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

Victor Nan Fernandez-Ayala, Xiao Tan and Dimos V. Dimarogonas

***Abstract*— In this work, we propose a distributed control scheme for multi-robot systems in the presence of multiple constraints using control barrier functions. The proposed scheme expands previous work where only one single constraint can be handled. Here we show how to transform multiple constraints to a collective one using a smoothly approximated minimum function. Additionally, human-in-the-loop control is also incorporated seamlessly to our control design, both through the nominal control in the optimization objective as well as a safety condition in the constraints. Possible failure regions are identified and a suitable fix is proposed. Two types of human-in-the-loop scenarios are tested on real multi-robot systems with multiple constraints, including collision avoidance, connectivity maintenance, and arena range limits.**

I. INTRODUCTION

Safety-critical control for dynamical systems has been researched and discussed for a long time. One way to achieve this is through control barrier functions (CBF), initially proposed by [1] and later developed in [2]–[4]. They are a convenient modular design tool based on the concept of set forward invariance, i.e., the system state should always remain in a safe set once it starts inside. This safety is represented as a linear constraint on the input, which is enforced using a computationally efficient implementation that leverages quadratic programs such that a pre-designed nominal controller is minimally modified to satisfy safety requirements. This approach has been widely investigated and applied with practical success.

CBFs are specially useful for the field of multi-agent systems (MAS), where different safety criteria have been explored, like inter-collision avoidance [5]–[8], connectivity maintenance [7], [9], and temporal logic tasks [10]. These are all solved either in a centralized manner [6], [9], where one module has access to the states of all agents, or with a pre-allocation scheme [5], [10] that distributes the linear constraint among the agents involved in a feasible, but non-optimal way.

Several works have been develop to create better distributed approaches to the CBF-induced quadratic program (QP) among agents. In [11], the authors propose an on-line ADMM-based distributed optimization scheme. Other distributed optimization algorithms [12], [13] could also be applied to this problem. These efficiently converge to the optimal solution, but no theoretical guarantees to the satisfaction of the safety conditions can be asserted.

Moreover, [5], [10], [11] assume that each agent has access to (part of) the states of the other agents that share a same coupling constraint. Alternatives [14] that do not assume any specific communication structure are limited and can only obtain an approximate solution. In contrast, in [15] a distributed implementation scheme to obtain the optimal solution to the CBF-induced quadratic program for multi-agent systems with a general connected communication graph is proposed. This is achieved through an equivalent quadratic program solved locally with an auxiliary decision variable, which allows for the optimal solution to be reached in finite time while always satisfying the constraints. The main limitation of this approach is that it can only deal with one single constraint.

Lastly, having a human-in-the-loop (HIL) feature provides additional flexibility, such as handling unexpected situations, detecting and correcting bad behaviours and supporting the automated decision making [16]. Some works combining it with CBFs already exist, like in [17] where CBFs ensure a human-controlled UAV remains inside a safety arena.

In this work, we expand [15] to multiple constraints through the use of the logarithmic approximation of the minimum function. This gives more versatility to the previous method. While some possible failure regions are created due to this transformation, a suitable engineering fix is presented and is demonstrated to work well in the experiments. In addition, human-in-the-loop control can be incorporated seamlessly to the proposed control scheme, both through the nominal control in the optimization objective as well as a safety condition in the constraints, which allows to effectively consider the human as a part of the MAS network.

Applications of this algorithm to the problem of precision agriculture are shown, particularly through two scenarios related with the EU CANOPIES project [18] associated with grape collection in vineyards, where a network of agents must help harvesting and pruning while satisfying collision avoidance constraints between each agent and humans.

II. PRELIMINARIES AND PROBLEM FORMULATION

Consider a multi-agent system with N agents indexed by $\mathcal{I} = \{1, 2, ..., N\}$ and the following system dynamics $\dot{\boldsymbol{x}}_i = \mathfrak{f}_i(\boldsymbol{x}_i) + \mathfrak{g}_i(\boldsymbol{x}_i)\boldsymbol{u}_i$, where the state $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$, and the control input $\boldsymbol{u}_i \in \mathbb{R}^{m_i}$ and $\mathfrak{f}_i(\boldsymbol{x}_i)$, $\mathfrak{g}_i(\boldsymbol{x}_i)$ are locally Lipschitz functions in $\boldsymbol{x}_i$. blk$(\mathfrak{g}_1, \mathfrak{g}_2, ..., \mathfrak{g}_N)$ denotes a block diagonal matrix with diagonal blocks $\mathfrak{g}_1, \mathfrak{g}_2, ..., \mathfrak{g}_N$, which can be either a vector or a matrix.

This work was supported by the the ERC CoG LEAFHOUND, the EU CANOPIES project, the Knut and Alice Wallenberg Foundation (KAW) and the Digital Futures Smart Construction project.

