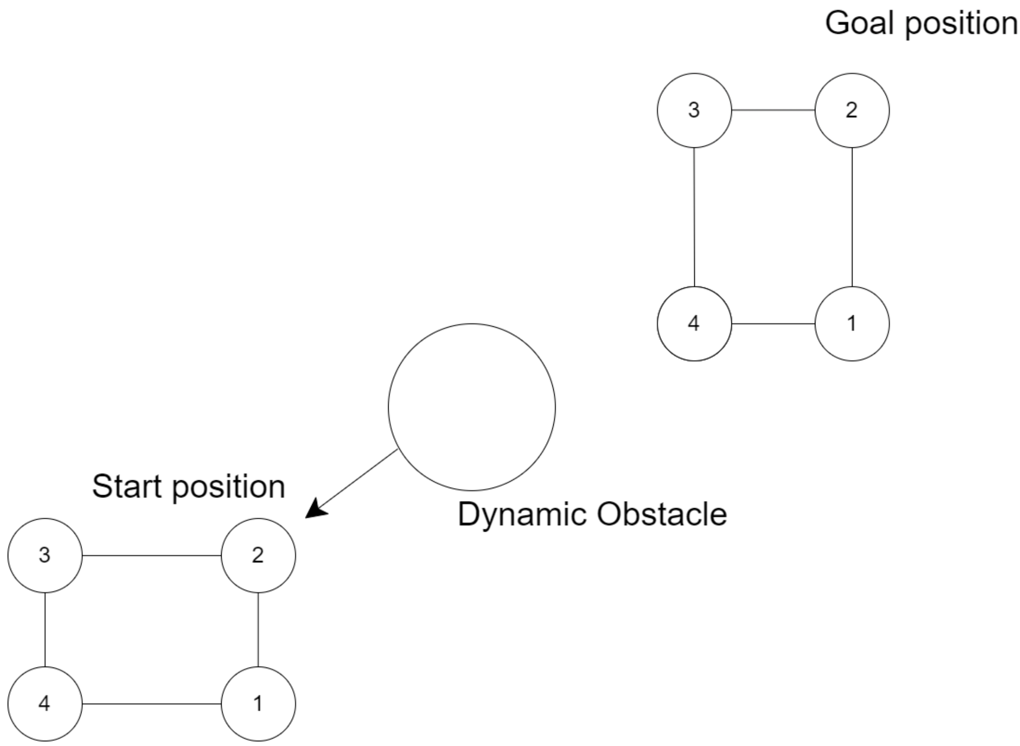


Figure 5.5 *Illustration of the scenario of robot formation at start position, moving obstacle and robot at goal position.*

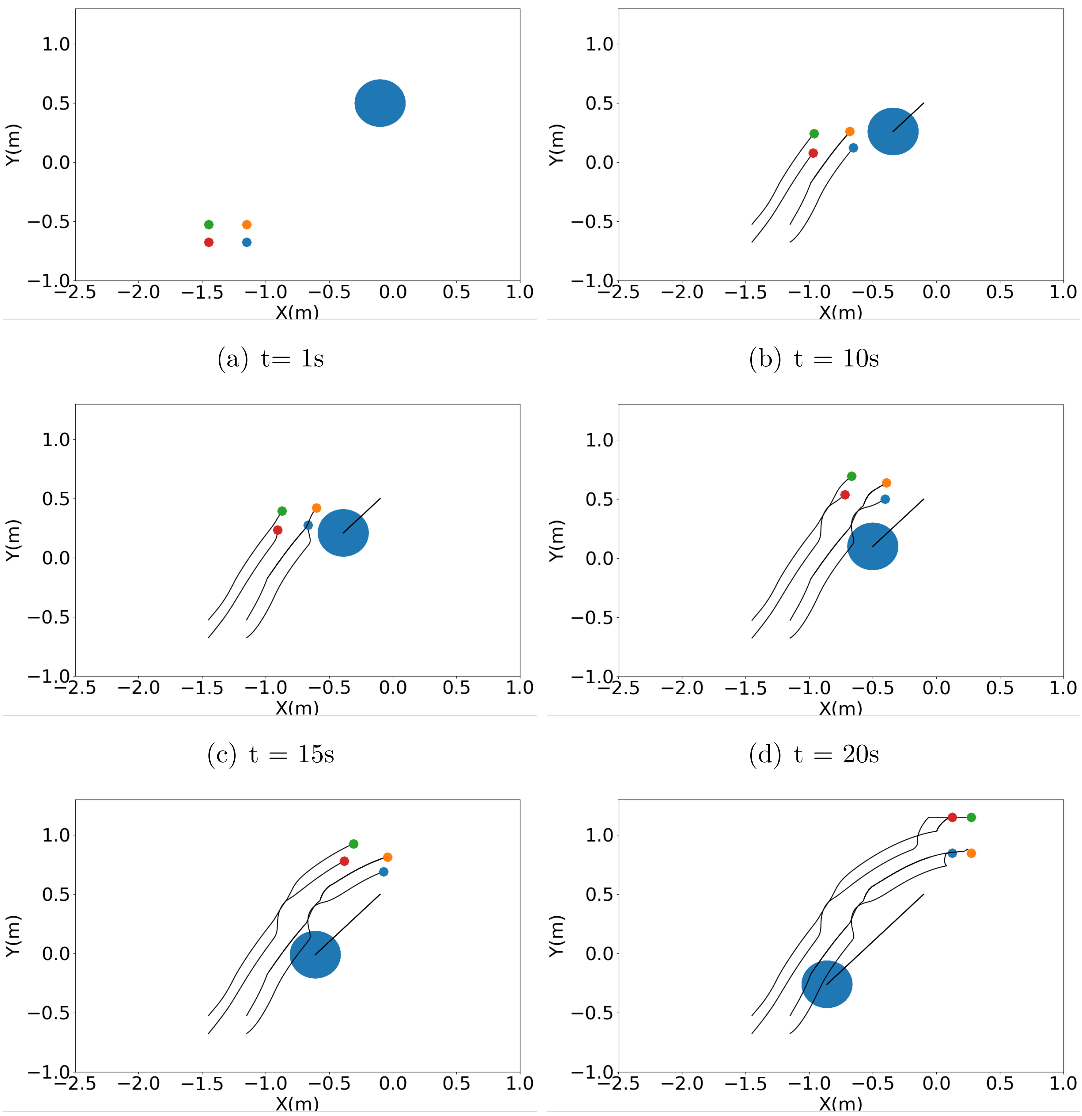
The parameters related to calculation of robot control input are given in Table 5.2 and weighted adjacency matrix for four robot formation is given in (5.1).

Table 5.2 *parameter settings.*

k_p	γ	$D_s(m)$	L(m)	B(m)
2	0.5	0.3	0.5	0.25

If we choose a higher γ such that asking the robot formation to move as close as possible to the obstacle it would lead to violation of the specification of obstacle avoidance, as the robot formation may not be able to react to the moving obstacle.

This would violate the specification of obstacle avoidance (3.4) incorporated into the controller, therefore a suitable γ can be tuned such that the objective of avoiding dynamic obstacles can be achieved.



(a) t= 1s (b) t = 10s

(c) t = 15s (d) t = 20s

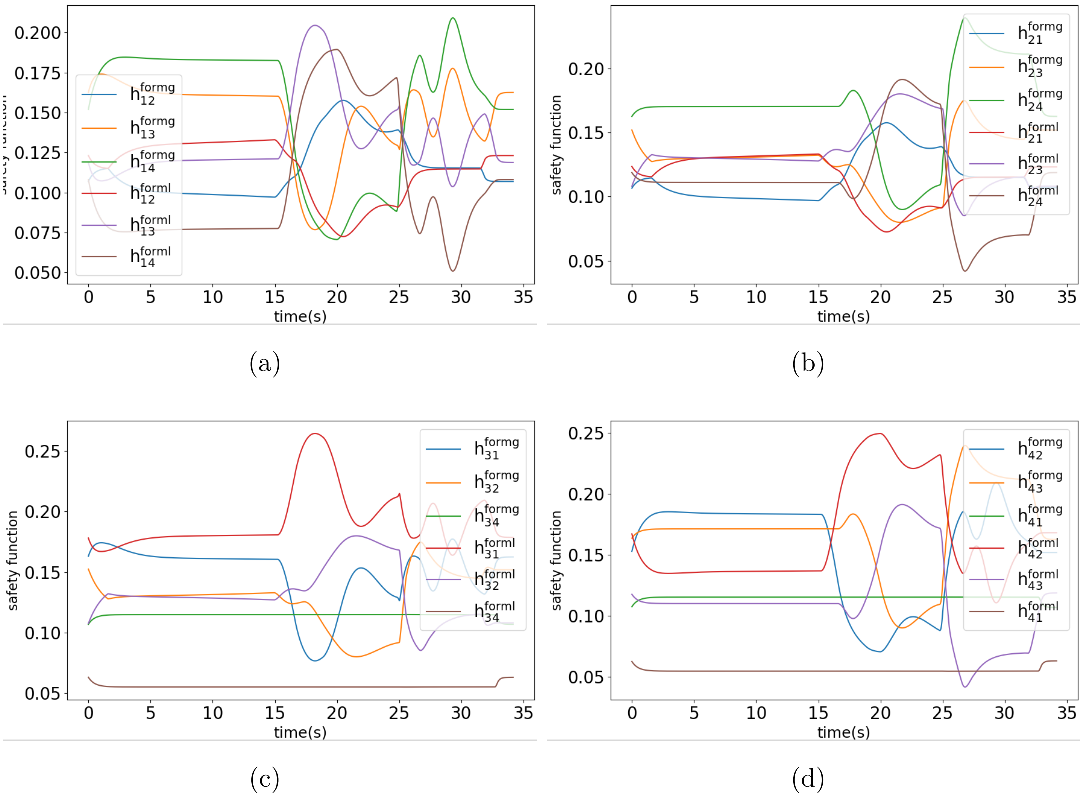
(e) t= 25s (f) t = 35s

Figure 5.6 *Snapshots of a four robot formation avoiding a dynamic obstacles and reaching its goal.*

In Figure 5.6 a four robot formation is shown avoiding an obstacle moving towards it. It can be clearly seen the control input from the QP controller enables the formation of robots to maneuver swiftly as the dynamic obstacle approaches around t = 10s to t = 20s and reach the goal.

As mentioned earlier the time plots of the safety functions for the specifications would allow one to evaluate the controllers performance in upholding them. Therefore, in the presence of dynamic obstacles the controllers performance can be evaluated by analysis of the following safety functions over time.

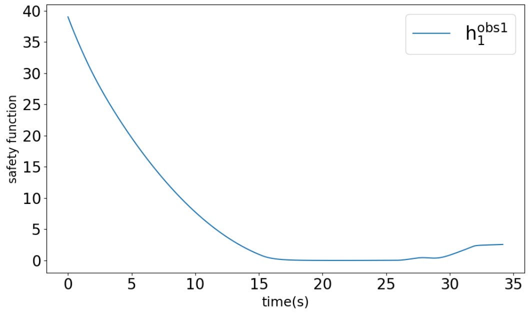
$$\begin{bmatrix} h_i^{obsj}(p_i) \\ h_{ij}^{formg}(p_i) \\ h_{ij}^{forml}(p_i) \end{bmatrix} \geq 0 \tag{5.3}$$



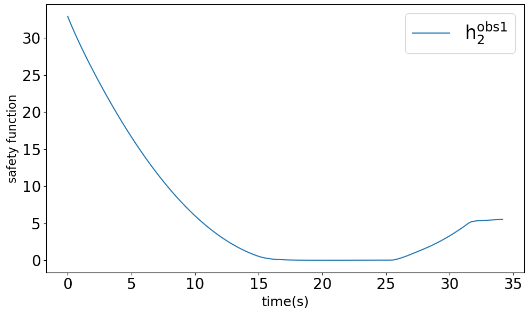
(a) (b) (c) (d)

Figure 5.7 *Plots of safety function for formation maintenance for each robot.*

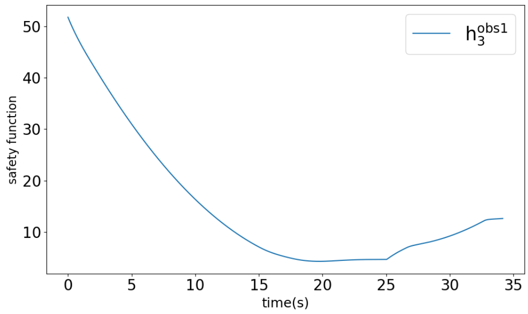
In the Figure 5.7 the safety functions related to formation maintenance stay positive over time, thus stating that the robots always stay in formation while avoiding the obstacle and navigating to their goal positions.



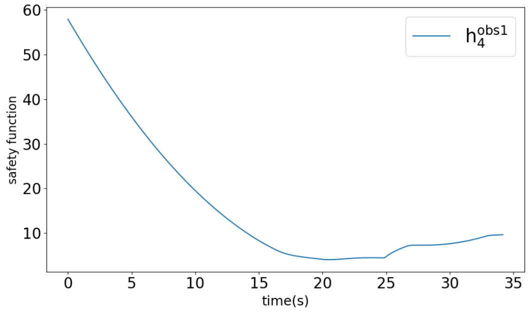
(a)



(b)



(c)



(d)

Figure 5.8 *Plots of safety function for obstacle avoidance for each robot.*

In Figure 5.8 starting at t = 15s till t = 35s the robot get near to the moving obstacle which results in the safety function getting closer to zero, but the control input given by the controller to the robots allows the maneuvering of the formation such that it's kept positive.

In Figure 5.9 a simulated result of robot formation colliding with the dynamic obstacle is shown. This result is obtained if there was no constraint of obstacle avoidance imposed on the controller.

