Multi-robot Collision avoidance using control barrier Functions

Luca Tirel, Ali Mohamed Ali Hassan Ali, Alessandro Reali
Dipartimento di Ingegneria informatica, automatica e gestionale
Sapienza, University of Rome
Rome, Lazio

***Abstract*—This project presents the application of safety barrier certificates for collision-free behaviors in multirobot systems through a quadratic programming approach that minimizes the difference between the actual and the nominal controllers subject to safety constraints, implemented to modify the nominal controllers. In particular, two tasks have been considered: the first one consisting in the position-swapping of all the robots involved in a multirobot system inside a confined workspace where collisions were very likely to occur. The second one consisting in a multirobot system in which each robot present in it must be driven to some final and pre-defined poses without any collision with fixed obstacles present in the environment or other vehicles; then, a leader-follower system has been implemented. The satisfaction of both tasks has been studied and simulated with Matlab and Simulink environments and by exploring the differences between centralized and decentralized barrier functions.**

1. Introduction

Navigation is a fundamental concern for teams of mobile robots. The main task that a designed coordinated controller must satisfy is that of driving the behaviour of the team while achieving and maintaining formations, covering areas, or collective transport. Secondly, safety, in terms of collision-avoidance, is oftentimes added as a secondary controller that overrides the existing controllers on individual robots if they are about to collide; it is critical that the robot avoids obstacles to ensure its own safety but also of the people's around it [6]. Thus, what is ultimately deployed on the system is a combination of a "formally" designed nominal controller together with a "hand-crafted" collision avoidance algorithm with the basic idea being that the multirobot system executes the nominal, primary controller most of the time, while the secondary, collision avoidance controller takes over when the robots are too close to each other [1]. Furthermore, the more the number of robots, the more the "robot density" increases, having the result that the collision-avoidance controller starts to dominate the behavior of the robot team: this means that the desired, global properties can no longer be ensured [3].
A possible solution to the problem of avoiding collisions is to make collision-avoidance an integral part of the coordinated control design. However, this significantly increases the complexity of the design-task and, more importantly, makes the many proposed design tools no longer applicable [3]. A remedy to this problem is to design a collision avoidance controller that is minimally invasive in the sense that it only modifies the nominal controller when collision is truly imminent [1].
When the robots are about to collide, barrier certificates act to ensure that the collisions are avoided. The key tool for producing these safety-critical controllers is to utilize control barrier certifications to prevent the robots from entering unsafe sets-naturally expressed in a minimally invasive fashion through the use of optimization-based controllers [3].
In particular, in this work, a safe set is defined over the joint state space of the multirobot team modeled as double integrators, which yields barrier certificates, i.e., control barrier functions, that are used to ensure that the robots remain in this set for all time. As a result, if a controller satisfies the safety barrier certificates, it is safe. To do so, a given control law for coordination can be implemented in the cost of a quadratic program (QP) based control law in which the actual control action is taken to be as close as possible to the nominal control action; this controller has as constrains those ones given by the safety barrier certificate that enforces collision free behavior.
The remainder of this project is organized as follows. Section 2 introduces the problem statement, focusing in particular on double integrator robots, their control and some reminds on the theory of Quadratic Programming; sections 3 revisits the concepts of (zeroing) control barrier functions, which are incorporated into the optimization based controller as safety barrier constraints; sections 4-5 delve into the issue of safety barrier certificates with quadratic programming, formulating them before in a centralized manner and the in a decentralized formulation assigning the admissible control space to different agents; section 6 exploits the concepts of safety barrier certificates by including the concepts of Relaxed and Feasible Safety Barriers; chapter 7 shows the issues of deadlock. Finally, in section 8 the experimental results are presented: these are split into two experiments, the first one dealing with the robots-switching positions while satisfying the control barrier function constraints, the second

one dealing with obstacle avoidance and leader-follower.

2. Problem Statement

2.1. System description

The robots used in paper [1] used as reference for this project are *Khepera robots*. For simplicity, they have been modeled as double integrators of the form

$$\begin{bmatrix}\dot{p}_i\\ \dot{v}_i\end{bmatrix} = \begin{bmatrix}0 & I_{2\times 2}\\ 0 & 0\end{bmatrix} \times \begin{bmatrix}p_i\\ v_i\end{bmatrix} + \begin{bmatrix}0\\ I_{2\times 2}\end{bmatrix} u_i \tag{1}$$

in which $p_i \in \mathbb{R}^2$, $v_i \in \mathbb{R}^2$, and $u_i \in \mathbb{R}^2$ are the positions, velocities, and inputs (acceleration commands) of agent i, respectively in a set of $\mathcal{M}$ = {1,...,N} mobile robots. Moreover, the velocity and acceleration of the i-th agent have been limited by $\|v_i\|_\infty \leq \beta_i$ and $\|u_i\|_\infty \leq \alpha_i$. Then, the relative distances and velocities of two pairs of agents are, respectively, reported as $\Delta p_{ij} = p_i - p_j$ and $\Delta v_{ij} = v_i - v_j$.

2.2. Control

All the tasks presented in this project have the common goal of developing a navigation system for a group of (variable-number) mobile robots, described as double integrators. To do so, a control objective must be defined with the determination of appropriate constraint on the inputs that guarantee the achievement of the desired configurations to the robot involved in the simulation while ensuring safety. Following the theory on [7], three types of contraints act on our systems:
Hard constraints: they represent constraints that must not be violated under any condition; in our case, it is “keep a safe distance from the other robots (and, if any) from the static obstacles in the environment”
Soft constraints: the objective of the control; in our case, it is to drive the robots to a desired configuration $(q_d, 0)$ (obs: in case of robots following a leader, the desired configuration of the followers is the configuration of the leader at that time).
Actuators constraints: bound on the maximum and minimum accelerations: $\|u_i\|_\infty \leq \alpha_i$.
In order to ensure that the nominal, primary control objective is respected to the highest degree possible, the collision avoidance strategy needs to be minimally invasive in the sense that it should modify the nominal controller as little as possible [1]. A Quadratic programming (QP) approach may be used to find control inputs that simultaneously satisfy the above different constraints while ensuring a minimal "invasiveness". QP is a process of solving certain mathematical optimization problems involving quadratic functions. In our specific approach, we seek to minimize a given quadratic function subject to linear constraints on the variables. Thus, we can write

$$\min_x \{\frac{1}{2}x^T H x + f^T x\} \tag{2}$$

subject to contraints

$$Ax \leq b \rightarrow \text{inequality constraint}$$
$$lb \leq x \leq ub \rightarrow \text{bound constraint.}$$

In our case, since the safety constraint is linear in the control signal, it is possible to choose our QP so that the quadratic cost function penalizes deviations from the nominal controller in a least-squares sense. Thus, the controller minimizes the difference (in norm) between the actual control command u_i and nominal control command $\hat{u}_i$, that’s to say $\|u_i - \hat{u}_i\|$ while ensuring safety, thus $A_{ij}u \leq b_{ij}$ (in constraint form).