Victor Nan Fernandez-Ayala, Xiao Tan and Dimos V. Dimarogonas are with the Division of Decision and Control Systems, School of EECS, Royal Institute of Technology (KTH), 100 44 Stockholm, Sweden (Email: vnfa, xiaotan, dimos@kth.se).

We denote the stacked state $\boldsymbol{x} := (\boldsymbol{x}_1^\top, \boldsymbol{x}_2^\top, ..., \boldsymbol{x}_N^\top)^\top \in \mathbb{R}^n, n := \sum_{i\in\mathcal{I}} n_i$, the stacked control input $\boldsymbol{u} := (\boldsymbol{u}_1^\top, \boldsymbol{u}_2^\top, ..., \boldsymbol{u}_N^\top)^\top \in \mathbb{R}^m, m = \sum_{i\in\mathcal{I}} m_i$, the stacked vector fields $\mathfrak{f} = (\mathfrak{f}_1^\top, \mathfrak{f}_2^\top, ..., \mathfrak{f}_N^\top)^\top$ and $\mathfrak{g} = \text{blk}(\mathfrak{g}_1, \mathfrak{g}_2, ..., \mathfrak{g}_N)$. Thus, the stacked dynamics is obtained as

$$\dot{\boldsymbol{x}} = \mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}. \tag{1}$$

The communication graph among the MAS is $\mathcal{G} = (\mathcal{I}, E)$, where $\mathcal{I}$ denotes the nodes or agents of the system and E are the edges connecting nodes that share information between themselves. The associated Laplacian matrix [19] representing the connections is denoted as L. In this paper, 1-hop information share will be considered so each agent has access to its own state and the one of its direct neighbours denoted as $N_i := \{j \in \mathcal{I} : (i,j) \in E\}$. The locally obtainable state is $\boldsymbol{x}_{loc,i} := (\boldsymbol{x}_i^\top, \boldsymbol{x}_{j_1}^\top, ..., \boldsymbol{x}_{j_{|N_i|}}^\top)^\top, j_k \in N_i$, for $k \in \{1, 2, ..., |N_i|\}$, i.e., $\boldsymbol{x}_{loc,i}$ stores the states of agent i and all it neighboring agents $j \in N_i$. Originally, a nominal controller using only the locally available state is used. There are several conventional multi-agent coordination algorithms that can be used as a nominal controller, i.e., consensus, coverage, leader-follower, formation control, etc. Since the focus of this work is on the safety-critical part, we will just consider the last one, but the methodology developed works for any of them. For the EU CANOPIES project, the formation controller could be useful for the collaborative manipulation and transportation of a basket full of grapes, where the robots are required to maintain a specific formation to distribute the load.

In particular, in the case of single integrator dynamics with connected communication topology, this controller can be written in stacked form as $\dot{\boldsymbol{x}} = \boldsymbol{u}_{nom} = -L(\boldsymbol{x} - \boldsymbol{x}_d)$, and can also be written as the following

$$\boldsymbol{u}_{nom,i} = -\sum_{j\in N_i} (\boldsymbol{x}_i - \boldsymbol{x}_j - (\boldsymbol{x}_{d,i} - \boldsymbol{x}_{d,j})), \text{ for } i \in \mathcal{I}, \tag{2}$$

where $\boldsymbol{x}_d := (\boldsymbol{x}_{d,1}^\top, \boldsymbol{x}_{d,2}^\top, ..., \boldsymbol{x}_{d,N}^\top)^\top \in \mathbb{R}^n$ and $\boldsymbol{x}_{d,i} \in \mathbb{R}^{n_i}$ are the stacked and agent-wise target state vector for the formation, respectively.

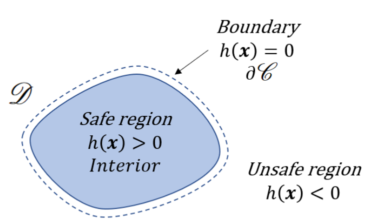


Fig. 1: Geometric representation of the safe region $\mathscr{C}$.

A safety set $\mathscr{C}$, shown in Fig. 1, represents a specific constraint of our design. It is defined as

$$\mathscr{C} = \{\boldsymbol{x} \in \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}, \tag{3}$$

where $h(\boldsymbol{x})$ is a differentiable function. We denote the set $\mathscr{D}$ as a superset of the safety set.