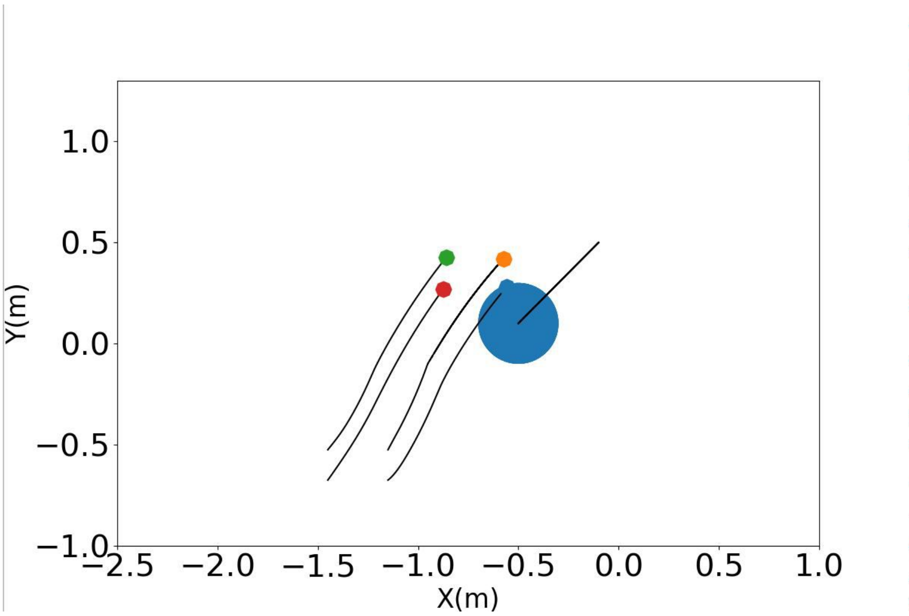


Figure 5.9 *illustration of robot behavior without specification of obstacle avoidance with respect to dynamic obstacle*

The scenario of formation of four robots avoiding the dynamic obstacle in its way and reach the goal position has been successfully evaluated in a simulated environment.

5.2 Formation of robots avoiding other formations

In this section the main focus is on performing formation avoidance between two robot formations, firstly the formation of robots are trying to reach their goal positions but are obstructed by another formation. The following sections present these scenarios of robot formations in way of each other and how the avoidance is achieved using the control algorithm developed in section 4.2.

5.2.1 Two robot formations avoiding each other

In Figure 5.11 the scenario of Two-Robot formations moving in the environment come across each other while trying to reach their respective goal positions. The Figure 5.10 shows the indented states of the robots during the start and end of the simulation.

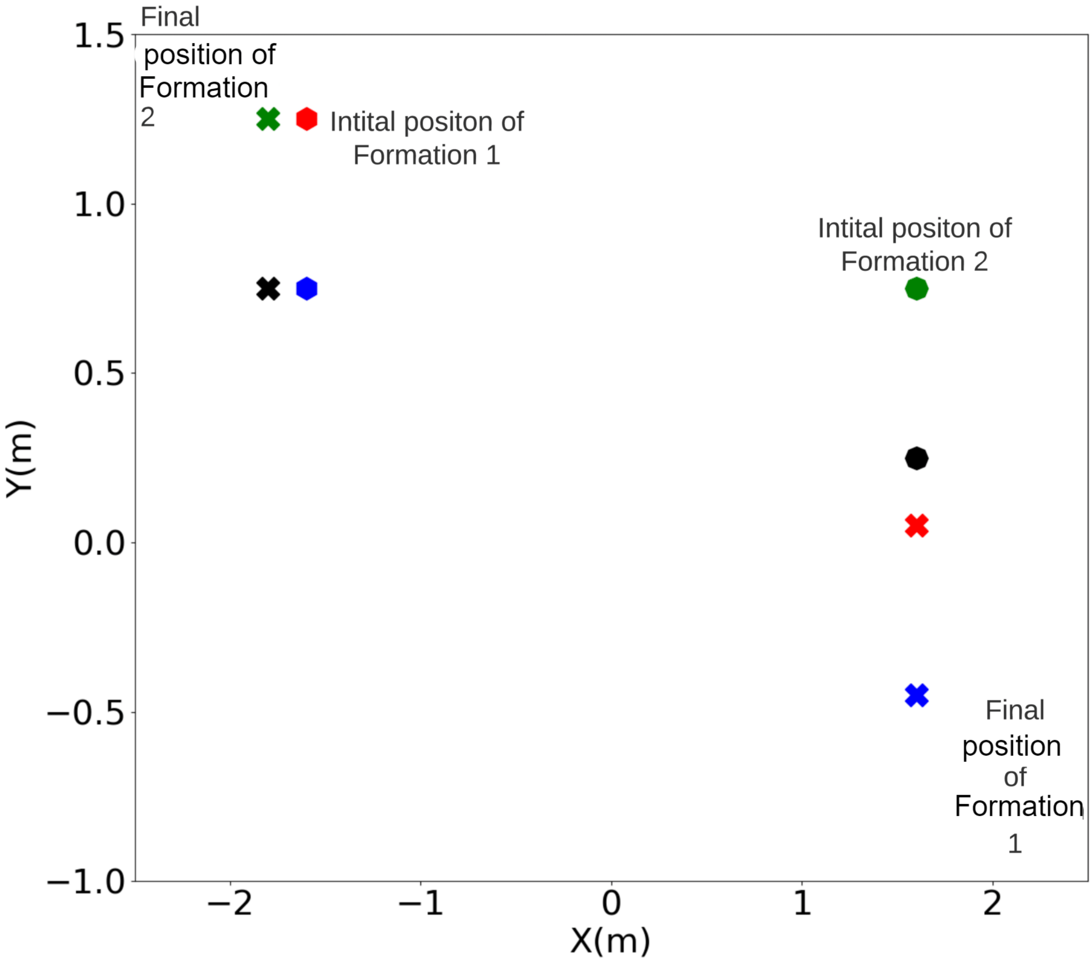


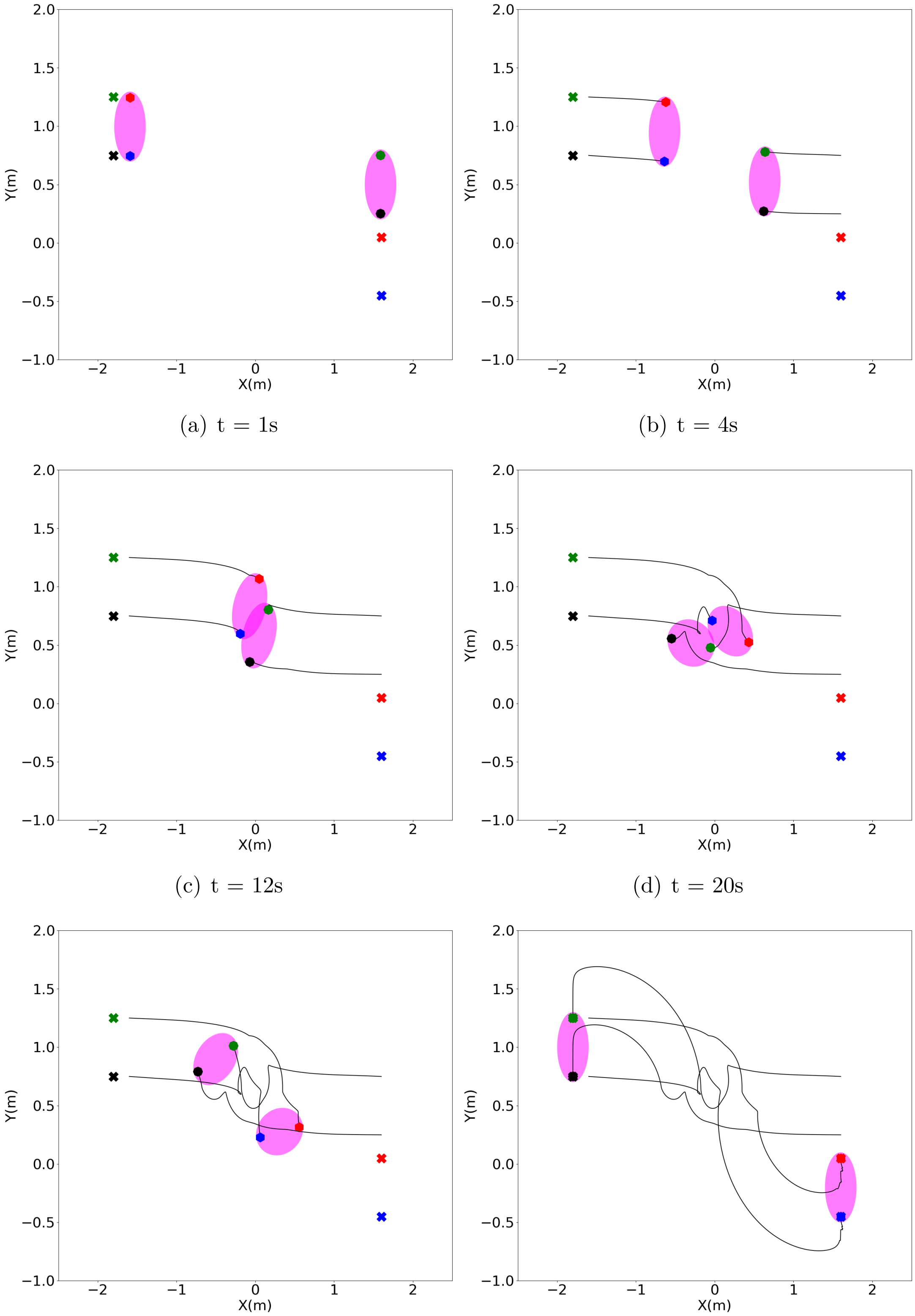
Figure 5.10 *Illustration of start and goal positions of the Two-robot formations.*

Table 5.3 *parameter setting.*

k_p	γ	L(m)	a(m)	b(m)
2	1	0.5	0.6	0.3

The adjacent weighted matrix for a two robot configuration as shown in figure 5.10 with dimensions considered in Table 5.3 can be given as:

$$\mathcal{A} = \begin{bmatrix} 0 & 0.5 \\ 0.5 & 0 \end{bmatrix} \tag{5.4}$$



(a) t = 1s

(b) t = 4s

(c) t = 12s

(d) t = 20s

(e) t = 30s

(f) t = 40s

Figure 5.11 *Snapshots of two Two-robot formations avoiding each other and reaching their goal.*

The ellipse enclosing the robot formations illustrates the region where robots of the opposite formation cannot enter, this can be seen in the Figure 5.11. The elliptical safety function (4.6) incorporated into the controller for formation avoidance allows the close maneuvering of the robot formation to avoid the other formation.

In order to successfully execute the task given to the robots must make sure the following safety functions formulated as in section 4.2 to stay positive.

$$\begin{bmatrix} h_i^{Afc}(p_i) \\ h_{ij}^{formg}(p_i) \\ h_{ij}^{forml}(p_i) \end{bmatrix} \geq 0 \tag{5.5}$$

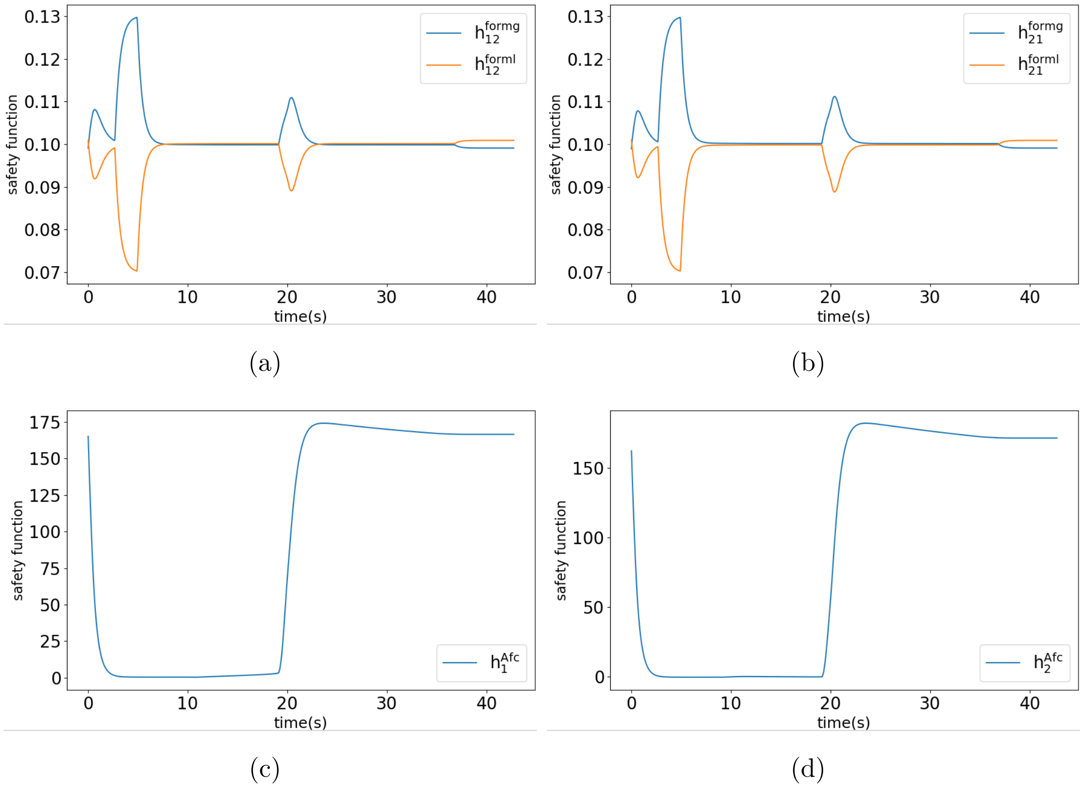


Figure 5.12 *Plots of safety functions for (a),(b) Formation maintenance and (c),(d) for obstacle avoidance for each robot in Formation 1.*

In Figure 5.12 and 5.13 the plots of safety function for both formations are shown where the controllers input to the robots enables them to keep the formation intact and keep the safety functions positive through out the simulation.

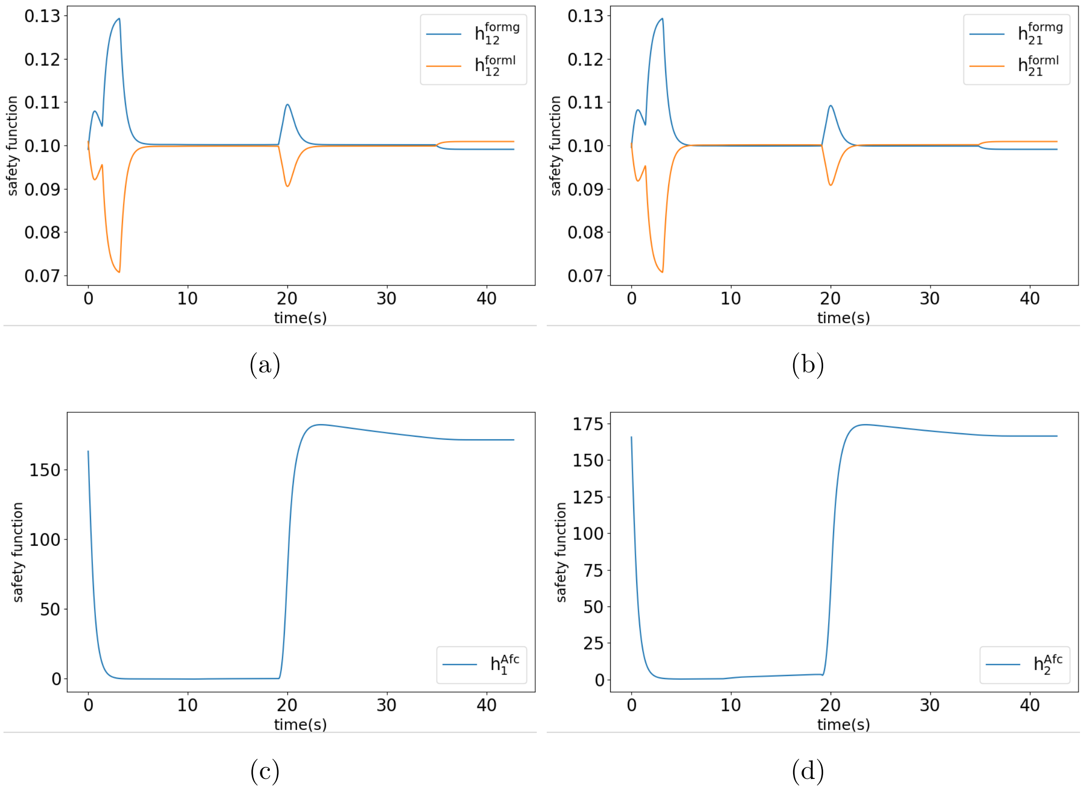


Figure 5.13 *Plots of safety functions for (a),(b) Formation maintenance and (c),(d) for obstacle avoidance for each robot in Formation 2*

At t = 20s it can be noticed that the safety function for avoiding formation come close to zero, but the controller's control input allows the robots to perform close maneuvering and avoid robots from the other formation and keep the system safe and intact. Once the robots manage to move and avoid each other they can successfully reach their goal positions and finish the task as shown in Figure 5.11.