3. Background on Control Barrier Functions

Control Barrier Functions (CBFs) are level-set functions used to enable controller synthesis for safety requirements in navigation from a nonlinear control perspective, that’s to say they have the goal of ensuring that the robots execute collision-free trajectories. They are similar to Lyapunov Functions (CLFs) in that they can be used to guarantee desired system properties without explicitly having to compute the forward reachable set and in guaranteeing the forward invariance of the desired set (called **safe set**) by constraining the time derivative of the CBFs within prescribed bounds. The fundamental idea behind CBFs is thus to design them in such a way that the agents always remain in the safe set [6].
The safe set, i.e. the set where we wish that the aggregate robot states should stay, may be defined as C $\subset \mathbb{R}^n$. The task is to design a controller that guarantees the forward invariance of set C, i.e., given a nonlinear system of the form $\dot{x} = f(x) + g(x)u$, its solutions that start in C stay in C for all time (if x(0) $\in$ C $\Rightarrow$ x(t) $\in$ C, $\forall$ t$\geq$ 0) [1]. The invariance of C can be established through the use of a **barrier function** (or **barrier certificate**) B : C $\rightarrow \mathbb{R}$, that satisfies the properties:

$$\inf_{x\in Int(C)} B(x) \geq 0, \quad \lim_{x\to\partial C} B(x) = \infty \tag{3}$$

Mathematically, we then assume that the set C can be defined as the level set to a particular smooth function $h(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ such that we have the following expressions:

$$\begin{aligned} C &= \{x \in \mathbb{R}^n | h(x) \geq 0\},\\ \partial C &= \{x \in \mathbb{R}^n | h(x) = 0\},\\ int(C) &= \{x \in \mathbb{R}^n | h(x) > 0\}.\end{aligned} \tag{4}$$

We note that the time derivative of h(x) along the state trajectories is

$$\frac{dh(x)}{dt} = \frac{\partial h(x)}{\partial x}\dot{x} = \frac{\partial h(x)}{\partial x}(f(x) + g(x)u) \xrightarrow{\text{Lie formalism}} \frac{dh(x)}{dt} = L_f h(x) + L_g h(x) u \tag{5}$$

Definition: Given a dynamical system $\dot{x} = f(x) + g(x)u$ and the set C for a continuously differentiable function h, if there exist a locally Lipschitz extended class $\mathcal{K}$ function

α (strictly increasing and $\alpha(0)$ = 0) and a set C $\subseteq$ D$\subset$ $\mathbb{R}^n$ such that, for all x $\in$ D,

$$\sup_{u \in U}\{L_f h(x) + L_g h(x)u + \alpha(h(x))\} \geq 0 \tag{6}$$

then the function h is called a **Zeroing Control Barrier Function** (ZCBF) defined on D and the set of feasible control inputs (control space) is [6] :

$$S(x) = \{u \in U | L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0\} \tag{7}$$

In order to guarantee that C is forward invariant, we may choose any Lipschitz continuous controller u : D $\rightarrow$ $\mathbb{R}$ such that u $\in$ S(x) for the system $\dot{X}$ renders the set C forward invariant; moreover, C is asymptotically stable in D.
The extended class-$\mathcal{K}$ function α regulates how fast the state of the system can approach the boundary of C. Therefore, different choices of it, lead to different behaviors near the boundary. In our case, the particular choice of α(h(x)) = $\gamma h^3(x)$ with γ > 0 will be adopted, which means that the controller needs to satisfy

$$L_f h(x) + L_g h(x)u + \gamma h^3(x) \geq 0 \tag{8}$$

to render the set C forward invariant. This form of the barrier function can equivalently be stated as a reciprocal barrier function of the form [1]:

$$B(x) = \frac{1}{h(x)} \tag{9}$$

4. Formulation of Control Barrier Functions

In this section we will deduce the control barrier functions to be used in the centralized and decentralized algorithms and defined the notion of neighborhood and the case of fixed obstacle to be avoided.

4.1. Centralized Control Barrier Function

In the Centralized Approach the authors of [1] used pairwise constraints within all robots to guarantee that a safety distance D_s between any of them was ensured. Algorithms to avoid imminent collision with static obstacles were developed by decelerating the agent to zero velocity with the maximum braking force. However, to avoid imminent collision with a moving agent, the relative velocity between two agents needs to be reduced to zero instead of the absolute velocity. Consider any two agents i and j, the relative position and relative velocity between them are $\Delta\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ and $\Delta\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. The normal component of the relative velocity ($\Delta\bar{v} = \|\Delta\dot{\mathbf{p}}_{ij}\| = \frac{\Delta\mathbf{p}_{ij}^T}{\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{v}_{ij}$) is the actual component that might lead to collision between agents i and j, while the tangent component of $\Delta\mathbf{v}_{ij}$ only leads to rotation around each other. Therefore, we need to regulate $\Delta\bar{v}$ so that imminent collisions can be avoided if the maximum relative braking force is applied.

Assuming the normal component of the relative velocity between agents i and j is $\Delta\bar{v}(t_0)$ at the current time instance t_0, it takes the time $T_b = \frac{0-\Delta\bar{v}(t_0)}{\alpha_i+\alpha_j}$ to reach $\Delta\bar{v}(t_0 + T_b) = 0$, while the maximum braking acceleration $(\alpha_i + \alpha_j)$ is applied to both robots. In order to remain farther away from the safety distance D_s, the following safety constraint needs to be satisfied

$$\|\Delta\mathbf{p}_{ij}\| + \int_{t_0}^{t_0+T_b} \Delta\bar{v}(t_0 + t)\,\mathrm{d}t \geq D_s, \quad \forall i \neq j \tag{10}$$

where $\Delta\bar{v}(t_0 + t) = \Delta\bar{v}(t_0) + (\alpha_i + \alpha_j)\,t$, which means that

$$\|\Delta\mathbf{p}_{ij}\| - \frac{(\Delta\bar{v})^2}{2(\alpha_i + \alpha_j)} \geq D_s, \quad \forall i \neq j \tag{11}$$

As such, the pairwise safe set C_{ij} is defined as

$$C_{ij} = \left\{(\mathbf{p}_i, \mathbf{v}_i) \in \mathbb{R}^4 \mid h_{ij}(\mathbf{p}, \mathbf{v}) \geq 0\right\} \quad \forall i \neq j \tag{12}$$

$$h_{ij}(\mathbf{p}, \mathbf{v}) = \sqrt{2(\alpha_i + \alpha_j)(\|\Delta\mathbf{p}_{ij}\| - D_s)} + \frac{\Delta\mathbf{p}_{ij}^T}{\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{v}_{ij} \tag{13}$$