Definition 1 (Extended class $\mathcal{K}$ function)**.** *A continuous, increasing function $\alpha : (-b, a) \longrightarrow [0, \infty)$ with $\alpha(0) = 0$, $a, b > 0$, is called an extended class $\mathcal{K}$ function.*

Definition 2 (CBF)**.** *Let set $\mathscr{C}$ be defined by* (3). *$h(\boldsymbol{x})$ is a control barrier function (CBF) for the stacked system* (1) *if there exists a locally Lipschitz extended class $\mathcal{K}$ function α such that:*

$$\sup_{\boldsymbol{u}\in\mathbb{R}^m} [L_{\mathfrak{f}} h(\boldsymbol{x}) + L_{\mathfrak{g}} h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x}))] \geq 0, \ \forall \boldsymbol{x} \in \mathscr{D} \supset \mathscr{C}, \tag{4}$$

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

Additionally, it has been shown [2], [3] that any locally Lipschitz control input $\boldsymbol{u}$ that satisfies the CBF constraint (4) renders the set $\mathscr{C}$ forward invariant and, if $\mathscr{C}$ is compact, it is also asymptotically stable.

Assumption 1. *Following [15],* (4) *is assumed to be in a form linear in $\boldsymbol{u}$ as shown in* (5).

$$\sum_{i\in\mathcal{I}} \boldsymbol{a}_i^\top(\boldsymbol{x}_{loc,i})\boldsymbol{u}_i + \sum_{i\in\mathcal{I}} b_i(\boldsymbol{x}_{loc,i}) \leq 0, \tag{5}$$

where $\boldsymbol{a}_i \in \mathbb{R}^{m_i}$ and $b_i \in \mathbb{R}$ refer to the agent-wise component of $-L_{\mathfrak{g}} h(\boldsymbol{x})$ and $-L_{\mathfrak{f}} h(\boldsymbol{x}) - \alpha(h(\boldsymbol{x}))$ terms from (4)*, respectively.*

The formulas to compute these terms are to be developed in the next section.

A. Problem formulation

When accounting for multiple safety constraints, a general controller that would always satisfy the constraints is given by the following Quadratic Problem (QP)

$$\begin{aligned} &\min_{\boldsymbol{u}} \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|^2 \\ &s.t. \ \sum_{i\in\mathcal{I}} \boldsymbol{a}_i^{m\top}(\boldsymbol{x}_{loc,i})\boldsymbol{u}_i + \sum_{i\in\mathcal{I}} b_i^m(\boldsymbol{x}_{loc,i}) \leq 0, \\ &\text{for } m = 1, 2, ..., M \end{aligned} \tag{6}$$

where M is the total number of all the safety constraints. Here we adopt a similar form as in (5), which is reasonable considering the common constraints including inter-agent collision avoidance, connectivity maintenance, etc.

B. Distributed Implementation

The aforementioned controller in (6) is based on a centralized approach and therefore needs more than just the locally available state. When dealing with only one constraint, i.e., $M = 1$ in (6), [15] proposed that instead of solving one centralized local QP, the following equivalent QP is to be solved by each individual agent $i \in \mathcal{I}$

$$\begin{aligned} &\min_{\boldsymbol{u}_i\in\mathbb{R}^{m_i}} \|\boldsymbol{u}_i - \boldsymbol{u}_{nom,i}\|^2 \\ &s.t. \ \boldsymbol{a}_i^\top(\boldsymbol{x}_{loc,i})\boldsymbol{u}_i + \sum_{j\in N_i} (y_i - y_j) + b_i(\boldsymbol{x}_{loc,i}) \leq 0, \end{aligned} \tag{7}$$

where the CBF condition only uses the locally available information $\boldsymbol{x}_{loc,i}$ and $\boldsymbol{y} = (y_1, y_2, ..., y_N) \in \mathbb{R}^N$ is an auxiliary variable. If all the local QPs are feasible, the CBF condition (4) is fulfilled for any $\boldsymbol{y}$. Define the following local variable

$$c_i = \frac{1}{\boldsymbol{a}_i^\top \boldsymbol{a}_i}(\boldsymbol{l}_i \boldsymbol{y} + \boldsymbol{a}_i^\top \boldsymbol{u}_{nom,i} + b_i), \tag{8}$$

where $\boldsymbol{l}_i$ is the i-th row of L and $\boldsymbol{y}$ is updated according to the discontinuous adaptative law (9) with positive gain k_0.

$$\dot{\boldsymbol{y}}_i = -k_0 \text{sign}(\sum_{j\in N_i} (\boldsymbol{c}_i - \boldsymbol{c}_j)). \tag{9}$$

Lemma 1. *Assume that $\boldsymbol{a}_i, \boldsymbol{u}_{nom,i}$ and $\boldsymbol{b}_i$ are slowly time-varying and the gain k_0 is large enough. Then $\mathbf{c}$ reaches consensus within a finite time and the optimal $\boldsymbol{y}$ has been found [15].*

Additionally, instead of solving the local QP from (7), the following analytical solution can be used

$$\boldsymbol{u}_i = \boldsymbol{u}_{nom,i} - \max(0, c_i)\boldsymbol{a}_i, \ \forall i \in \mathcal{I}. \tag{10}$$

There are two types of functions identified in [15] whose CBF condition can be written in the form of (7), namely, $h(\boldsymbol{x}) = \sum_{k\in\mathcal{I}} b_k(\boldsymbol{x}_i)$ and $h(\boldsymbol{x}) = \sum_{e\in E} b_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$. We note that $b_k(\boldsymbol{x}_i)$, $b_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are not constraints. In this paper, we assume there are two types of constraints, given below

1) Individual constraints $h_k(\boldsymbol{x}_i)$ that only depend on the state of one agent i.
2) Dual constraints $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ which depend on the state of two neighboring agents.