5.2.2 Four-robot formations maneuvering close to each other

In this section two Four-Robot formations maneuver close to each other trying to reach their goal positions in a simulated environment. Figure 5.14 shows the robots initial and final configurations, where they would come across each other while navigating in their respective formations to reach goal positions.

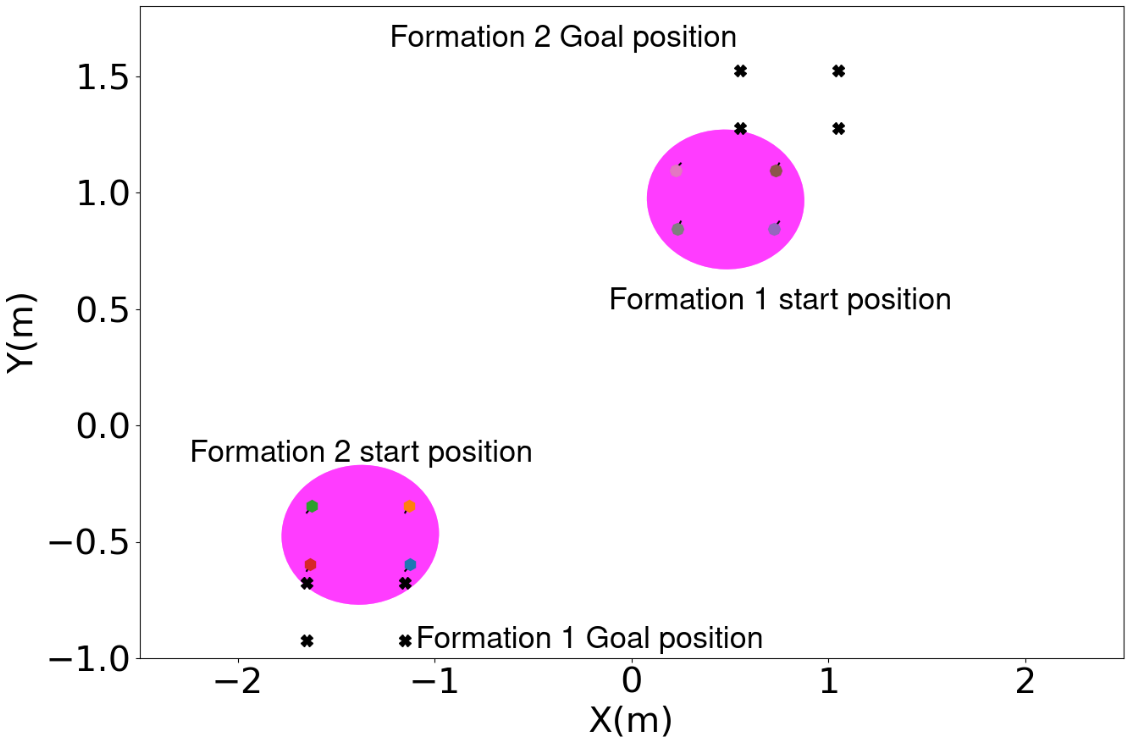


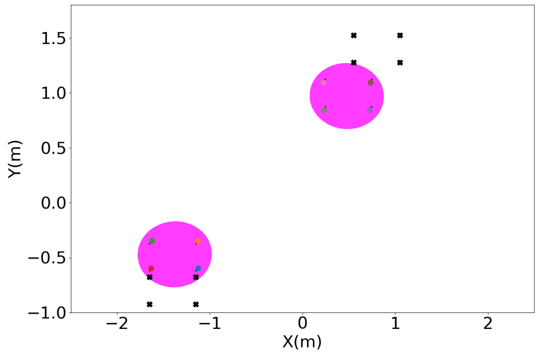
Figure 5.14 *An Illustration of start and intended goal positions of two 4-robot formations*

Table 5.4 *parameter setting.*

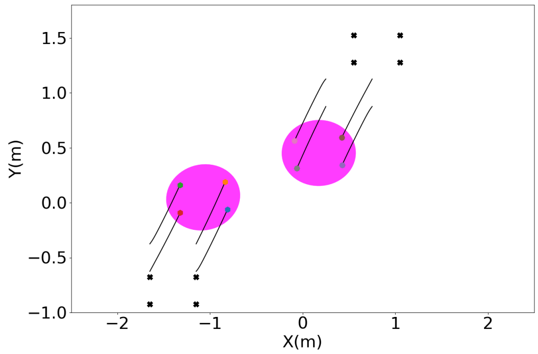
k_p	γ	$D_s(m)$	L(m)	B(m)	a(m)	b(m)
2	0.1	0.3	0.5	0.25	0.6	0.3

The parameters used in the calculation of the controller's control input are shown in Table 5.4. The weighted adjacency matrix used for both four robot formations is given in (5.1). As discussed in section 4.2 the safety function for the formation avoidance developed in this thesis constructs an elliptical barrier around the robot formation such that all the robots fall under this barrier, defining the safety around all robots.

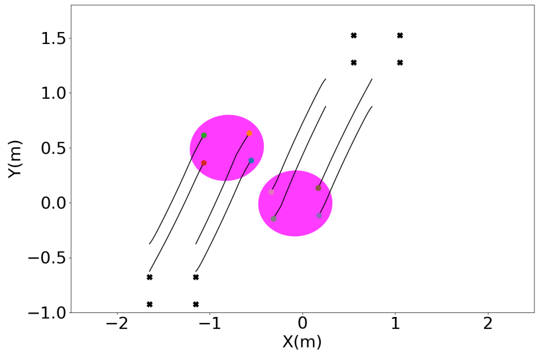
This would compel the controller's QP optimizer to provide control input from the safe set values which would allow the formations to avoid each other and return to their nominal behaviour as shown in Figure 5.15.



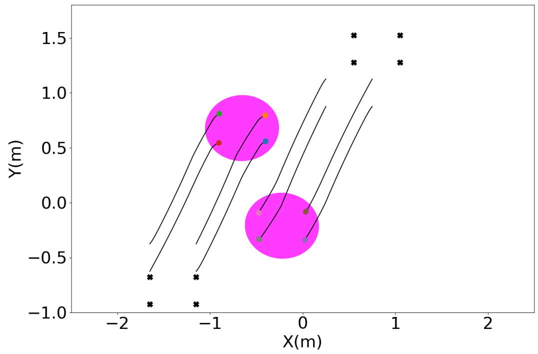
(a) t = 1s



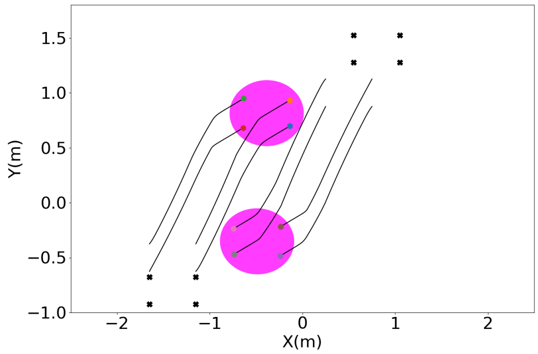
(b) t = 4s



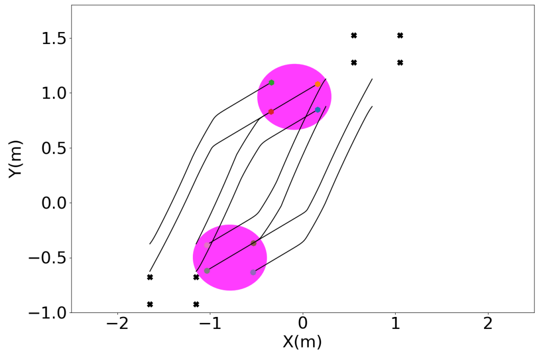
(c) t = 25s



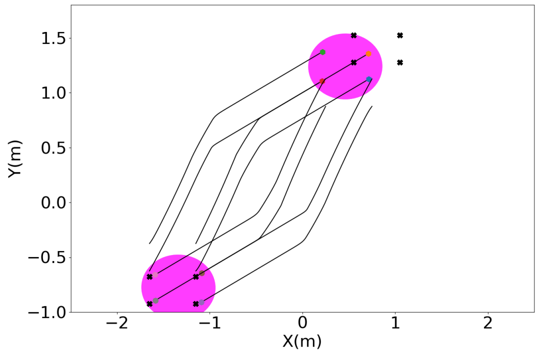
(d) t = 45s



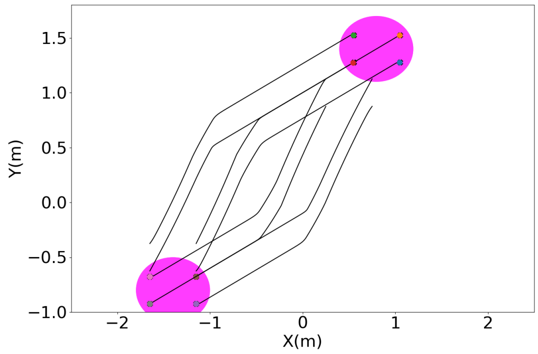
(e) t = 65s



(f) t = 90s



(g) t = 110s



(h) t = 120s

Figure 5.15 *Snapshots of two Four-robot formations avoiding each other and reaching their goal.*

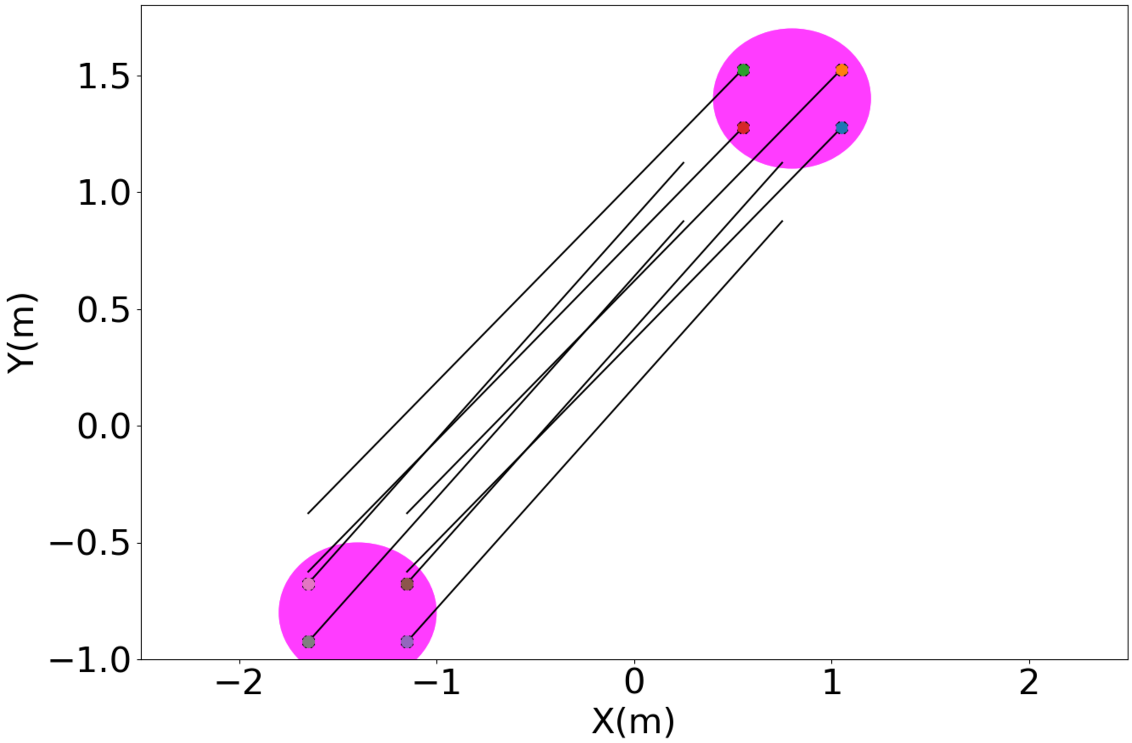


Figure 5.16 *Snapshot of formations going to their goal without avoiding each other under nominal control.*

In the absence of the specification incorporated into the controller for avoiding other robot formations, the behaviour of the robots can be shown in Figure 5.16 where the robot formations are bound to collide with one another.

The following safety function values for formation maintenance and avoiding other formations need to remain positive through out the simulation,

$$\begin{bmatrix} h_i^{Afc}(p_i) \\ h_{ij}^{formg}(p_i) \\ h_{ij}^{forml}(p_i) \end{bmatrix} \geq 0 \tag{5.6}$$

This would allow one to analyse the controller's capability to uphold the specifications defined in chapter 3 when multi-robot formations encounter one another during their task execution.