4.2. Fixed Obstacle Control Barrier Function

We note that the safety barrier certificates can also deal with static or moving obstacles. We can bound the obstacles with circles and consider them as agents with no control input. For example, consider a moving obstacle k with radius R_k centered at $\mathbf{p}_k$, a ZCBF $\bar{h}_{ik}(\mathbf{p}, \mathbf{v})$ can be designed to ensure that agent i does not collide with obstacle k as:

$$\bar{h}_{ik}(\mathbf{p}, \mathbf{v}) = \sqrt{2\alpha_i\left(\|\Delta\mathbf{p}_{ik}\| - \left(\frac{D_s}{2} + R_k\right)\right)} + \frac{\Delta\mathbf{p}_{ik}^T}{\|\Delta\mathbf{p}_{ik}\|}\Delta\mathbf{v}_{ik} \tag{14}$$

This will ensure that the safe distance is enforced between them, avoiding collisions.

4.3. Centralized Control Barrier Function with neighborhood sets

In the centralized approach the robot swarm has to be modeled as a complete not oriented graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$. The number of edges in a complete graph are $\frac{N\cdot(N-1)}{2}$, that is polynomial in $\mathcal{N}$. This will create computational inefficiency as the number of robots increases. Therefore one can think to define a neighbourhood radius and consider only the edges, and thus the constraints, relative to robots that lies within it, and

$$D_N^i = D_s + \frac{1}{2(\alpha_i + \alpha_{\min})}\left(\sqrt[3]{\frac{2(\alpha_i + \alpha_{\max})}{\gamma}} + \beta_i + \beta_{\max}\right)^2 \tag{15}$$

is the choice of the radius of the neighborhood, $\alpha_{\min}$ =

$\min_{j\in M}\{\alpha_j\}$ and $\alpha_{\max} = \max_{j\in M}\{\alpha_j\}$ are lower and upper bounds of all agents' acceleration limits, and $\beta_{\max} = \max_{j\in M}\{\beta_j\}$ is the upper bound of all agents' speed limits. With this notion of neighborhood, we will say that each agent only needs to consider its nearby agents to avoid collision.

In this way only the nearest agents will be considered for the constraints activation, reducing the overall computational complexity, but still not lowering the polynomial dependency of the complexity from the number of robots.

5. Quadratic programming

In this section the notion of Quadratic programming will be deduced for the following algorithms: centralized, centralized with neighborhood activation of control barrier function, decentralized and finally decentralized with neighborhood.

A quadratic programming problem is an optimization problem involving a quadratic cost function and linear constraints, and has the following form :

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^T Q x + c^T x \\ \text{subject to} \quad & Ax \le b, \end{aligned}$$

where Q and c^T are a positive semi-definite matrix and a non-negative vector, x is the optimization variable vector, A is the linear constraints matrix. In order to avoid collisions, the controller should modify the nominal controller as little as possible. This will result in the definition of the minimization objective as the squared difference between the safe optimal control, and the nominal one.

5.1. Centralized Quadratic Programming

Starting from the definitions of centralized control barrier function defined at equation (13) where $h_{ij}(\mathbf{p},\mathbf{v})$ is the level set function of the set C_{ij} as well as the ZCBF candidate used to ensure the forward invariance of C_{ij}. For ease of notation, we use h_{ij} to denote $h_{ij}(\mathbf{p},\mathbf{v})$. As seen, the forward invariance of C_{ij} requires the satisfaction of the ZCBF constraint in (8). the safety barrier constraint can be written as

$$\begin{aligned} -\Delta\mathbf{p}_{ij}^T\Delta\mathbf{u}_{ij} \le & \gamma h_{ij}^3 \|\Delta\mathbf{p}_{ij}\| - \frac{\left(\Delta\mathbf{v}_{ij}^T\Delta\mathbf{p}_{ij}\right)^2}{\|\Delta\mathbf{p}_{ij}\|^2} + \|\Delta\mathbf{v}_{ij}\|^2 \\ & + \frac{(\alpha_i+\alpha_j)\,\Delta\mathbf{v}_{ij}^T\Delta\mathbf{p}_{ij}}{\sqrt{2(\alpha_i+\alpha_j)(\|\Delta\mathbf{p}_{ij}\|-D_s)}}, \quad \forall i \ne j \end{aligned} \tag{16}$$

thus the safety barrier constraint can be written as a linear constraint in $\mathbf{u}_i$ and $\mathbf{u}_j$, which in turn can be represented as a linear set of inequality constraints as $A_{ij}\mathbf{u} \le b_{ij}$, with

$$\begin{gathered} A_{ij} = [0,\ldots,\underbrace{-\Delta\mathbf{p}_{ij}^T}_{\text{agent } i},\ldots,\underbrace{\Delta\mathbf{p}_{ij}^T}_{\text{agent } j},\ldots,0] \\ \text{and } b_{ij} = \gamma h_{ij}^3\|\Delta\mathbf{p}_{ij}\| - \frac{\left(\Delta\mathbf{v}_{ij}^T\Delta\mathbf{p}_{ij}\right)^2}{\|\Delta\mathbf{p}_{ij}\|^2} + \\ \frac{(\alpha_i+\alpha_j)\,\Delta\mathbf{v}_{ij}^T\Delta\mathbf{p}_{ij}}{\sqrt{2(\alpha_i+\alpha_j)(\|\Delta\mathbf{p}_{ij}\|-D_s)}} + \\ \|\Delta\mathbf{v}_{ij}\|^2 . \end{gathered} \tag{17}$$

The nominal primary control objective must be respected to the highest degree possible. And, as we have seen, the safety constraints are linear in the control signal. As previously stated, we can add a quadratic cost that penalizes deviations from the nominal controller (in the least-squares sense), resulting in a quadratic programming (QP) problem. As a consequence, the QP-based controller minimizes the difference between the actual control command $\mathbf{u}_i$ and nominal control command $\hat{\mathbf{u}}_i$, while ensuring safety using the centralized safety barrier certificates discussed in the prior section

$$\begin{aligned} \mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{R}^{2N}}{\text{argmin}} J(\mathbf{u}) = \sum_{i=1}^{N} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 \\ \text{s.t. } A_{ij}\mathbf{u} \le b_{ij}, \quad \forall i \ne j \\ \|\mathbf{u}_i\|_\infty \le \alpha_i, \quad \forall i \in M \end{aligned} \tag{18}$$

The resulting controller $u*$ mimics the nominal controller $\hat{\mathbf{u}}$ completely when the system is safe, and only modifies its behavior when collisions are truly imminent, which is what we mean by minimally invasive safe modification of the nominal controller $\hat{\mathbf{u}}$.