In the following we will show how these two types of constraints can be transformed into the desired form in [15]. We will show the derivations for the second case since the derivation for the first case follows the same line and is thus omitted. In the experiment both types of constraints are demonstrated.

III. Main result

The method shown in [15] only works when one collective constraint is taken into account. To satisfy more than one constraint, this solution needs to be adapted. Here we will extend the solution to more constraints through the use of the minimum function.

A. Minimum function approximation

For the system to stay safe, all the functions $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ have to be non-negative. Here, to group them all together into one equation, the minimum function can be used, that is, we require $\min_{e\in E} h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0$. However, this does not follow the format from [15]. Instead, observe that

$$\min_{e\in E} h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq -\frac{1}{p}\log\left(\sum_{e\in E} e^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j)}\right), \tag{11}$$

for $(i, j) \in e$, where p is a positive constant relating to the error of the approximation, with higher values meaning less error, but more computationally expensive to calculate due to the exponential. Denoting $h_e(\boldsymbol{x}_i, \boldsymbol{x}_j)$ as h_e for simplicity, one obtains

$$-\frac{1}{p}\log\left(\sum_{e\in E} e^{-ph_e}\right) \geq 0 \Leftrightarrow \log\left(\sum_{e\in E} e^{-ph_e}\right) \leq 0, \tag{12}$$

using the exponential, the condition in (12) is equivalent to

$$\sum_{e\in E} e^{-ph_e} \leq 1 \Leftrightarrow 1 - \sum_{e\in E} e^{-ph_e} \geq 0, \tag{13}$$

and finally, incorporating everything inside the summation sign, we define

$$h(\boldsymbol{x}) = \sum_{e\in E}\left(\frac{1}{|E|} - e^{-ph_e}\right) =: \sum_{e\in E} \tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0, \tag{14}$$

where $|E|$ denotes the cardinality of the set E. Now this satisfies the desired form from [15]. Additionally, the gradient of the CBF needed for the CBF condition (4) is calculated as

$$\nabla h(\boldsymbol{x}) = \sum_{e\in E} pe^{-ph_e}\nabla h_e =: \sum_{e\in E} \nabla\tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{15}$$

This result works for both dual and individual constraints. We note that it is more conservative than just using the minimum function due to the approximation in (11).

Assumption 2. *For both types of constraints, $\alpha(h(\boldsymbol{x}))$ is linear, that is*

$$\alpha(h(\boldsymbol{x})) = a \cdot h(\boldsymbol{x}), \tag{16}$$

where a is a constant positive parameter.

Implementing this solution into (4) gives us

$$\nabla h^T(\boldsymbol{x})(\mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}) + \alpha(h(\boldsymbol{x})) \geq 0, \tag{17}$$

which can further be written as

$$\sum_{e\in E}\nabla\tilde{h}_e^\top(\boldsymbol{x}_i, \boldsymbol{x}_j)\mathfrak{g}\boldsymbol{u} + \sum_{e\in E}\Big(\nabla\tilde{h}_e^\top(\boldsymbol{x}_i, \boldsymbol{x}_j)\mathfrak{f} + \alpha(\tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j))\Big) \geq 0. \tag{18}$$

The final step is to derive a formula for each $\boldsymbol{a}_i$ and b_i in (6) for each agent $i \in \mathcal{I}$.

B. Formulation for dual constraints of the same type

Denote

$$\boldsymbol{a}_{i,dual} = -\sum_{e\in E} I_e(i)\nabla_{\boldsymbol{x}_i}\tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j)\mathfrak{g}_i^\top, \tag{19}$$

$$b_{i,dual} = -\sum_{e\in E} I_e(i)\left(\nabla_{\boldsymbol{x}_i}\tilde{h}_e(\boldsymbol{x}_i, \boldsymbol{x}_j)^\top\mathfrak{f}_i + \frac{a}{2}\left(\frac{1}{\beta} - e^{-ph_e(\boldsymbol{x}_i,\boldsymbol{x}_j)}\right)\right), \tag{20}$$

where in this case $\beta = |E|$ when all functions $\tilde{h}_e$ are of the same type, for example collision avoidance between neighbours in the network. The indicator function $I_e(i)$ is

$$I_e(i) = \begin{cases} 1, & \text{if edge } e \text{ contains } i \\ 0, & \text{if edge } e \text{ does not contain } i \end{cases} \tag{21}$$

Thus, $\boldsymbol{a}_{i,dual}$ and $b_{i,dual}$ only use local information of agent i and its neighbours $j \in N_i$.