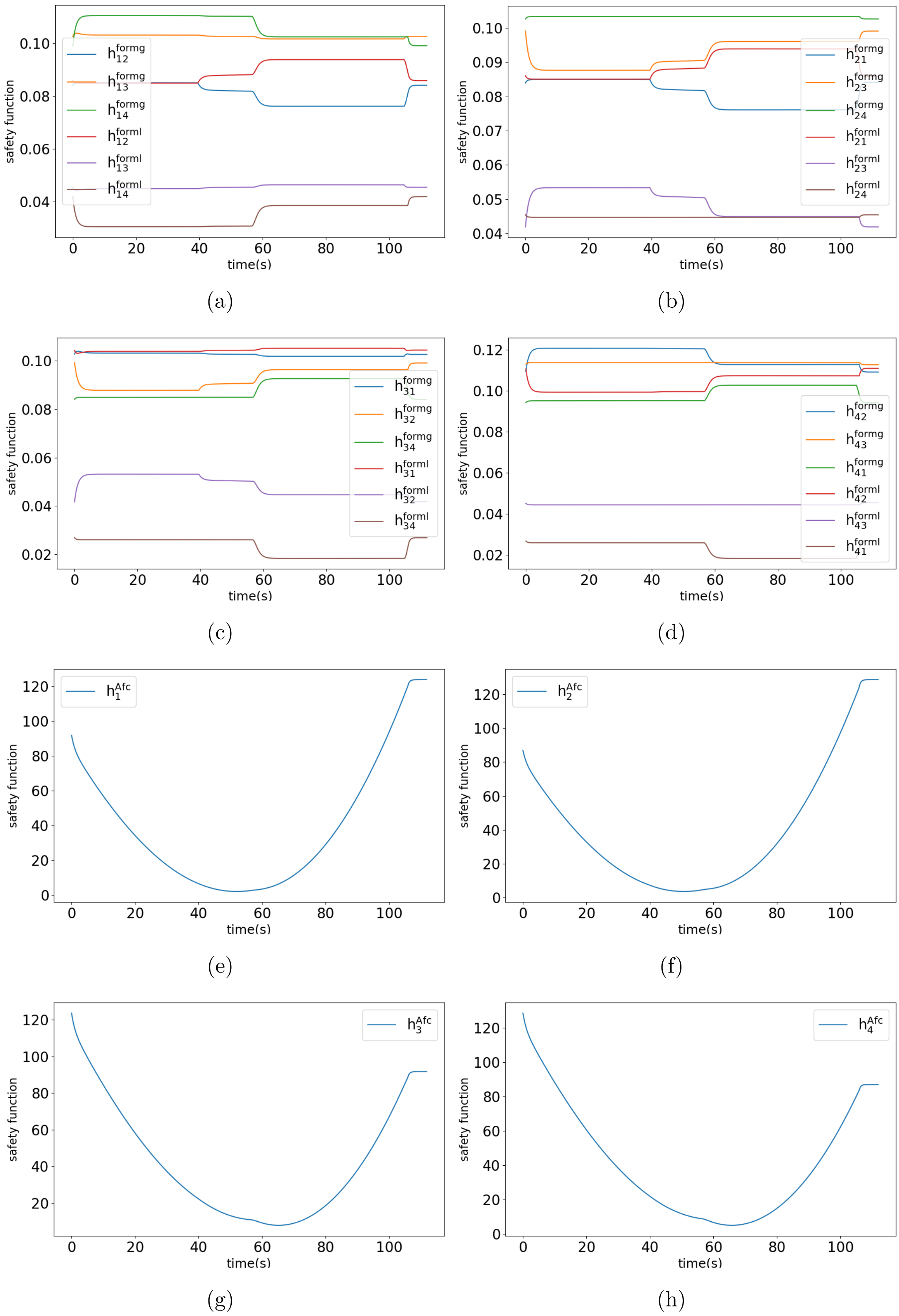


Figure 5.17 *Plots of safety function for formation maintenance and avoiding other formation for four robots in Formation 1.*

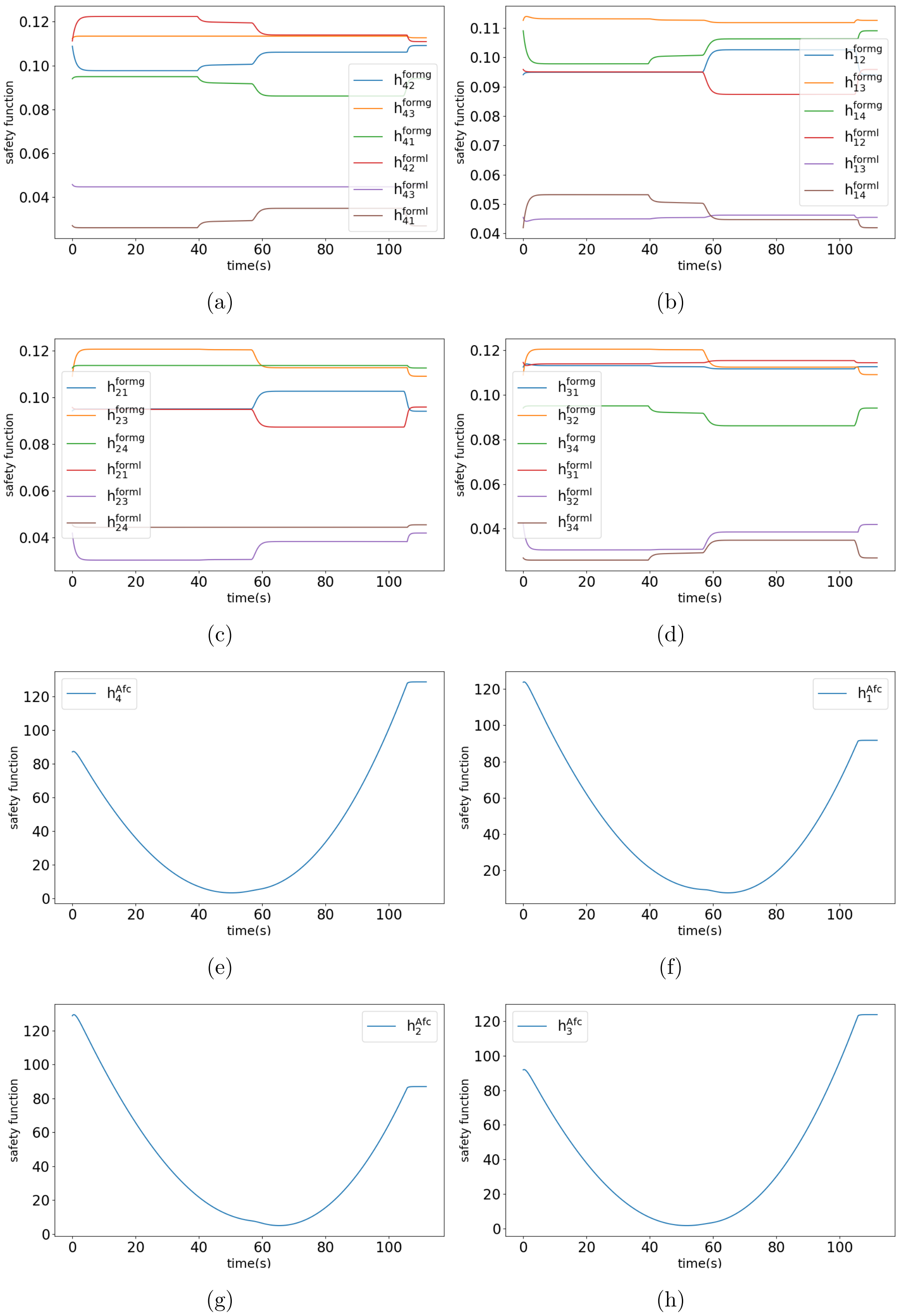


Figure 5.18 *Plots of safety function for formation maintenance and avoiding other formation for four robots in Formation 2.*

In Figure 5.17 and 5.18 it can clearly be seen that the safety functions are kept positive through the experiment for both the formations clearly indicating the controller's control input satisfied the specification of formation maintenance while trying to avoid collisions with other formations. The safety functions for avoiding other formations in Figure 5.17 and 5.17 can be seen getting close to zero about t = 60s as the formations get close to one another, but stay positive as the controllers control input allows them to avoid collisions.

Thus the scenario of four robot formations avoiding each other was successfully evaluated in a simulated environment.

Effect of γ on robots behaviour

The extended class $\mathcal{K}$ function γ used in the formulation of the safe set discussed in section 2.2 affects the behaviour of the robots,i.e it controls how fast the robot system can approach the boundary of the safe set. In Figure 5.19 the robot moves towards the obstacle through the control input given by the controller under different γ values.

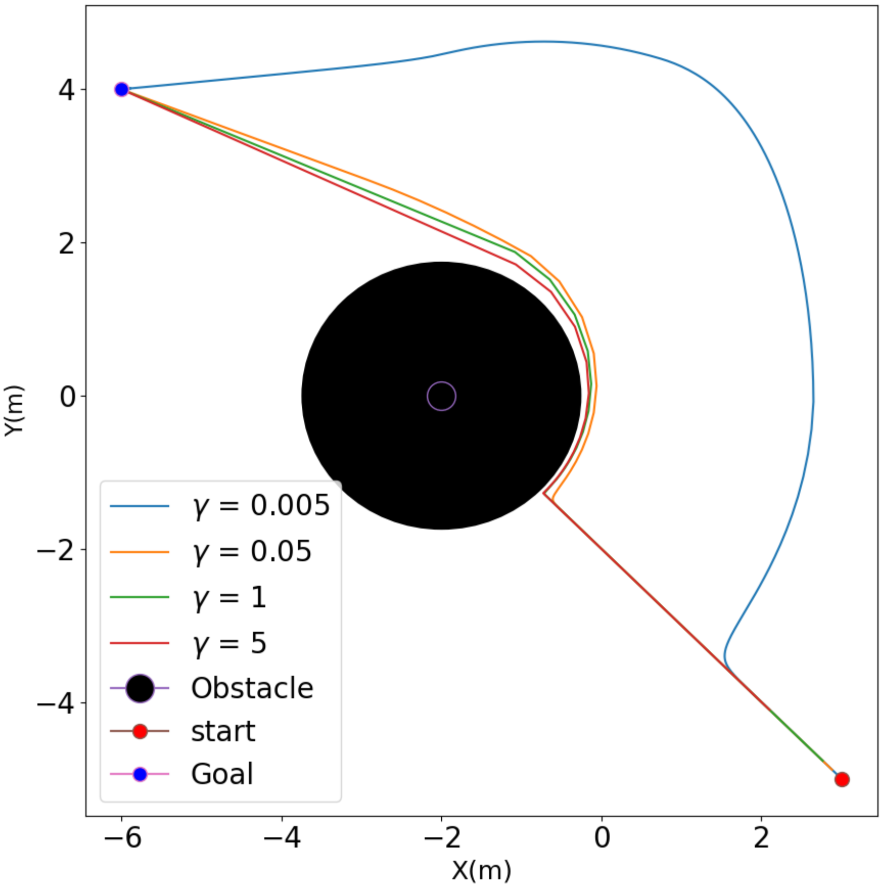


Figure 5.19 *Plot of robots motion under different γ values.*

It demonstrates that one can fine-tune the values of γ to allow the robots to react early to the presence of the obstacles. This would enable the robots to keep themselves safe in presence of dynamic obstacles sensed far away and react quickly accordingly.

In this chapter simulated results of multi robot formations going through various scenarios has been presented and evaluated successfully.

6 Demonstration and Evaluation in real robotic platform

In this chapter, the control algorithm developed in this thesis is deployed and tested on physical systems, namely mobile robots. The following sections will discuss the apparatus and software used in testing scenarios for evaluating the performance of the algorithm.

6.1 Experimental Setup

The main experimental setup to implement the algorithm developed includes:

- TurtleBot3 burger robots.
- Zed2 ceiling camera for localisation of the robots moving on the ground.
- Remote Desktop with Intel i7 processor,NVIDIA Quadro P1000 and Ubuntu 20.04 LTS operating system.
- ROS noetic framework for communication between robots

Figure 6.1 shows the layout of the system architecture used to deploy the algorithm on mobile robots.

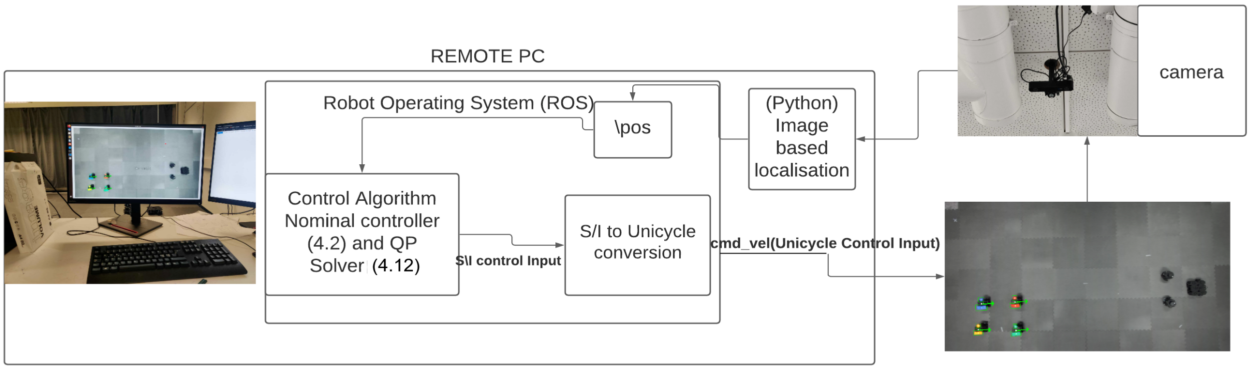


Figure 6.1 *Overview of the experimental setup for the execution of control algorithm on physical systems.*

6.1.1 Robot Operating System

Robot Operating System (ROS) is one of the most popular open-source operating systems for robots. ROS enables low-level device control, communication messages between different processes, package management and other useful utilities.

To implement the algorithm developed in this thesis ROS noetic is used to establish the communication between the multi-robot systems. The algorithm can be deployed on each robot such that the control strategy is run as a distributed system where each robot only requires set variables to be communicated within each other as designed in the communication graph and execute cooperative control tasks smoothly.

6.1.2 TurtleBots

TurtleBots are one of the most popular open source mobile robot platforms used for research and education purposes. These mobile robots are ROS-based robots which allows the communication between the robots to be established using ROS nodes.

We establish a ssh connection between the turtlebots and the master remote PC, the table 6.1 shows the respective turtlebot IP addresses to their color used to set up the multi robot system.

Table 6.1 *IP addresses of the turtlebots.*

Robots	IP Address
Turtlebot3-Red	192.168.0.201
Turtlebot3-Blue	192.168.0.137
Turtlebot3-Green	192.168.0.138
Turtlebot3-Orange	192.168.0.223

6.1.3 Image based Localisation

To perform various cooperative control algorithms there is a necessity to acquire the positions of the robots involved. Therefore to implement the algorithm developed on real robots, a ceiling camera is used to visualize the robots on the ground. This ceiling camera has all the robots in its field of view. There are several ways to implement image-based localization to obtain positions and orientations of different robots in the camera's view. In this thesis, a colour based object detection algorithm is used to localize the robots, where each robot is assigned a specific colour.