In the centralized approach the QP-based controller requires central computation and coordination, which suffers from poor scalability, reactiveness, and robustness as the number of agents grows, thus more scalable approaches will be needed.

5.2. Centralized Quadratic Programming with neighborhood notion

The QP-based controller in (18) can be simplified by only checking the safety of a multirobot system using disk information graphs defined at equation (15)

$$\begin{aligned} \mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{R}^{2N}}{\text{argmin}} J(\mathbf{u}) = \sum_{i=1}^{N} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 \\ \text{s.t. } A_{ij}\mathbf{u} \le b_{ij} \quad \forall i \in M, \forall j \in N_i \\ \|\mathbf{u}_i\|_\infty \le \alpha_i \quad \forall i \in M \end{aligned} \tag{19}$$

The radius D_N^i of the disk information graph can be designed by choosing appropriate γ, such that D_N^i is no larger than the sensing range of agents. Note that this notion of neighborhood is still valid when the safety barrier certificates are distributed to individual agent.

5.3. Decentralized Approach

The centralized safety barrier certificates proposed in Section 4.1 ensure probably safe multirobot coordination, while the reliance on a central coordination unit potentially compromises the multirobot system's scalability, reactiveness, and robustness. To address those issues, the agent i-th will run his own version of the QP problem, fed with its specific control u_i, its dynamic defined by $f_i(x)$ and $g_i(x)$, and considering pairwise barriers between robot i-th and all the other robots that are, at most (considering neighborhood notion), $N-1$.

The safety barrier certificates define the admissible control space, which can be further partitioned into smaller subsets according to each agents' acceleration limits, taking into account that some agents may be more agile with respect to others, and can perform faster collision avoidance maneuvers. More specifically, the pairwise safety barrier constraint between agent i and agent j is distributed to each agent as:

$$\begin{aligned} -\Delta \mathbf{p}_{ij}^T \mathbf{u}_i &\leq \frac{\alpha_i}{\alpha_i+\alpha_j} b_{ij} \\ \Delta \mathbf{p}_{ij}^T \mathbf{u}_j &\leq \frac{\alpha_j}{\alpha_i+\alpha_j} b_{ij}. \end{aligned} \tag{20}$$

We can note that nothing changed except for the a gain in the constraints, based on agent-to-agent bounds on control.

With the decentralized safety barrier constraints and the notion of neighborhood, the sensing and computation tasks are completely distributed to each individual agent. Each robot

$i \in M$ runs their own version of QP-based controller

$$\begin{aligned} \mathbf{u}_i^* = \underset{\mathbf{u}_i \in \mathbb{R}^2}{\operatorname{argmin}} & J(\mathbf{u}_i) = \|\mathbf{u}_i - \hat{\mathbf{u}}_i\| \\ \text{s.t. } & \bar{A}_{ij} \mathbf{u}_i \leq \bar{b}_{ij}, \quad \forall j \in N_i \\ & \|\mathbf{u}_i\|_\infty \leq \alpha_i \end{aligned} \tag{21}$$

where $\bar{A}_{ij} = -\Delta \mathbf{p}_{ij}^T, \bar{b}_{ij} = \frac{\alpha_i}{\alpha_i+\alpha_j} b_{ij}$.

6. Relaxed Safety Barrier Certificates & Feasible Safety Barrier Certificates

6.1. Relaxed Safety Barrier Certificates

The decentralized safety barrier certificates remove the need for central computation at the cost of using a more conservative collision avoidance strategy, i.e., the nominal controller might be modified earlier than truly necessary to avoid collisions.

Motivated by the need to enlarge the admissible control space, the relaxed ZCBF is introduced through a relaxation parameter $k_r(t) \in [1, \infty)$, which is continuous in t.

The idea now is to insist that $\dot{h}(x) \geq -k_r(t)\kappa(h(x))$, and prove that we can still formally guarantee the forward invariance of the set C when the ZCBF constraint is relaxed with $k_r(t)$. And the relaxed, feasible control set $S_r(x)$ becomes

$$\begin{gathered} S_r(x) = \{u \in U \mid L_f h(x) + L_g h(x) u + k_r(t)\kappa(h(x)) \geq 0\} \\ x \in D \end{gathered} \tag{22}$$

Since $k_r(t)$ can be freely adjusted online in response to changing environmental conditions, it is selected as an optimization decision variable in the QP-based controller.

For agent i, the pairwise safety barrier constraint with agent j is relaxed with a relaxation factor k_{rj}

$$\begin{aligned} -\Delta \mathbf{p}_{ij}^T \Delta \mathbf{u}_{ij} \leq & k_{rj} \gamma h_{ij}^3 \|\Delta \mathbf{p}_{ij}\| \\ & - \frac{\left(\Delta \mathbf{v}_{ij}^T \Delta \mathbf{p}_{ij}\right)^2}{\|\Delta \mathbf{p}_{ij}\|^2} + \|\Delta \mathbf{v}_{ij}\|^2 \\ & + \frac{(\alpha_i+\alpha_j) \Delta \mathbf{v}_{ij}^T \Delta \mathbf{p}_{ij}}{\sqrt{2(\alpha_i+\alpha_j)(\|\Delta \mathbf{p}_{ij}\| - D_s)}} \quad \forall i \neq j \end{aligned} \tag{23}$$

which are assembled into the relaxed safety barrier certificates. In the nominal case, there is no relaxation on the safety barrier constraints, i.e., $\hat{K}_r = \left[\hat{k}_{r1}, \hat{k}_{r2}, \ldots, \hat{k}_{rM}\right]^T = \mathbf{1}$ with $M = |N_i|$ being the cardinality of agent i 's neighborhood set. The actual relaxation vector $K_r = [k_{r1}, k_{r2}, \ldots, k_{rM}]^T$ is added to the cost of the QP-based controller to penalize deviation from $\hat{K}_r$ $(\mathbf{u}_i^*, K_r^*) = \underset{\mathbf{u}_i \in \mathbb{R}^2, K_r \in \mathbb{R}^M}{\operatorname{argmin}} J(\mathbf{u}_i, K_r) = \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 + c_K \left\|K_r - \hat{K}_r\right\|^2$ s.t. $\tilde{A}_{ij}\mathbf{u}_i \leq \frac{\alpha_i}{\alpha_i+\alpha_j}\tilde{b}_{ij}, \quad \forall j \in N_i \ \|\mathbf{u}_i\|_\infty \leq \alpha_i$, where c_K is the weight for the cost of relaxation, $\tilde{A}_{ij} = \left[-\mathbf{p}_{ij}^T\right]$

$$\begin{aligned} \tilde{b}_{ij} = & k_{rj} \gamma h_{ij}^3 \|\Delta \mathbf{p}_{ij}\| - \frac{\left(\Delta \mathbf{v}_{ij}^T \Delta \mathbf{p}_{ij}\right)^2}{\|\Delta \mathbf{p}_{ij}\|^2} + \\ & + \frac{(\alpha_i+\alpha_j) \Delta \mathbf{v}_{ij}^T \Delta \mathbf{p}_{ij}}{\sqrt{2(\alpha_i+\alpha_j)(\|\Delta \mathbf{p}_{ij}\| - D_s)}}. \end{aligned} \tag{24}$$