C. Formulation for individual constraints of the same type

Following the same derivation, we can obtain the coefficients for individual constraints of the same type. Denote

$$\boldsymbol{a}_{i,ind} = -\sum_{k\in K} \nabla \tilde{h}_k(\boldsymbol{x}_i)\mathfrak{g}_i^\top, \tag{22}$$

$$b_{i,ind} = -\sum_{k\in K}\Big(\nabla \tilde{h}_k(\boldsymbol{x}_i)^\top \mathfrak{f}_i + a\left(\frac{1}{\beta} - e^{-ph_k(\boldsymbol{x}_i)}\right)\Big), \tag{23}$$

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

D. Formulation for multiple constraints

Each type of function h_e and h_k generates a constraint and therefore an $\boldsymbol{a}_i$ and b_i for each agent following (19), (20), (22) and (23). The $\boldsymbol{a}_i$ terms can be directly summed up to get the final formulation, which can be easily seen from the derivation for (14) using a bigger M, representing all the safety constraints in the system, instead of the edge set E

$$\boldsymbol{a}_i = \sum_{l_e\in L_E} \boldsymbol{a}_{i,dual}^{l_e} + \sum_{l_k\in L_K} \boldsymbol{a}_{i,ind}^{l_k}, \tag{24}$$

while for the b_i terms a minor modification is needed since $\beta = M = |L_E||E| + |L_K||\mathcal{I}||K|$ and

$$\boldsymbol{b}_i = \sum_{l_e\in L_E} \boldsymbol{b}_{i,dual}^{l_e} + \sum_{l_k\in L_K} \boldsymbol{b}_{i,ind}^{l_k}, \tag{25}$$

where $\mathcal{I}$ is the set of nodes in the network, $l_e \in L_E$, $l_k \in L_K$, and L_E and L_K are the different type of dual and individual constraints, respectively. For example, if two types of dual constraints are considered, like collision avoidance (CA) and connectivity maintenance (CM), $L_E = \{CA, CM\}$, and if three types of individual constraints are used, like three polygons to be avoided by the robots, $L_K = \{pol_1, pol_2, pol_3\}$.

Now we summarize our main results in the following theorem.

Theorem 1. *Consider the multi-agent system with multiple safety constraints. Assume that Assumptions 1 and 2 hold, together with the assumptions of Lemma 1, and $\boldsymbol{a}_i$ and $\boldsymbol{b}_i$ are computed using* (24) *and* (25)*, respectively. Then, as long as the local QPs given in* (7) *are feasible, i.e., $\sum_{j\in N_i}(y_i - y_j) + b_i(\boldsymbol{x}_{loc,i}) \leq 0$ whenever $\boldsymbol{a}_i = 0, \forall i \in \mathcal{I}$, the multi-agent system will satisfy all the constraints $h_e \geq 0$ and $h_k \geq 0$ in* (11) *for all time.*

Proof. This result follows from (11)-(18). □

Additionally, the distributed nature of the algorithm allows it to scale better than the centralized alternative that considers the whole network.

E. Error region analysis

One basic assumption in [15] is that $\boldsymbol{a}_i(\boldsymbol{x}_{loc,i}) \neq 0$ for all time. However, this may not hold since now there is a summation of several terms in $\boldsymbol{a}_i$, which means that there are certain regions where $\boldsymbol{a}_i = 0$ and $\sum_{j\in N_i}(y_i - y_j) + b_i(\boldsymbol{x}_{loc,i}) > 0$. Therefore, for those cases, the CBF condition (5) would be violated.

Define the "error region" as

$$\mathcal{R} = \{\boldsymbol{x} : \boldsymbol{a}_i(\boldsymbol{x}_{loc,i}) = 0 \text{ for some } i \in \mathcal{I}\}. \tag{26}$$

Nevertheless, this region has zero volume as it is only composed of individual points in the problem space. Therefore, these points are almost never reached, except when one initializes the problem in those points. To resolve this, a simple distributed implementation fix is proposed, that is, to set $y_i = 0$ when the agent i is in region $\mathcal{R}$ and let $\boldsymbol{u}_{nom,i}$ drive it away from $\mathcal{R}$. In practice, due to the zero volume characteristic of $\mathcal{R}$, it can almost always be guaranteed to work.