This feature extraction algorithm enables the detection of a specific colour and gives good accuracy in localization data. Both the position and orientation data acquired from the Localization algorithm is published to ROS topics such that this information can be used by both the robots and users.

6.1.4 Implementation in ROS Framework

This section discusses in detail how the above-described entities have been combined to implement the developed control algorithm on the mobile robots. Figure 6.1 shows the layout of the whole experimental setup working in cohesion right from the image data acquired from the ceiling camera which is fed to the Remote PC.

The Remote PC takes in the image data and performs feature extraction to obtain the localization data of the respective robots. Then the Remote PC publishes the localization data into the ROS network with respective namespace for the robots to be distinguished.

The control algorithm to perform the prescribed tasks is then run on the remote PC(master) and ROS enables the communication between the master and the turtlebots. Thus specific control inputs can be sent to individual robots from the controller running and the obtained results are discussed in the following sections.

6.2 Single integrator to Unicycle Dynamics

The Unicycle model of a turtlebot (non holonomic mobile robot) with state $[x\ y\ \theta]^T$ is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{6.1}$$

where $(x, y) \in \mathbb{R}^2$ represents the 2D position of the robot, θ is the orientation, v and w are the linear and angular velocity of the robot.

The kinematic model of the robot considered for the development of control algorithm is a single integrator model as mentioned in chapter 3. But the objective of developing the controller is to deploy on real physical robots which are differential drive robots.

These differential drive robots cannot execute single integrator commands due to the non-holonomic constrain imposed on them .i.e, differential drive robots cannot drive sideways. Hence the single integrator control inputs need to be converted to unicycle control commands.

In [33] the authors present a method to convert single integrator input to unicycle input by opting for near-identity diffeomorphism (NID) between the desired single integrator model and the unicycle model.

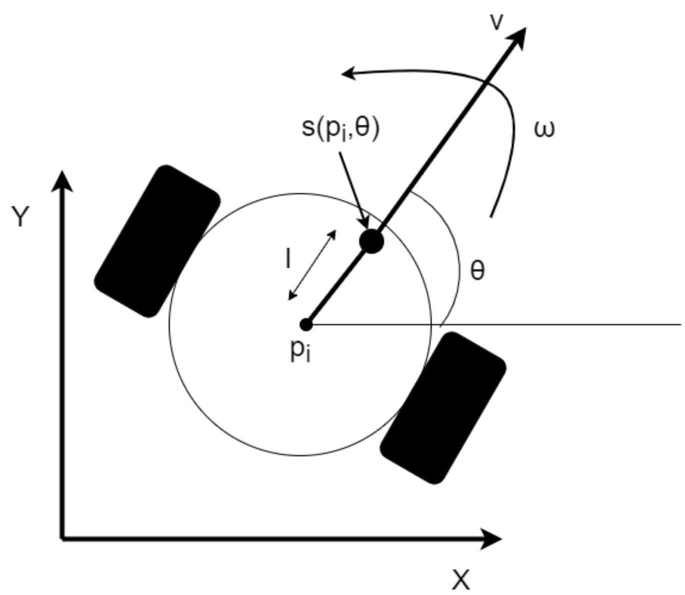


Figure 6.2 *A pictorial illustration of a differential drive robot, with variables used in diffeomorphism superimposed. Center of robot is given by p_i and look ahead point on the robot at distance l given by $s(p_i, \theta)$. The unicycle inputs linear and angular velocity given by v and ω*

Therefore from [33] the single integrator input generated by the control algorithm can be mapped to corresponding unicycle input for a non holonomic robot i as shown in Figure 6.2 is given by,

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\frac{1}{l}\sin(\theta) & \frac{1}{l}\cos(\theta) \end{bmatrix} \dot{s}(p_i, \theta) \tag{6.2}$$

where $s(p_i, \theta)$ is the look ahead point on the robot i and $l \in \mathbb{R}$ is the distance of the look ahead point from the robot centre.

By using (6.2) the single integrator output from the controller can be transformed to linear and angular velocity inputs and fed to the mobile robots.

6.3 Formation Control and reaching goal

In this section a formation of four turtlebots is made to move from a start position to goal position and orient itself while maintaining formation.

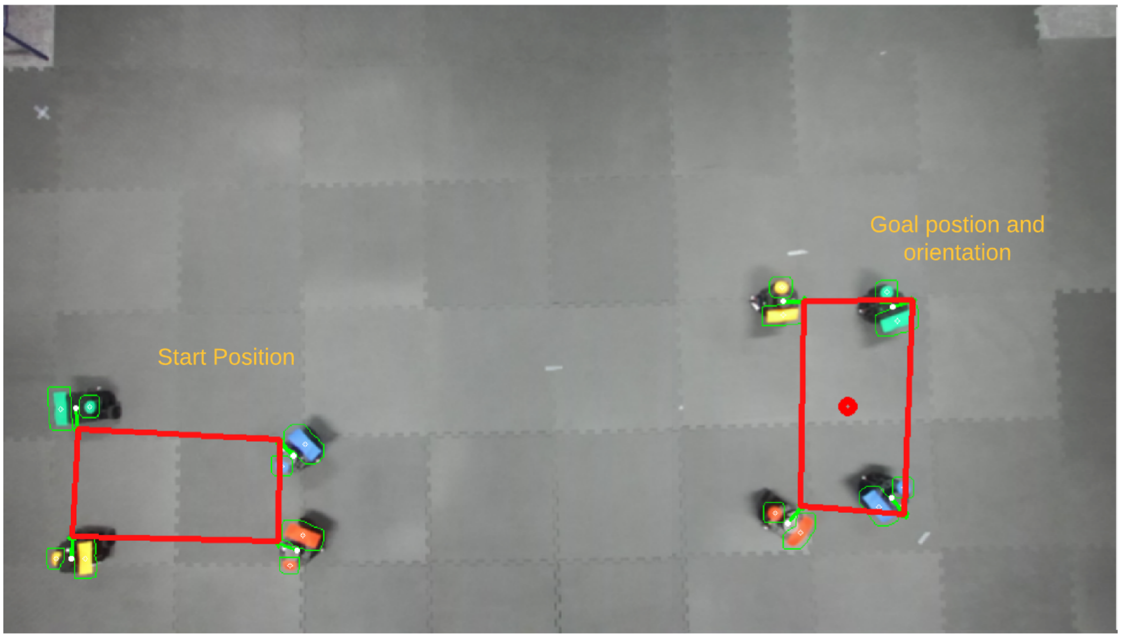


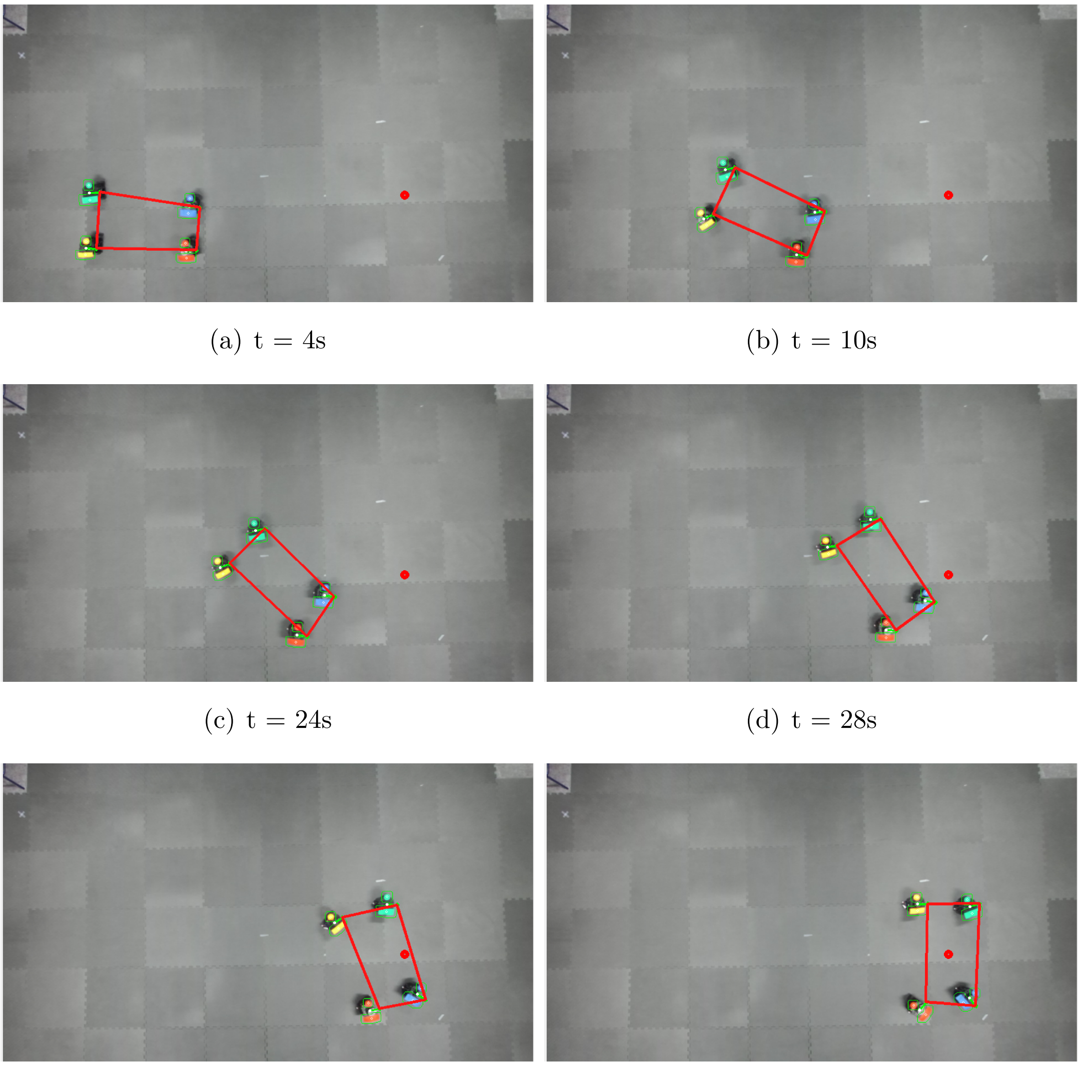
Figure 6.3 *Illustration of the intended scenario of formation of four turtlebots reaching goal*

Table 6.2 *parameter settings for four robot formation to reach goal.*

k_p	γ	$D_s(m)$	L(m)	B(m)
2	1	0.3	0.5	0.25

In Figure 6.3 the desired outcome is presented where the multi robot formation translates to the assigned goal position.

The parameters involved in the calculated of the control input are given in Table 5.1. The weighted adjacency matrix for the four robot formation is given in (5.1) allowing the controller to impose the requirements for desired formation shape on the physical robots.



(a) t = 4s (b) t = 10s

(c) t = 24s (d) t = 28s

(e) t = 30s (f) t = 35s

Figure 6.4 *Snapshots of formation of turtlebots navigation to a goal position.*

In Figure 6.4 it can be clearly noticed that the formation of turtlebots maintain their formation and navigate to the assigned goal.

The performance of the control algorithm on this scenario can be evaluated by analysis of safety functions for formation maintenance being positive given as,

$$\begin{bmatrix} h_{ij}^{formg}(p_i) \\ h_{ij}^{forml}(p_i) \end{bmatrix} \geq 0 \tag{6.3}$$

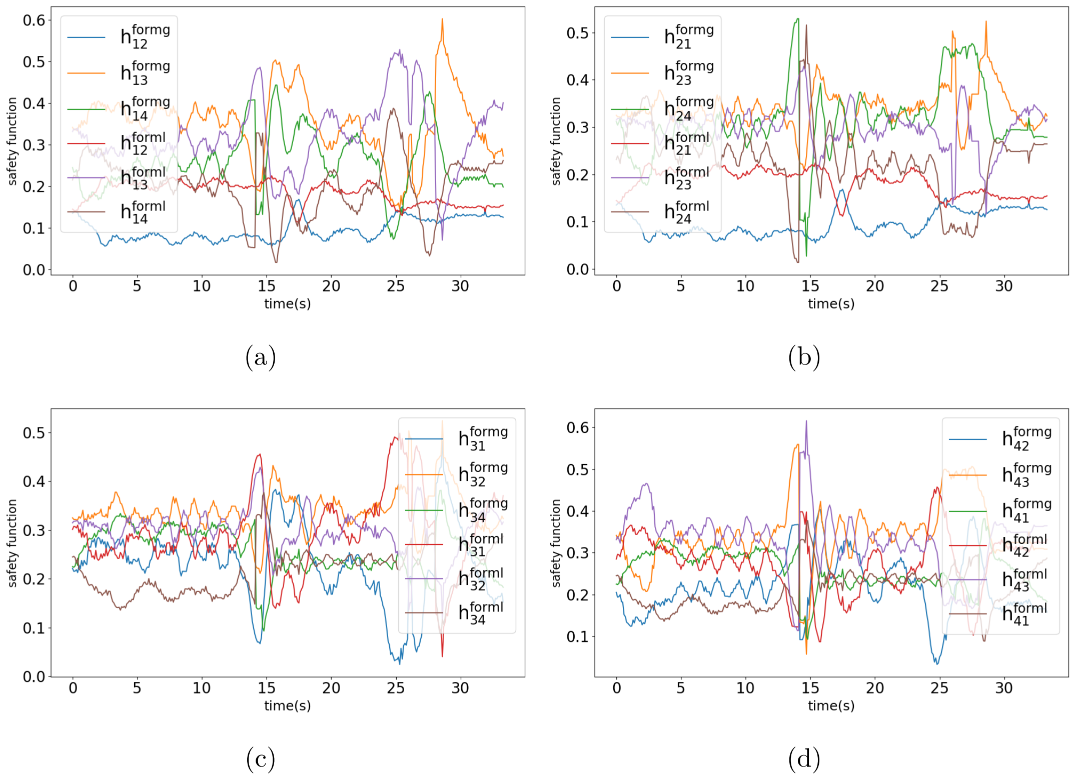


Figure 6.5 *Plots of safety functions for formation maintenance for four robots.*

The Figure 6.5 shows how the plots of the safety functions defined for formation maintenance are kept positive by the controller's control input given to the robots.

This experimental scenario Let us us evaluate the control algorithms capability of translating and orienting the formation of actual mobile robots to desired position while maintaining formation.

6.4 Formation Obstacle avoidance

In this section the scenario of formation of four turtlebots moving through a passage between two obstacles and reaching the goal position is evaluated. This scenario would allows to test the capability of the robots to avoid the obstacles while keeping the formation intact as defined in the specifications of the controller and reach the goal position smoothly.