Let the optimization parameter be $\mu_i = \left[\mathbf{u}_i^T, \sqrt{c_K} K_r^T\right]^T$, the QP-based controller can be rearranged as

$$\begin{aligned} \mu_i^* = \underset{\mu_i \in \mathbb{R}^{2+M}}{\operatorname{argmin}} \quad & J(\mu_i) = \|\mu_i - \hat{\mu}_i\|^2, \\ \text{s.t.} \quad & \bar{A}_{ij}\mu_i \leq \frac{\alpha_i}{\alpha_i+\alpha_j}\bar{b}_{ij}, \quad \forall j \in N_i \\ & \|\mathbf{u}_i\|_\infty \leq \alpha_i \end{aligned}$$

where $\hat{\mu}_i = \left[\hat{\mathbf{u}}_i^T, \sqrt{c_K}\hat{K}_r^T\right]^T$, $\bar{A}_{ij} = \left[-\mathbf{p}_{ij}^T, 0, \ldots, -\frac{\alpha_i h_{ij}^3 \|\Delta \mathbf{p}_{ij}\|}{(\alpha_i+\alpha_j)\sqrt{cK}}, \ldots 0\right]$,

$$\begin{aligned} \bar{b}_{ij} = & - \frac{\left(\Delta \mathbf{v}_{ij}^T \Delta \mathbf{p}_{ij}\right)^2}{\|\Delta \mathbf{p}_{ij}\|^2} + \|\Delta \mathbf{v}_{ij}\|^2 \\ & + \frac{(\alpha_i+\alpha_j) \Delta \mathbf{v}_{ij}^T \Delta \mathbf{p}_{ij}}{\sqrt{2(\alpha_i+\alpha_j)(\|\Delta \mathbf{p}_{ij}\| - D_s)}}. \end{aligned}$$

By putting K_r into the optimization cost, the QP-based controller will automatically expand the admissible control space for those agents that are unnecessarily constrained.

It should be noted that the continuity of K_r is guaranteed when using the QP-based controller, as long as the nominal controller $\hat{\mathbf{u}}_i$ is Lipschitz continuous.

The safety barrier certificates ensure safety as long as all agents' control actions stay within the admissible control space. However, the admissible control space might become empty in extreme scenarios, where the QP-based controller becomes infeasible, i.e., no solution exists and safety cannot be guaranteed. This can be remedied by the robustness feature of ZCBF, which means the ZCBF will force the states back to the safe set C when violation occurs. But it is important to synthesize safety barrier certificates with guaranteed feasibility such that a solution to the QP-based controller always exists.

6.2. Feasible Safety Barrier Certificates

Note that as the swarm size N grows, Su might become empty, which is a feasibility problem. This can be remedied by the robustness feature of ZCBF, which means the ZCBF will force the states back to the safe set C when violation occurs [26]. But it is important to synthesize safety barrier certificates with guaranteed feasibility such that a solution to the QP-based controller always exists.
Intuitively, if all agents can decelerate to zero velocity without colliding into each other, then they can stay static to remain safe thereafter. Therefore, we can synthesize a ZCBF which ensures that decelerating to zero velocity is always a feasible safe control action. This is achieved by designing two modes of operations: normal mode and braking mode. The safety barrier certificates operate in the normal mode when the QP-based controller is feasible, and switch to the braking mode when it is infeasible. In the braking mode, the agent will apply the maximum braking force until it stops.

The decentralized QP-based controller using the feasible safety barrier certificates can be written as

$$\begin{aligned} \mathbf{u}_i^* = \underset{\mathbf{u}_i \in \mathbb{R}^2}{\operatorname{argmin}} J(\mathbf{u}_i) &= \|\mathbf{u}_i - \hat{\mathbf{u}}_i\| \\ \text{s.t. } \hat{A}_{ij}\mathbf{u}_i &\leq \hat{b}_{ij}, \quad \forall j \neq i, \\ \|\mathbf{u}_i\|_\infty &\leq \alpha_i \end{aligned} \tag{25}$$

where $\hat{A}_{ij} = \frac{\|\mathbf{v}_i\|\|\mathbf{v}_j\|}{8\alpha_i\alpha_j}\mathbf{v}_j + \frac{\mathbf{v}_i\mathbf{v}_j\|\mathbf{v}_j\|}{8\alpha_i\alpha_j\|\mathbf{v}_i\|}\mathbf{v}_i - \frac{\Delta\mathbf{p}_{ij}\mathbf{v}_i}{2\alpha_i\|\mathbf{v}_i\|}\mathbf{v}_i - \frac{\|\mathbf{v}_i\|\Delta\mathbf{p}_{ij}}{2\alpha_i} + \frac{D_s\mathbf{v}_i}{\alpha_i} + \frac{\|\mathbf{v}_j\|^2\mathbf{v}_i}{4\alpha_i\alpha_j}, \hat{b}_{ij} = \Delta\mathbf{p}_{ij}\Delta\mathbf{v}_{ij} + \frac{\|\mathbf{v}_i\|}{2\alpha_i}\Delta\mathbf{v}_{ij}\mathbf{v}_i + \frac{1}{2}\gamma\hat{h}_{ij}^3$. Note that if $\mathbf{v}_i = \mathbf{0}$ (agent i is stationary), the limit of $\hat{A}_{ij}$ is used, i.e., $\lim_{\mathbf{v}_i\to\mathbf{0}}\hat{A}_{ij} = 0$ and $\lim_{\mathbf{v}_i\to\mathbf{0}}\hat{A}_{ji} = -\frac{\Delta\mathbf{p}_{ji}\mathbf{v}_j}{2\alpha_j\|\mathbf{v}_j\|}\mathbf{v}_j - \frac{\|\mathbf{v}_j\|\Delta\mathbf{p}_{ji}}{2\alpha_j} + \frac{D_s\mathbf{v}_j}{\alpha_j}$. Intuitively, this means that agent i is free to choose its control action when it is stationary, while other agents can always react fast enough to avoid colliding with it.