F. Adding HIL

Two human-in-the-loop features will be considered. For the first one, one of the agents is controlled by a human operator together with the nominal controller in (2), that is

$$\boldsymbol{u}_{mix,i} = \boldsymbol{u}_{HIL} + \boldsymbol{u}_{nom,i}, \tag{27}$$

$\boldsymbol{u}_{mix,i}$ will then be passed through the distributed QP (7) as the new nominal controller to obtain the final controller. For the second feature, the HIL element will be a human within workspace. For example, a person could enter into the experiment's arena and interact with the agents in some way, like herding/chasing the robots. In this case, the human behaviour can be considered as an external agent fully controlled by the human.

Combining the different safety conditions with a HIL feature, the CBF-enabled mixer module shown in Fig. 2 is created.

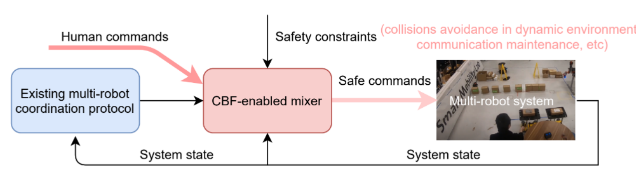


Fig. 2: CBF-enabled mixer module from (7), (24) and (25).

IV. Experiments

Two testing scenarios have been created related to the EU CANOPIES project, where the lab setting has been adapted to simulate an artificial vineyard, as shown in Fig. 3, where our network of robots must work in a collaborative manner while satisfying some basic safety constraints. These are, remaining inside the vineyard/arena, avoiding inner obstacles, inter-collision avoidance and connectivity maintenance between robots, and avoidance with a human worker in the same environment.

Fig. 3: Artificial vineyard lab setting.

The robots used for the experiments are the Nexus 4WD holonomic robots, shown in Fig. 4, using either a NVIDIA Jetson TX2 or Intel NUC as onboard computer running a standard Ubuntu 18.04 distribution with ROS Melodic.

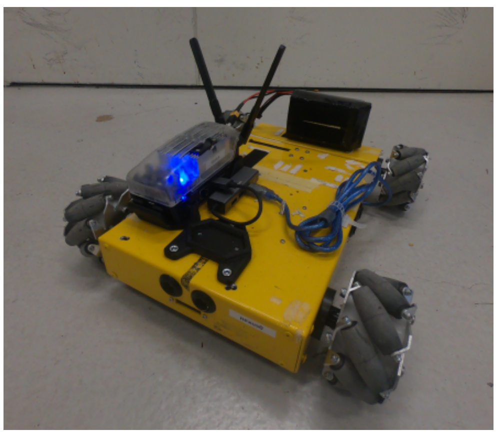


Fig. 4: Nexus 4WD robot with a NVIDIA Jetson TX2.

The robots receive their position (state) through the use of markers and the Qualisys motion capture system. A diagram of this process is shown in Fig. 5.

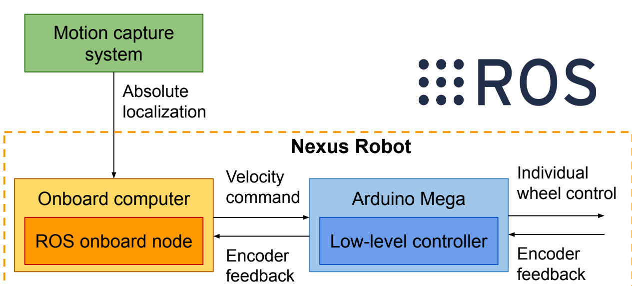


Fig. 5: Diagram of the communications for each Nexus.

Five Nexus have been used for the experiments, following the communication topology in Fig. 6. The systems dynamics are modelled as a simple integrator $\dot{\boldsymbol{x}} = \boldsymbol{u}$, which is equivalent to (1) with $\mathfrak{f}(\boldsymbol{x}) = 0_{10}$ and $\mathfrak{g}(\boldsymbol{x})$ being the identity matrix $\mathbb{I}_{10}$.

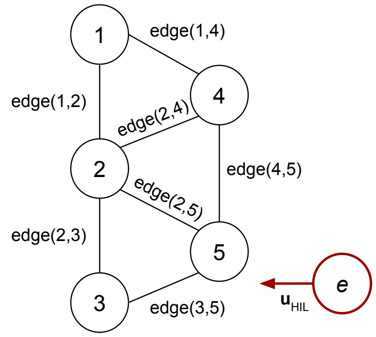


Fig. 6: Network created by 5 agents and extra human e.

A. Formation controller with an extra human

In this first case, the nominal controller is designed such that the robots stay together in the specific formation shown in Fig. 6, while the safety constraints explained at the beginning of this section are always maintained. In particular, the distributed formation controller from (2) is used for each agent as the nominal controller. This could, for example, represent a swarm of logistic robots that enter the vineyard in a close formation to collaboratively carry an empty basket. Thereafter, they get loaded with grapes and take them back to a storage area while avoiding human workers fulfilling other tasks.