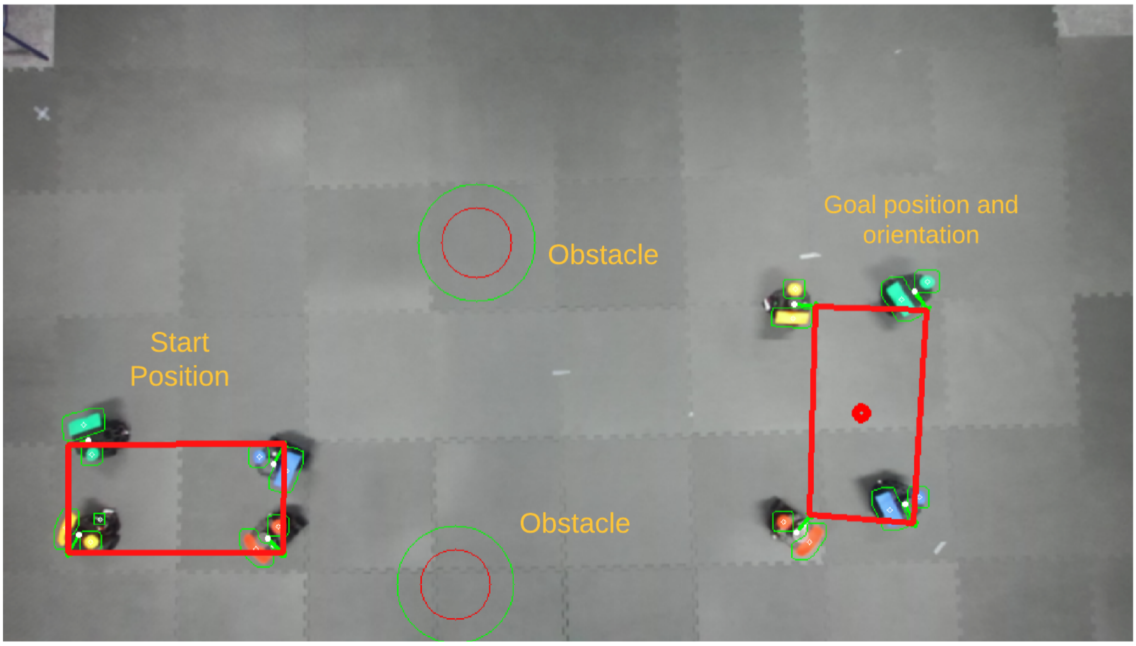
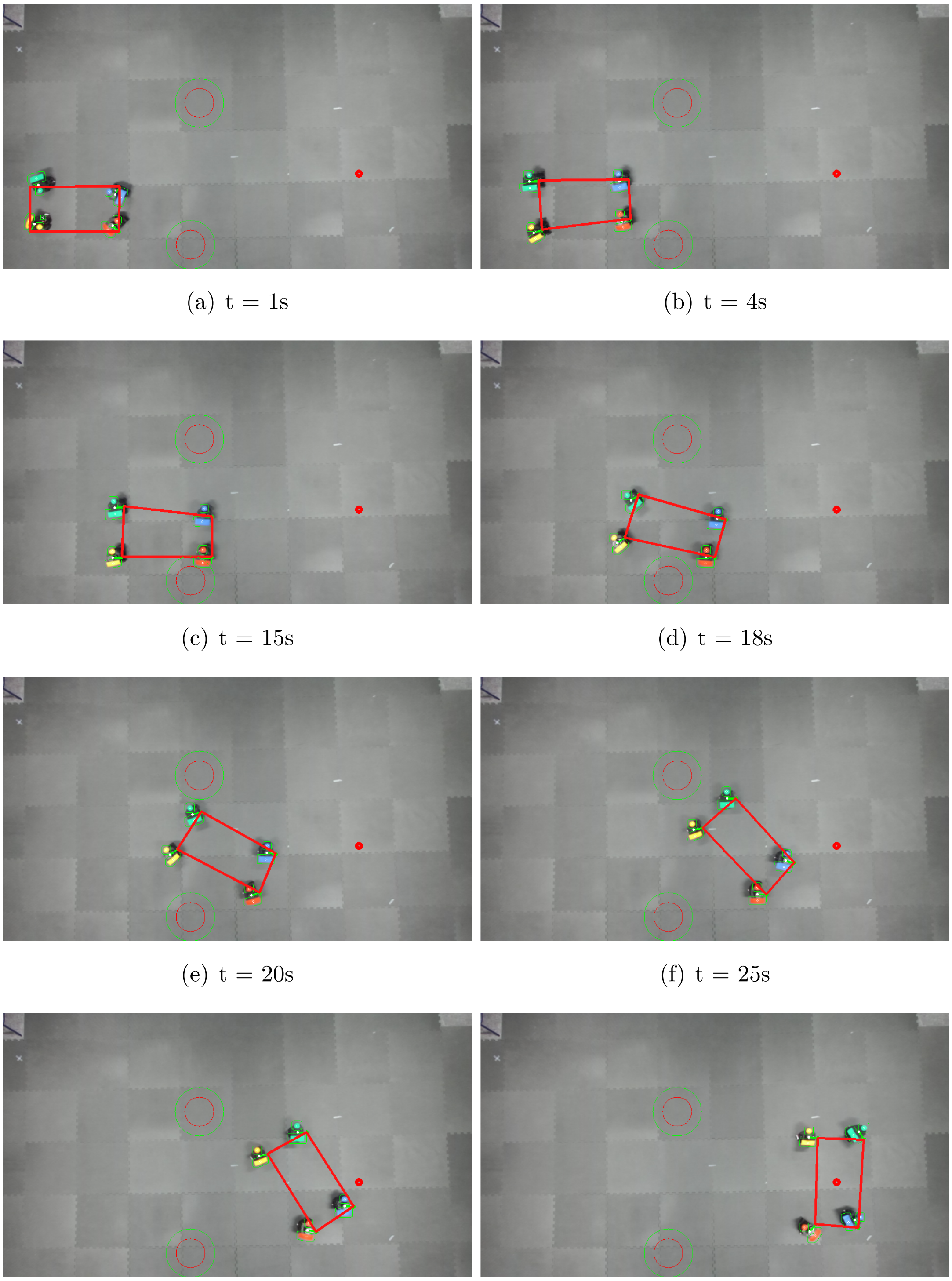


Figure 6.6 *Illustration of robot formation avoiding obstacle to reach its goal*

Table 6.3 *parameter settings for four robot formation to avoid obstacles and reach goal.*

k_p	γ	$D_s(m)$	L(m)	B(m)
2	1	0.3	0.5	0.25

The weighted adjacency matrix considered for the four robot formation is given in (5.1).



(a) t = 1s (b) t = 4s

(c) t = 15s (d) t = 18s

(e) t = 20s (f) t = 25s

(g) t = 30s (h) t = 35s

Figure 6.7 *Snapshots of robot formation avoiding obstacles in its way to reach goal position. The red circle representing the obstacle, green circle is the safe area outside the obstacle that robot can enter while maneuvering the formation to goal.*

The following safety functions need to be positive in order to evaluate the controllers capability of navigating the formation to desired goal while avoiding the obstacles.

$$\begin{bmatrix} h_i^{obsj}(p_i) \\ h_{ij}^{formg}(p_i) \\ h_{ij}^{forml}(p_i) \end{bmatrix} \geq 0 \tag{6.4}$$

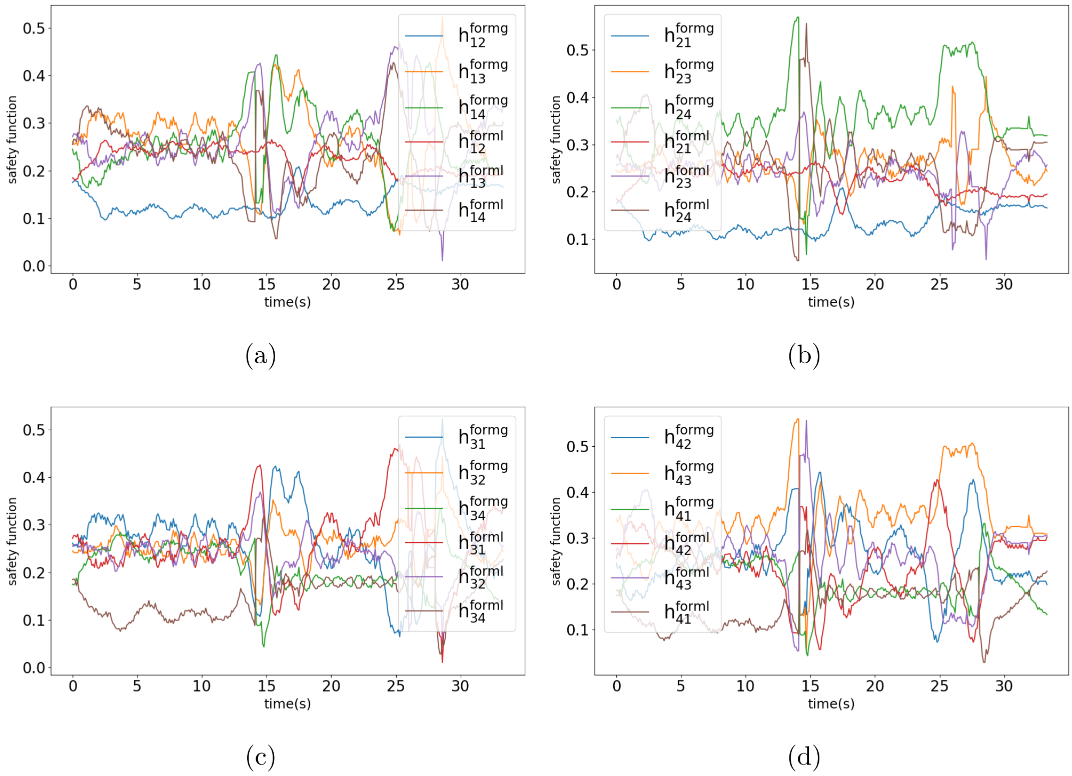


Figure 6.8 *Plots of safety function of four robots for formation maintenance.*

In Figure 6.8 the safety functions responsible for formation maintenance stay positive throughout the experiment clearly showing that the controller was able to satisfy the specification for formation maintenance.

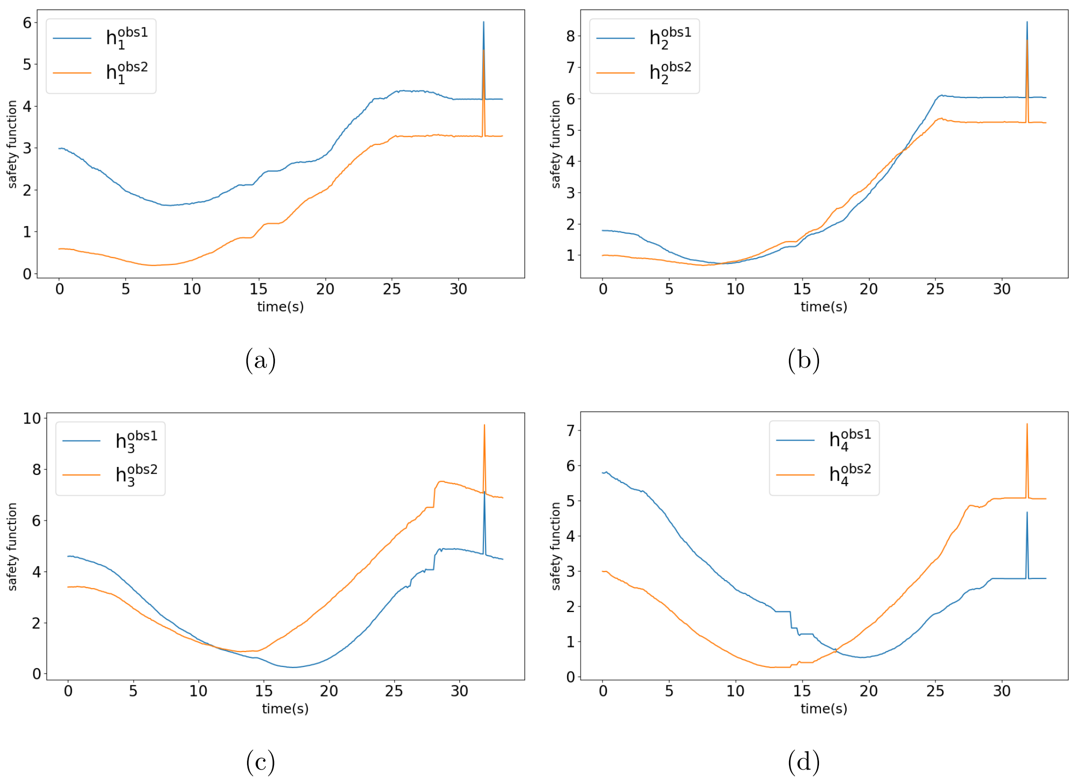


Figure 6.9 *Plots of safety function of four robots for avoiding obstacles.*

In Figure 6.9 the plots of the safety function defined for obstacle avoidance are shown, jitters in the plots can be noticed about t = 25s which could be the noise in robots localization that needs to be accounted for while implementing the algorithm on real robot systems.

The results provided in this sections allows us to verify the control algorithm developed was successfully able to navigate a four robot formation of mobile robots through the environment avoiding obstacles.

6.5 Two-robot formations maneuvering close to each other

In this section two 2-robot formations are made to come across each other, this would trigger the specification incorporated for avoiding other robot formations. This allows to perform analysis on how the barrier certificates with a elliptical safety function works on physical systems.