This QP-based controller might still face feasibility issues. However, the agent can always decelerate to zero velocity safely due to Lemma VI.1. This results in the following hybrid braking controller

$$\mathbf{u}_i = \begin{cases} \mathbf{u}_i^*, & \text{if } \mathbf{u}_i^* \text{ exists,} \\ -\alpha_i\mathbf{v}_i, & \text{if } \mathbf{u}_i^* \text{ does not exist, and } \|\mathbf{v}_i\| \neq 0 \\ 0, & \text{if } \mathbf{u}_i^* \text{ does not exist, and } \|\mathbf{v}_i\| = 0 \end{cases} \tag{26}$$

7. Deadlock

When the objectives of multiple agents conflict with the safety barrier certificates, the agents might get stuck into a deadlock. In the deadlock scenario, the agents are safe but their tasks cannot be completed. Deadlock occurs because the safety barrier certificates are designed to take the local information only. In order to detect and resolve the deadlock issue.

Definition: A robot agent i is said to be stuck in a deadlock, if it remains stationary $(u_i = 0$ and $v_i = 0)$ and the nominal control command $\hat{u}_i \neq 0$.

With this definition, the deadlock scenarios can be further classified into three types based on the solution to the QP problem . The admissible control space for the QP problem is a convex polygon P_i defined as the intersection of multiple half spaces, i.e.,

$$P_i = \left\{\mathbf{u}_i \in \mathbb{R}^2 \mid \bar{A}_{ij}\mathbf{u}_i \leq \bar{b}_{ij}, \quad \forall i \neq j\right\} \tag{27}$$

where P_i is a decentralized counterpart of the centralized admissible control space S_u. The size of the feasible control space, termed the width of the feasible set, can be evaluated with a Linear Program (LP)

$$\begin{array}{ll} \min_{\mathbf{u}_i\in\mathbb{R}^2, \delta_{LP}\in\mathbb{R}} & \delta_{LP} \\ \text{s.t.} & \bar{A}_{ij}\mathbf{u}_i \leq \bar{b}_{ij} + \delta_{LP}, \quad \forall i \neq j, \\ & \|\mathbf{u}_i\|_\infty \leq \alpha_i \end{array} \tag{28}$$

The solution of the LP characterizes how much control margin is left for the strictest safety barrier constraint. If $\delta_{LP} \leq 0$, the corresponding QP is solvable. A more negative δ_{LP} indicates larger admissible control space. Otherwise, no feasible control option is available, and the admissible control space is empty. The deadlock scenarios are categorized into the following three cases based on the relation between u_i and P_i 1) Type 1 deadlock: $\delta_{LP} < 0, u_i = 0 \in \operatorname{vertex}(P_i)$; 2) Type 2 deadlock: $\delta_{LP} < 0, u_i = 0 \in \operatorname{edge}(P_i)$; and 3) Type 3 deadlock: $\delta_{LP} \geq 0$. It should be noted that these three types of deadlock comprise all possible types of deadlocks. This is because u_i is either on the edge or the vertex of P_i, when the optimal solution of the constrained QP controller $(u_i = 0)$ is different from the unconstrained optimal solution $(\hat{u}_i \neq 0)$, due to Karush-KuhnTucker (KKT) conditions.

One way to resolve the deadlock scenarios is to perturb the QP controller so that the robot agents can move around each other when they get stuck. The QP controller needs to be perturbed consistently, because multiple robot agents might still be acting against each other with random perturbations. Inspired by the traffic rule used in transportation to resolve conflicts , the following consistent perturbation method is proposed to resolve different deadlocks 1) Type 1 deadlock:

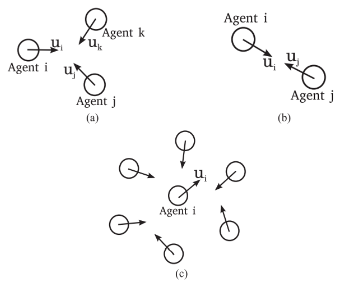

Figure 1. Three types of Deadlocks for robot agent i in a multirobot system. (a) Type 1 deadlock. (b) Type 2 deadlock. (c) Type 3 deadlock.

As illustrated in Fig. 1, the left barrier constraint is relaxed $\left(k_{\gamma(\text{ left })}>1\right)$ and the right barrier constraint is compressed $\left(k_{\gamma(\text{ right })}<1\right)$. 2) Type 2 deadlock: As illustrated in Fig. 9(b), $\hat{\mathbf{u}}_i$ is perturbed with $\delta^{\perp}=k_{\delta}\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}\hat{u}_i$, which is a normal perturbation to the left of $\hat{u}_i$. 3) Type 3 deadlock: $\mathbf{u}_i=0$, no perturbation is performed since no admissible perturbation is available.

8. Experiments and Results

8.1. Task1: Robots switching positions

The first experiment we performed concerns the safe driving of a set of 8 robots in a crowded environment, avoiding collisions. The robots starts from positions that are not fully symmetrical between them and are driven to reference set-points that are almost symmetrical with respect to the origin. All the simulations are performed using MATLAB and Simulink, using mainly Symbolic Toolbox and Optimization Toolbox. The desired, and unsafe, control action is firstly computed using a simple PID controller and fed to the QP optimization problem, that compute the safe control action. The definition of the barrier and its Jacobian is done symbolically and used to compute the constraints.

8.1.1. Centralized Safety Barrier certificates approach. First the problem is solved in a centralized way using a pairwise constraint for each couple of robots. This results in a total number of $\frac{N \cdot N-1}{2}$ constraints always activated, plus the ones on the control input.

In the following plot we can see robot trajectories when we use a fully centralized controller.

If the system was controlled thought the PID controller only collisions between robots would happens in the crowded area. In fact we can see from the next plot how the control action along x-axis vary for the first robot in the two approaches, driving the robot in a safe way through the crowded area.

8.1.2. Decentralized Safety Barrier certificates approach. The algorithm is then adapted to the decentralized case, in order to use a control system on each robot, lowering complexity of the problem to be solved. This will reduce the number of constraints involving barriers to $N-1$, that is linear in N.