To create the single combined constraint, we first analyze each type separately. For **dual constraints**, taking the case of connectivity maintenance, each edge creates a different constraint, which implies the CBF function is the following

$$h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) = r^2 - ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 \geq 0, \tag{28}$$

where r is the maximum allowed safety distance, $||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = (x_i - x_j)^2 - (y_i - y_j)^2$ and $\forall (i,j) \in E$.

We approximate the seven dual constraints as in (14) with $|E| = 7$. The gradient shown in (15) is used, with the specific gradient of the connectivity maintenance condition being

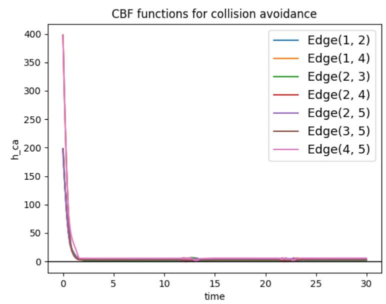
$$\nabla h_e(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{bmatrix} \frac{\partial h_e}{\partial x_i} \\ \frac{\partial h_e}{\partial y_i} \\ \frac{\partial h_e}{\partial x_j} \\ \frac{\partial h_e}{\partial y_j} \end{bmatrix} = \begin{bmatrix} -2(x_i - x_j) \\ -2(y_i - y_j) \\ 2(x_i - x_j) \\ 2(y_i - y_j) \end{bmatrix}, \tag{29}$$

for each dual constraint pair. Using (19) and (20) we can compute the $\boldsymbol{a}_{i,dual}$ and $b_{i,dual}$ terms for the QP. Similar terms can be developed for inter-collision avoidance constraints.

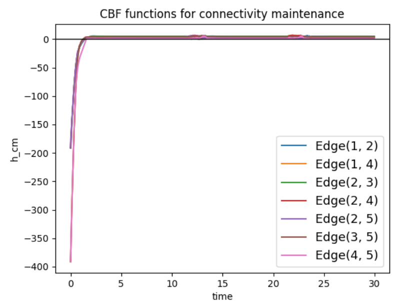
For **individual constraints**, we consider the rectangular arena limits for each agent, that is, there is one constraint per side ($|K| = 4$ and $|I| = 5$) and the corresponding functions become $h_k^{x_{\max}} = x_{\max} - x_i$, $h_k^{x_{\min}} = x_i - x_{\min}$, $h_k^{y_{\max}} = y_{\max} - y_i$ and $h_k^{y_{\max}} = y_i - y_{\min}$; with the gradient

$$\nabla h_i = \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{30}$$

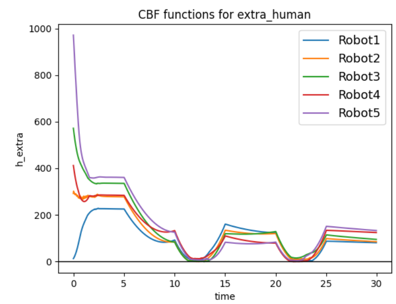
Computing the $\boldsymbol{a}_{i,ind}$ and $b_{i,ind}$ terms using (22) and (23), allows to obtain the final single constraint variables $\boldsymbol{a}_i$ and b_i from (24) and (25).



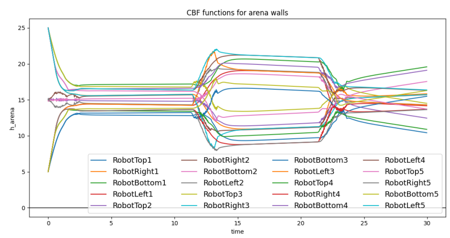
(a) CBFs h_{ca} for collision avoidance.



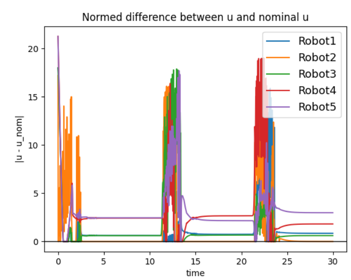
(b) CBFs h_{cm} for connectivity maintenance.



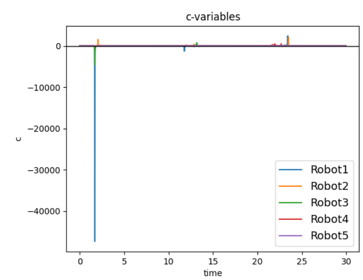
(c) CBFs h_{extra} for avoidance with human.



(d) CBFs h_{arena} for each side of the rectangular arena.



(e) Normed difference $\|\boldsymbol{u}(t) - \boldsymbol{u}_{nom}\|$.



(f) Auxiliary variable c (8).

Fig. 7: Evolution of CBFs for collision avoidance (a), connectivity maintenance (b), arena (d) and avoidance with human (c), as well as the normed difference of the control input (e) and the consensus of c (f) for the first experiment scenario.