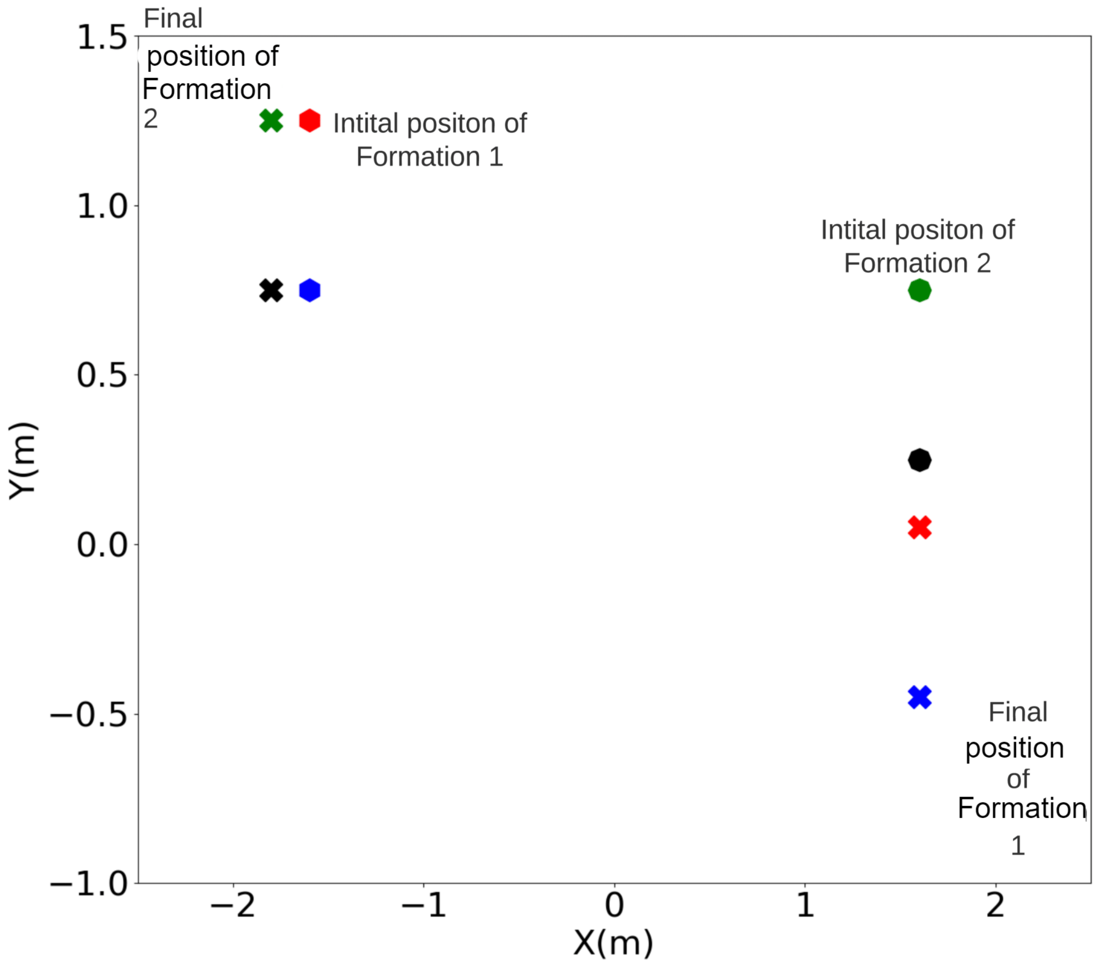
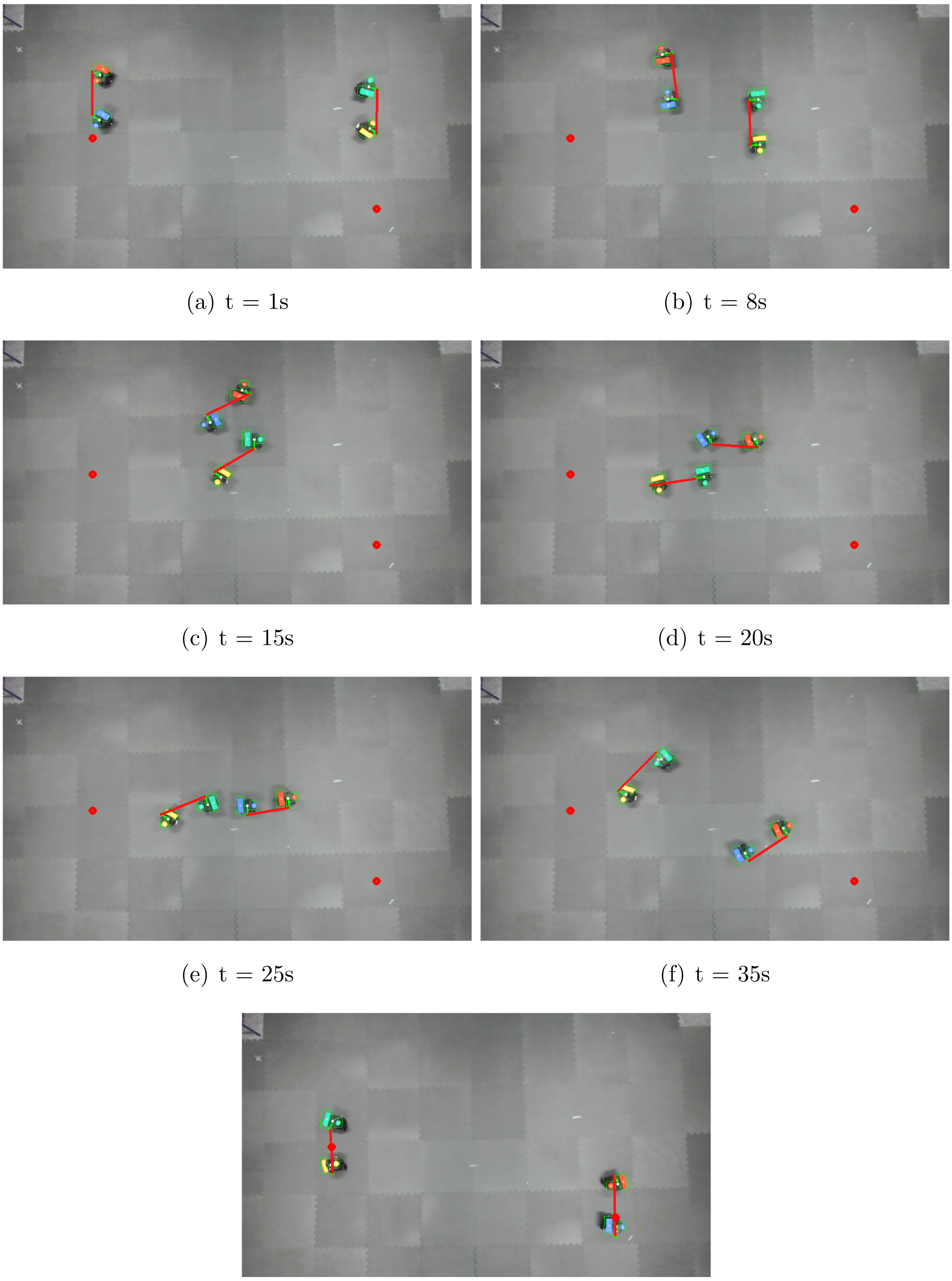


Figure 6.10 *Illustration of two robot formations start positions and their respective goal positions.*

Table 6.4 *parameter setting.*

k_p	γ	L(m)	a(m)	b(m)
2	1	0.5	0.6	0.3

The parameters related to the calculation of the control input are shown in Table 6.4 and the weighted adjacency matrix for the formation is given in 5.4.



(a) t = 1s (b) t = 8s

(c) t = 15s (d) t = 20s

(e) t = 25s (f) t = 35s

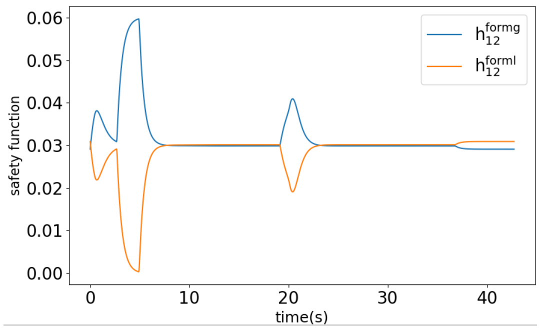
(g) t = 40s

Figure 6.11 *Snapshots of two-robot formations avoiding each other.*

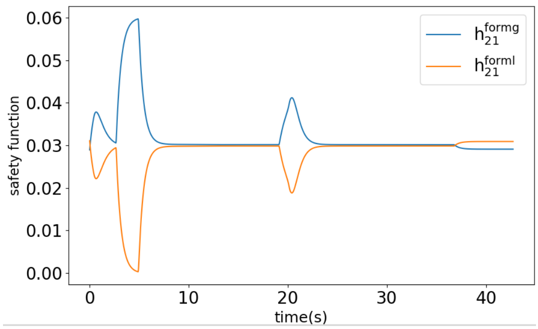
The Figure 6.11 shows two-Robot formations moving opposite to each other trying to reach their goal positions as described in the Figure 6.10, about t = 15s it can be seen that they maneuver close to one another and successfully reach their respective goal positions.

The following safety functions for avoiding formations and maintaining formation need to be positive during the experiment to evaluate the controllers performance,

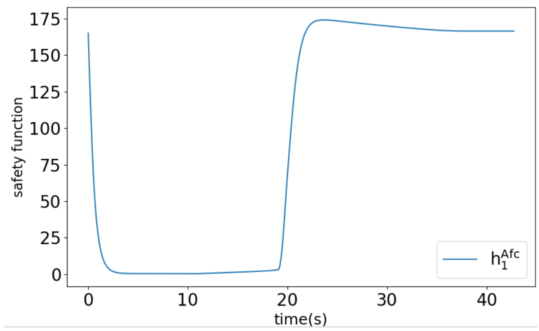
$$\begin{bmatrix} h_i^{Afc}(p_i) \\ h_{ij}^{formg}(p_i) \\ h_{ij}^{forml}(p_i) \end{bmatrix} \geq 0 \tag{6.5}$$



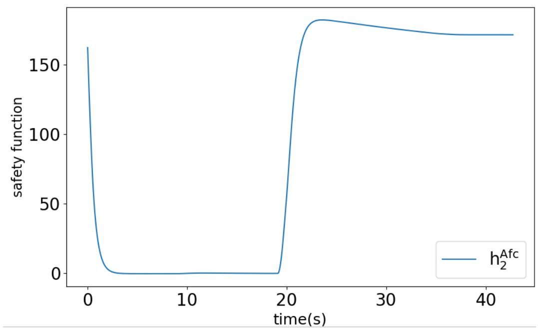
(a)



(b)



(c)



(d)

Figure 6.12 *Plots of safety function of formation maintenance (a),(b) and avoiding other formations (c),(d) for robots in formation 1.*

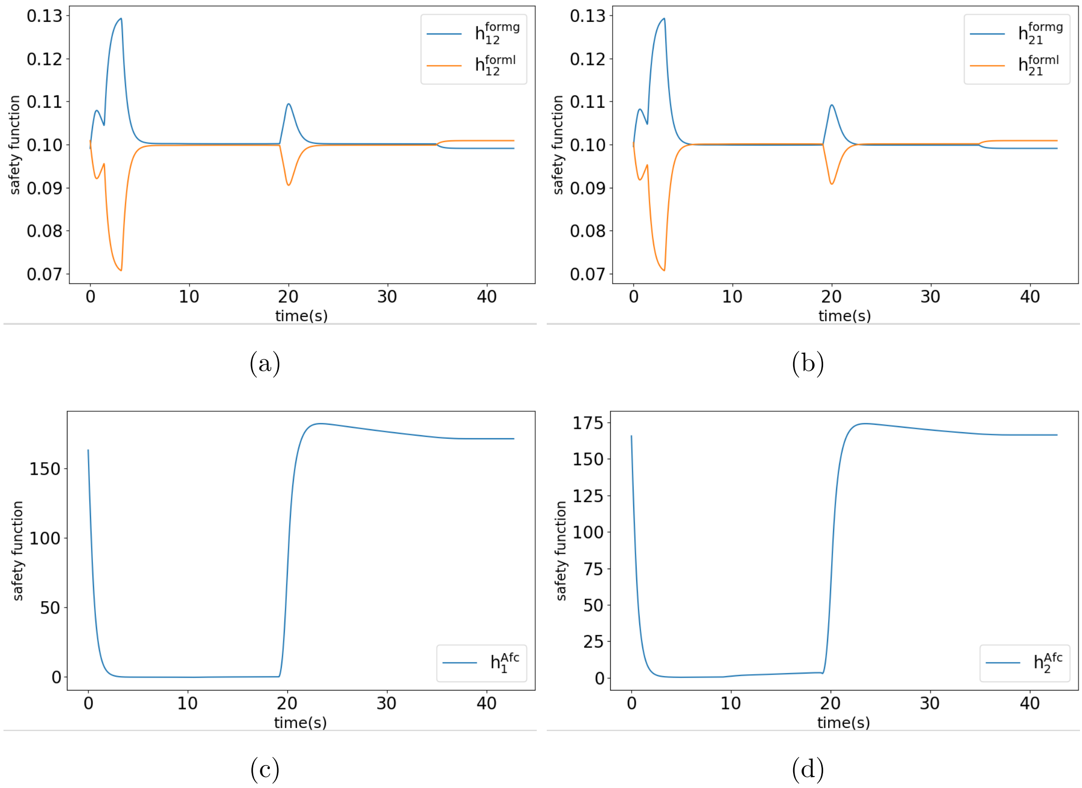


Figure 6.13 *Plots of safety function of formation maintenance (a),(b) and avoiding other formations (c),(d) for robots in formation 2.*

In figure 6.13 plots of the safety functions of the four robots responsible for avoiding other formations is shown, about t = 15s the robot formations get closer and try to maneuver to avoid each other, the safety function gets close to zero but still remains positive. The controller's control input was able to satisfy the specifications for avoiding formations.

In this chapter the main objective was to deploy the control algorithm on real mobile robots, test and verify the algorithm execution and make sure it replicates the simulated scenarios as closely as possible. Hence the implementation of the control algorithm on physical systems was carried out successfully.

7 Conclusions and Future Work

A real-time distributed control algorithm for controlling a group of mobile robots to function cooperatively with multiple objectives is presented in this thesis.

The principle of control barrier certificates was used to model the control laws where the main aim of the control algorithm was to generate a safe control input for the team of mobile robots.

The algorithm developed incorporated the specifications of formation maintenance, obstacle avoidance and avoiding other formations into the controller, allowing smooth task execution for multi-robot systems in dynamic environment.

An effective elliptical control barrier certificate was designed to enable close maneuvering of multi-robot systems to avoid each other with reduced communication cost.

A thorough discussion of the algorithm's working on simulated environment has been presented where multi-robot formations have been tested for different scenarios one would come across in a warehouse.

The control algorithm was deployed on physical systems namely turtlebots and its performance was analysed. The algorithm was able to achieve the desired outcomes but there was some noise recorded while analysing the time plots of safety function. The results obtained through simulations and implementation on physical robots have shown that the control algorithm's effectiveness in being able to handle the cooperative control problems in real world warehouse environment.

7.1 Future Work

There is a wide variety of scope for future research into this thesis work,

- The model developed in this thesis considers a Single integrator model for the robots, whereas this can be extended to a double integrator model and check for more robustness in the controller performance.

- Solve the localization problem when feature-based(image-based) localization cannot be implemented. A further study into Multi-robot SLAM would enable the prospect of implementing the control algorithm for different use cases.

- Another prospect would be building a custom robot for the use case of warehouse automation, the mobile robots used in the project are more of testing robots for the verification of the controller behaviour for set constraints. But a custom robot could be built for a set warehouses environment and be able to execute tasks such as product placement, surveillance etc.