In the following plot we can see the trajectories in the decentralized case:

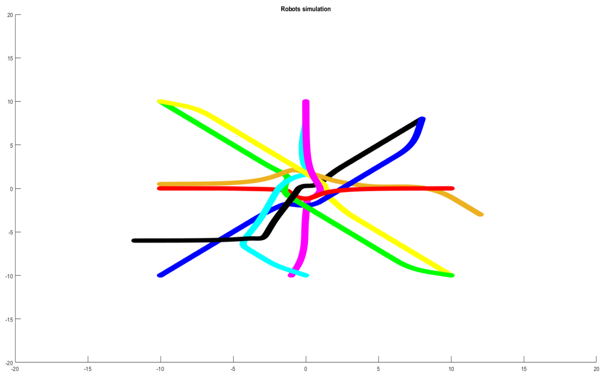

Figure 2. Robots' trajectories in the centralized case

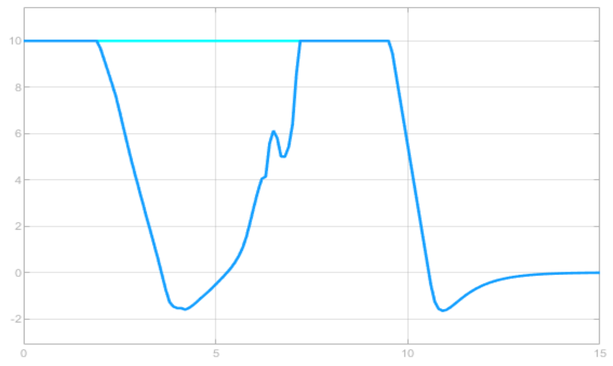

Figure 3. Comparison of the control input along x-axis of robot 1 from PID (cyan) and barriers (blue)

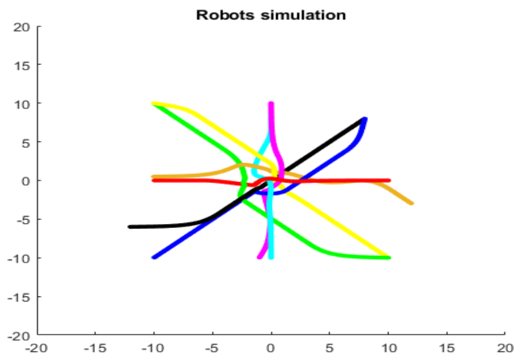

Figure 4. Robots' trajectories in the decentralized case

Now we would like to show how the position along y-axis of the third robot (the one that in Figures 2 and 4 has the cyan colour) is varying in the two cases. We expected that in the decentralized approach, due to a lack of central coordination in the robot movements, the reference position will be reached in a slower or almost similar way to the centralized case.

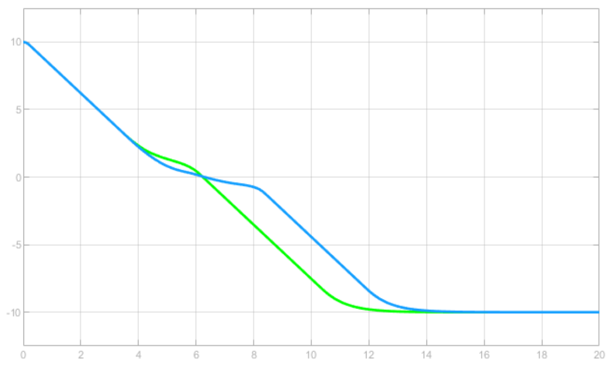

Figure 5. Position of robot 3 along y-axis using centralized (green) vs decentralized (blue) approaches

8.1.3. Neighborhood. In the section we present the results achieved using the notion of neighbours, thus activating the pairwise barriers only if other robots are detected to be within the neighborhood area. This will reduce the number of constraints to enforce, thus reducing the complexity of the QP problems.

In the following plots we can see the robots trajectories in the centralized and decentralized cases using neighborhood approach.

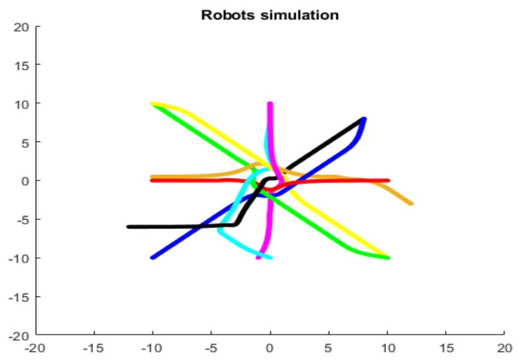

Figure 6. Robots' trajectories in the centralized case using neighbours

In the following plot we can see the barrier between the first and second robot in the centralized case, comparing the neighborhood strategy with the nominal one.

We can see in the neighborhood case that, starting from a specific time instant dependent on robots' relative distances, the pairwise barrier got activated while in the nominal case it is always active.

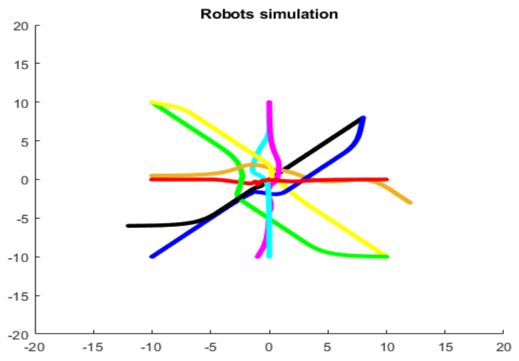

Figure 7. Robots' trajectories in the decentralized case using neighbours

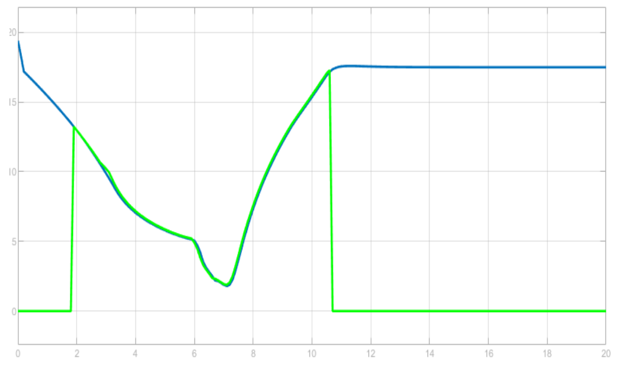

Figure 8. Pairwise barrier values between robot 1 and 2 using nominal (blue) vs neighborhood (green) centralized approaches

8.2. Task2: Robot-driving in an environment with static obstacles

In the second part of the project, the theory on barrier functions has been applied to fulfill two tasks: collision avoidance with static obstacles and leader following; then, the two tasks have been combined together to create a simulation in which three robots follow a leader one in an environment with obstacles.

In *obstacle avoidance*, three robots are put in a environment populated by two static obstacles. The goal is to move the three robots from their starting positions to some goal reference positions assigned by the controller, while avoiding collisions each other through a Centralized CBFs given by the pairwise safety constraint $h_{ij}(\mathbf{p},\mathbf{v}) = \sqrt{2\left(\alpha_i+\alpha_j\right)\left(\|\Delta\mathbf{p}_{ij}\|-D_s\right)} + \frac{\Delta\mathbf{p}_{ij}^T}{\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{v}_{ij}$; moreover, each robot must also avoid a collision with the static obstacles as well: in this case, the safety constraint is the one having expression $\bar{h}_{ik}(\mathbf{p},\mathbf{v}) = \sqrt{2\alpha_i\left(\|\Delta\mathbf{p}_{ik}\|-\left(\frac{D_s}{2}+R_k\right)\right)} + \frac{\Delta\mathbf{p}_{ik}^T}{\|\Delta\mathbf{p}_{ik}\|}\Delta\mathbf{v}_{ik}$. Here, because the obstacles are considered to be static, their velocity v_k is equal to zero. In figure (9) it is represented the schematization of the trajectories the robots have traveled in the environment, by avoiding themselves and the obstacles(in light-blue and purple).

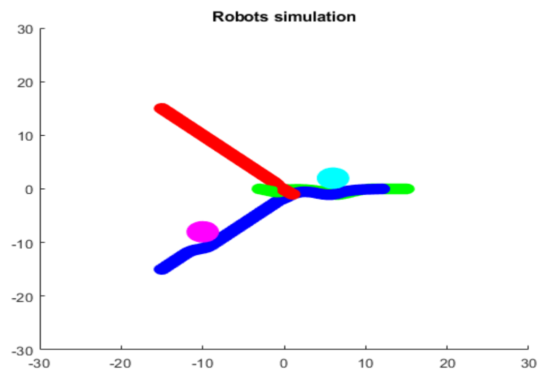

Figure 9. Robots' trajectories in a collision avoidance environment with static obstacles

In figure 10, we can see the comparison between the two different barrier functions approaches for other robots avoidance and for robot-obstacle avoidance.

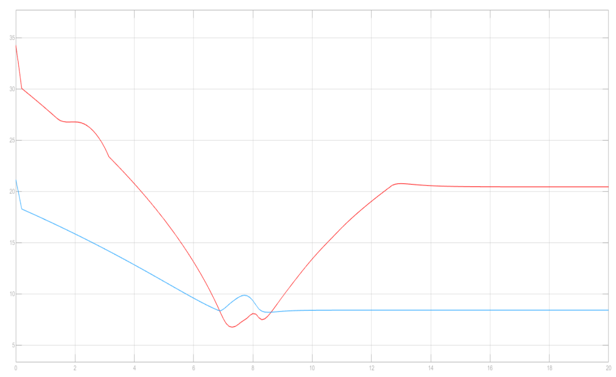

Figure 10. in RED the control barrier functions h12 between robots 1 and 2, in BLUE the control barrier function between robot 1 and obstacle 1

In *leader-follower* approach, we have four robots, in an environment without obstacles, in which one of the robots is the leader, that must reach a given target position and the other three robots are the followers, whose goal is to update their reference position step by step at each time, by following the leader position. In particular, we can say that, given n robots (called followers), each one having position and velocity (p_i, v_i), the goal of the tracking control problem consists in following a fictitious vehicle (called leader), having position and velocity (p_L, v_L). From a control viewpoint, calling (x_f, y_f) the x-y positions of a generic follower robot and (x_L, y_L) the x-y of the leader robot, acting as reference positions for the follower one, the goal is to steer the following quantities to zero: $e_x = x_L - x_f$ and $e_y = y_L - y_f$. In figure (11) it is represented the schematization of the trajectories of three follower robots chasing a leader (in red).

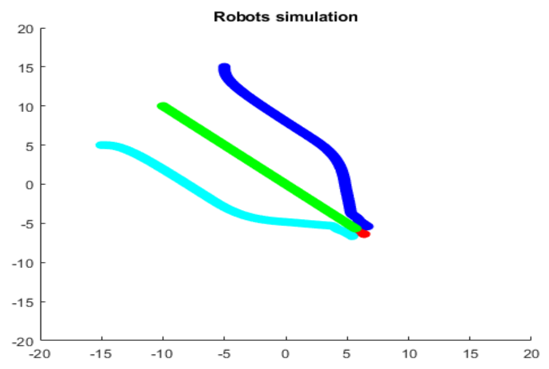

Figure 11. 3 follower robots chasing a leader one

Finally, the two approaches have been combined in order to create a navigation system in which four robots move in an environment with two static obstacles: the leader robot has the goal of reaching a target position, the three followers must chase it while avoiding collision among them and with the obstacles. The results are shown in figure (12).

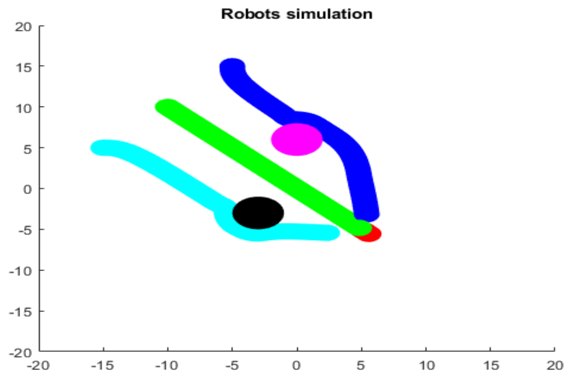

Figure 12. 3 follower robots chasing a leader one while avoiding static obstacles

9. Conclusions

In this project, we have designed some agents subjected to safety barrier certificates, realized by optimization-based controllers and changing the strategies and the tasks of the problems; this approach, even if limited by the use of an approximation of the Keplera robots to double integrators, provides a mathematical solution to multi-agent collision avoidance. A better approach, that is far from the goal of the main reference paper for this project [1], could provide more accurate results by modelling the robots with nonholonomic dynamics. Moreover, some literature improvements in future could manage better the problem of deadlocks.

References

[1] Li Wang, Aaron D. Ames and Magnus Egerstedt *Safety Barrier Certificates for Collisions-Free Multirobot Systems*,IEEE TRANSACTIONS ON ROBOTICS, VOL. 33, NO. 3, JUNE 2017

[2] Li Wang, Aaron D. Ames, and Magnus Egerstedt, *Multi-objective Compositions for Collision-Free Connectivity*

[3] Urs Borrmann, Li Wang, Aaron D. Ames, Magnus Egerstedt, *Control Barrier Certificates for Safe Swarm Behavior*

[4] Mohit Srinivasan and Samuel Coogan, *Control Of Mobile Robots Using Barrier Functions Under Temporal Logic Specifications*, JOURNAL OF LATEX CLASS FILES, VOL. 14, NO. 8, AUGUST 2015

[5] Wenhao Luo,Wen Sun, Ashish Kapoor, *Multi-Robot Collision Avoidance under Uncertainty with Probabilistic Safety Barrier Certificates*

[6] Li Wang, Aaron Ames, and Magnus Egerstedt, *Safety Barrier Certificates for Heterogeneous Multi-Robot Systems*

[7] A. D. Ames, J. W. Grizzles, 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, pp. 6271–6278.