References

[1] https://www.aboutamazon.com/news/innovation-at-amazon/the-story-behind-amazons-next-generation-robot. Accessed: 2021-10-13.

[2] S. Ahmadzadeh and M. Ghanavati. "Navigation of Mobile Robot Using the PSO Particle Swarm Optimization". In: *Journal of Academic and Applied Studies* 2 (2012).

[3] J. Alonso-Mora et al. "Distributed multi-robot formation control in dynamic environments". en. In: *Auton. Robots* 43.5 (2019), pp. 1079–1100.

[4] A. D. Ames, J. W. Grizzle, and P. Tabuada. "Control barrier function based quadratic programs with application to adaptive cruise control". In: *53rd IEEE Conference on Decision and Control*. IEEE, 2014.

[5] A. D. Ames et al. "Control barrier function based quadratic programs for safety critical systems". In: *IEEE Trans. Automat. Contr.* 62.8 (2017), pp. 3861–3876.

[6] A. D. Ames et al. "Control barrier functions: Theory and applications". In: (2019). eprint: 1903.11199.

[7] B. D. O. Anderson et al. "Maintaining a directed, triangular formation of mobile autonomous agents". In: *Commun. Inf. Syst.* 11.1 (2011), pp. 1–16.

[8] B. D. Anderson et al. "Rigid graph control architectures for autonomous formations". In: *IEEE Control Systems Magazine* 28.6 (2008), pp. 48–63. DOI: 10.1109/MCS.2008.929280.

[9] *Automated warehouse*. en. https://locusrobotics.com/features/autonomous-robots/. Accessed: 2021-10-13. Aug. 2016.

[10] T Balch and R. C. Arkin. "Behavior-based formation control for multirobot teams". In: *IEEE Trans. Rob. Autom.* 14.6 (1998), pp. 926–939.

[11] *Bolt - IAM robotics*. en. https://www.iamrobotics.com/products/bolt/. Accessed: 2021-10-13. Mar. 2021.

[12] J. Borenstein and Y. Koren. "Real-time obstacle avoidance for fast mobile robots". In: *IEEE Transactions on Systems, Man, and Cybernetics* 19.5 (1989), pp. 1179–1187. DOI: 10.1109/21.44033.

[13] U. Borrmann et al. "Control Barrier Certificates for Safe Swarm Behavior". In: *IFAC-PapersOnLine* 48.27 (2015). Analysis and Design of Hybrid Systems ADHS, pp. 68–73. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2015.11.154. URL: https://www.sciencedirect.com/science/article/pii/S240589631502412X.

[14] H. Dan et al. “Control barrier function-based persistent coverage with performance guarantee and application to object search scenario”. In: *2020 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2020.

[15] R. D’Andrea. “Guest Editorial: A Revolution in the Warehouse: A Retrospective on Kiva Systems and the Grand Challenges Ahead”. In: *IEEE Transactions on Automation Science and Engineering* 9.4 (2012), pp. 638–639. DOI: 10.1109/TASE.2012.2214676.

[16] J. P. Desai, J. P. Ostrowski, and V Kumar. “Modeling and control of formations of nonholonomic mobile robots”. In: *IEEE Trans. Rob. Autom.* 17.6 (2001), pp. 905–908.

[17] J. Fax and R. Murray. “Information flow and cooperative control of vehicle formations”. In: *IEEE Transactions on Automatic Control* 49.9 (2004), pp. 1465–1476. DOI: 10.1109/TAC.2004.834433.

[18] J. Fottner et al. “Autonomous Systems in Intralogistics - State of the Art and Future Research Challenges”. In: *Logist. Res.* 14 (2021), p. 2.

[19] D Fox, W Burgard, and S Thrun. “The dynamic window approach to collision avoidance“”. In: *IEEE Robot. Autom. Mag* 4 (1997-03), pp. 23–33,

[20] G. Fragapane et al. “Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics”. en. In: *Ann. Oper. Res.* (2020).

[21] J. E. Graver, B Servatius, and H Servatius. *Combinatorial rigidity.* American Mathematical Society, 1993.

[22] J Guo et al. *Adaptive control schemes for mobile robot formations with triangularized structures“.*

[23] A. Hegde and D. Ghose. “Multi-UAV Collaborative Transportation of Payloads With Obstacle Avoidance”. In: *IEEE Control Systems Letters* 6 (2022), pp. 926–931. DOI: 10.1109/LCSYS.2021.3087339.

[24] *Industrial & commercial autonomous robots for large-scale solutions.* en. https://waypointrobotics.com/commercial-industrial-manufacturing-robots/. Accessed: 2021-10-14. Sept. 2017.

[25] O. Khatib. “Real-time obstacle avoidance for manipulators and mobile robots”. In: *Autonomous Robot Vehicles.* New York, NY: Springer New York, 1986, pp. 396–404.

[26] D. Koung et al. “Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system”. In: *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2020, pp. 92–97. DOI: 10.1109/ICARCV50220.2020.9305426.

[27] Y. Liu and R. Bucknall. “A survey of formation control and motion planning of multiple unmanned vehicles”. en. In: *Robotica* 36.7 (2018), pp. 1019–1047.

[28] M. Maghenem et al. “Consensus-based formation control of nonholonomic robots using a strict lyapunov function”. en. In: *IFAC-PapersOnLine* 50.1 (2017), pp. 2439–2444.

[29] G. Notomista and M. Egerstedt. “Persistification of Robotic Tasks”. In: *IEEE Transactions on Control Systems Technology* 29.2 (2021), pp. 756–767. DOI: 10.1109/TCST.2020.2978913.

[30] A. Pandey. “Mobile Robot Navigation and Obstacle Avoidance Techniques: A Review”. In: *Int. robot. autom. j.* 2.3 (2017).

[31] S. Prajna and A. Jadbabaie. “Safety Verification of Hybrid Systems Using Barrier Certificates”. In: *Hybrid Systems: Computation and Control*. Ed. by R. Alur and G. J. Pappas. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 477–492. ISBN: 978-3-540-24743-2.

[32] H. Rezaee and F. Abdollahi. “A Decentralized Cooperative Control Scheme With Obstacle Avoidance for a Team of Mobile Robots”. In: *IEEE Transactions on Industrial Electronics* 61.1 (2014), pp. 347–354. DOI: 10.1109/TIE.2013.2245612.

[33] *Robotarium*. https://www.robotarium.gatech.edu/examples/formation-control. Accessed: 2021-10-25.

[34] S. Singh et al. “Impact of COVID-19 on logistics systems and disruptions in food supply chain”. In: *International Journal of Production Research* 59.7 (2021), pp. 1993–2008. DOI: 10.1080/00207543.2020.1792000. eprint: https://doi.org/10.1080/00207543.2020.1792000. URL: https://doi.org/10.1080/00207543.2020.1792000.

[35] Z. Sun et al. “Leader-follower formation control of underactuated surface vehicles based on sliding mode control and parameter estimation”. In: *ISA Transactions* 72 (2018), pp. 15–24. ISSN: 0019-0578. DOI: https://doi.org/10.1016/j.isatra.2017.11.008. URL: https://www.sciencedirect.com/science/article/pii/S001905781730616X.

[36] P. Wieland and F. Allgöwer. "CONSTRUCTIVE SAFETY USING CONTROL BARRIER FUNCTIONS". In: *IFAC Proceedings Volumes* 40.12 (2007). 7th IFAC Symposium on Nonlinear Control Systems, pp. 462–467. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20070822-3-ZA-2920.00076. URL: https://www.sciencedirect.com/science/article/pii/S1474667016355690.

[37] J. Zhang, J. Yan, and P. Zhang. "Multi-UAV Formation Control Based on a Novel Back-Stepping Approach". In: *IEEE Transactions on Vehicular Technology* 69.3 (2020), pp. 2437–2448. DOI: 10.1109/TVT.2020.2964847.

[38] M. Zhang et al. "Dynamic artificial potential field based multi-robot formation control". In: *2010 IEEE Instrumentation Measurement Technology Conference Proceedings*. 2010, pp. 1530–1534. DOI: 10.1109/IMTC.2010.5488238.

[39] Y. Zhang, D.-W. Gong, and J.-H. Zhang. "Robot path planning in uncertain environment using multi-objective particle swarm optimization". en. In: *Neurocomputing* 103 (2013), pp. 172–185.

Appendices

A.1 Derivation of safety function for Obstacle avoidance

$$\dot{h}_i^{obsj}(p_i) = \frac{\partial h_i^{obsj}}{\partial p_i} u_i + \frac{\partial h_i^{obsj}}{\partial p_{obsj}} \dot{p}_{obsj} \quad \geq -\gamma h_i^{obsj}(p_i) \; \forall i \in \mathcal{V}_k, \quad \forall j \in \{1, \ldots, o\} \tag{1}$$

$$\frac{\partial h_i^{obsj}}{\partial p_i} = 2(p_i - p_{obsj}) \tag{2}$$

$$\frac{\partial h_i^{obsj}}{\partial p_{obsj}} = -2(p_i - p_{obsj}) \tag{3}$$

$$\Big[-2(p_i - p_{obsj})\Big] \Big[u_i\Big] \leq \gamma h_i^{obsj}(p_i) + \frac{\partial h_i^{obsj}}{\partial p_{obsj}} \dot{p}_{obsj} \tag{4}$$

A.2 Derivation of safety function for formation maintenance

Greater case:

$$\dot{h}_{formgij}(p_i) = \frac{\partial h_{ij}^{formg}}{\partial p_i} u_i + \frac{\partial h_{ij}^{formg}}{\partial p_j} \dot{p}_j + \quad \geq -\gamma h_{ij}^{formg}(p_i) \quad \forall (i,j) \in \mathcal{E}_k \tag{5}$$

$$\frac{\partial h_{ij}^{formg}}{\partial p_i} = 2(p_i - p_j) \tag{6}$$

$$\frac{\partial h_{ij}^{formg}}{\partial p_j} = -2(p_i - p_j) \tag{7}$$

$$\Big[-2(p_i - p_j)\Big] \Big[u_i\Big] \leq \gamma h_{ij}^{formg}(p_i) + \frac{\partial h_{ij}^{formg}}{\partial p_j} \dot{p}_j \tag{8}$$

Lesser Case:

$$\dot{h}_{formlij}(p_i) = \frac{\partial h_{ij}^{forml}}{\partial p_i} u_i + \frac{\partial h_{ij}^{forml}}{\partial p_j} \dot{p}_j \quad \geq -\gamma h_{ij}^{forml}(p_i) \quad \forall (i,j) \in \mathcal{E}_k \tag{9}$$

$$\frac{\partial h_{ij}^{forml}}{\partial p_i} = -2(p_i - p_j) \tag{10}$$

$$\frac{\partial h_{ij}^{forml}}{\partial p_j} = 2(p_i - p_j) \tag{11}$$

$$\Big[2(p_i - p_j)\Big] \Big[u_i\Big] \leq \gamma h_{ij}^{forml}(p_i) + \frac{\partial h_{ij}^{forml}}{\partial p_j} \dot{p}_j \tag{12}$$

A.3 Derivation of safety function for avoiding other formations

$$\dot{h}_i^{Afc}(p_i) = \frac{\partial h_i^{Afc}}{\partial p_{ix}} u_x + \frac{\partial h_i^{Afc}}{\partial p_{iy}} u_y + \frac{\partial h_i^{Afc}}{\partial x_m^c} \dot{x}_c + \frac{\partial h_i^{Afc}}{\partial y_m^c} \dot{y}_c + \frac{\partial h_i^{Afc}}{\partial \alpha} \dot{\alpha} \\ \geq -\gamma h_i^{Afc}(p_i) \quad \forall i \in \mathcal{V}^k,\ m \neq k \tag{13}$$

$$\frac{\partial h_i^{Afc}}{\partial p_{ix}} = 2\left(p_{ix} - x_m^c\right)\left(\frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}\right) + 2\left(p_{iy} - y_m^c\right)\cos\alpha\sin\alpha\left(\frac{1}{a^2} - \frac{1}{b^2}\right) \tag{14}$$

$$\frac{\partial h_i^{Afc}}{\partial p_{iy}} = 2(p_{iy} - y_m^c)\left(\frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}\right) + 2\left(p_{ix} - x_m^c\right)\cos\alpha\sin\alpha\left(\frac{1}{a^2} - \frac{1}{b^2}\right) \tag{15}$$

$$\frac{\partial h_i^{Afc}}{\partial x_m^c} = -1\left(2\left(p_{ix} - x_m^c\right)\left(\frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}\right) + 2\left(p_{iy} - y_m^c\right)\cos\alpha\sin\alpha\left(\frac{1}{a^2} - \frac{1}{b^2}\right)\right) \tag{16}$$

$$\frac{\partial h_i^{Afc}}{\partial y_m^c} = -1\left(2(p_{iy} - y_m^c)\left(\frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}\right) + 2\left(p_{ix} - x_m^c\cos\alpha\sin\alpha\right)\left(\frac{1}{a^2} - \frac{1}{b^2}\right)\right) \tag{17}$$

$$\frac{\partial h_i^{Afc}}{\partial \alpha} = (p_{ix} - x_m^c)^2\left(\frac{2}{a^2}\cos\alpha(-\sin\alpha) + \frac{2}{b^2}\sin\alpha\cos\alpha\right) + (p_{iy} - y_m^c)^2\left(\frac{2}{a^2}\sin\alpha\cos\alpha - \right. \\ \left. \frac{2}{b^2}\sin\alpha\cos\alpha\right) + 2(p_{ix} - x_m^c)(p_{iy} - y_m^c)\left(\frac{1}{a^2} - \frac{1}{b^2}\right)(\cos^2\alpha - \sin^2\alpha) \tag{18}$$

let r $= 2\left(p_{ix} - x_m^c\right)\left(\frac{\cos^2\alpha}{a^2} + \frac{\sin^2\alpha}{b^2}\right) + 2\left(p_{iy} - y_m^c\right)\left(\frac{1}{a^2} - \frac{1}{b^2}\right)$, p $= 2(p_{iy} - y_m^c)\left(\frac{\sin^2\alpha}{a^2} + \frac{\cos^2\alpha}{b^2}\right) + 2\left(p_{ix} - x_m^c\right)\left(\frac{1}{a^2} - \frac{1}{b^2}\right)$

$$\begin{bmatrix} -r & -p \end{bmatrix}\begin{bmatrix} u_x \\ u_y \end{bmatrix} \leq \gamma h_i^{Afc}(p_i) + \frac{\partial h_i^{Afc}}{\partial x_m^c}\dot{x}_c + \frac{\partial h_i^{Afc}}{\partial y_m^c}\dot{y}_c + \frac{\partial h_i^{Afc}}{\partial \alpha}\dot{\alpha} \tag{19}$$