Collision avoidance with an extra human is similar to the one between robots, but adapted for individual type constraints and therefore, with a modification to account for the speed and trajectory of the human. Since the gradient computed only takes into account the robot state $\boldsymbol{x}_i$ and not the one from the human $\boldsymbol{x}_e$, we need to add this extra term to b_i, such that

$$\boldsymbol{a}_{i,extra} = -\nabla_{\boldsymbol{x}_i} \tilde{h}(\boldsymbol{x}_i, \boldsymbol{x}_e) \mathfrak{g}_i^\top, \tag{31}$$

$$b_{i,extra} = -\left(\nabla_{\boldsymbol{x}_i} \tilde{h}(\boldsymbol{x}_i, \boldsymbol{x}_e)^\top \mathfrak{f}_i + a \left(\frac{1}{|I|} - e^{-p h_e(\boldsymbol{x}_i, \boldsymbol{x}_e)} \right) \right) - \nabla_{\boldsymbol{x}_e} \tilde{h}(\boldsymbol{x}_e, \boldsymbol{x}_i) \mathfrak{g}_i^\top. \tag{32}$$

which can then be added to (24) and (25) as a standard individual type constraint $l_k = (extra)$.

The results for a random execution of this first case can be seen in Fig. 7 and a video of it can be found in [20]. Four different types of CBFs are shown, *(a)* for the collision avoidance between robots, *(b)* for the connectivity maintenance between robots, *(c)* for the collision avoidance between each robot and the human and *(d)* for bounding the robots inside the four walls of the arena (Top, Right, Bottom and Left). All of them are positive for all time and therefore the conditions are fulfilled, even when initialized from a non-satisfying point, that is when $h_e < 0$ and/or $h_k < 0$, since the agents quickly reposition to be safe. Additionally, *(e)* shows the difference between the nominal controller and the output of the QP and *(f)* shows the evolution of the c variable from (8).

B. Mixed initiative control with human commands

This second experimental scenario is very similar with the first one, since the safety constraints and initial nominal formation controller are the same and the single constraint is constructed in the same manner. The difference is the use of mixed initiative control (27) to direct one of the robots through human commands (instead of having a human worker in the same environment). With this human command, we can directly control the movement of one of the agents, and therefore of the whole network thanks to its communication graph in Fig. 6. Additionally, it shows the versatility of the method and the capability of working for different cases. Results for this second case are omitted since the CBF evolution shows a similar response to the first case, but a video of this experimental scenario can also be found in [20].

V. Conclusion

In this work, a distributed implementation for CBF-induced QPs for multi-agent systems has been developed to allow for more than one dual (and/or individual) constraints through the use of the minimum function approximation. This allows for each agent to solve its local QP while still remaining always safe. Additionally, the regions where the algorithm might fail were analyzed and a correction to fix this problem was proposed. Two application cases related to the precision agriculture problem of grape collection inside the EU CANOPIES project were considered to show the behaviour of the novel controller and validate the performance and safety of the proposed implementation.

REFERENCES

[1] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings*, vol. 40, no. 12, pp. 462–467, 2007.

[2] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," in *Proc. IFAC Conf. Anal. Design Hybrid Syst.*, vol. 48, 2015, pp. 54–61.

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

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

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

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

[7] D. Panagou, D. M. Stipanović, and P. G. Voulgaris, "Distributed coordination control for multi-robot networks using lyapunov-like barrier functions," *IEEE Transactions on Automatic Control*, vol. 61, no. 3, pp. 617–632, 2015.

[8] M. Jankovic, M. Santillo, and Y. Wang, "Multi-agent systems with cbf-based controllers – collision avoidance and liveness from instability," *arXiv*, 07 2022. [Online]. Available: https://arxiv.org/abs/2207.04915

[9] B. Capelli and L. Sabattini, "Connectivity maintenance: Global and optimized approach through control barrier functions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 5590–5596.

[10] L. Lindemann and D. V. Dimarogonas, "Barrier function based collaborative control of multiple robots under signal temporal logic tasks," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1916–1928, 2020.

[11] Y. Chen, M. Santillo, M. Jankovic, and A. D. Ames, "Online decentralized decision making with inequality constraints: an ADMM approach," *IEEE Control Systems Letters*, vol. 5, no. 6, pp. 2156–2161, 2020.

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

[13] M. Santilli, G. Oliva, and A. Gasparri, "Distributed finite-time algorithm for a class of quadratic optimization problems with time-varying linear constraints," in *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 4380–4386.

[14] M. Manao and I. Masumi, "Consensus-based control barrier function for swarm," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

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

[16] W. Li, D. Sadigh, S. Sastry, and S. Seshia, "Synthesis for human-in-the-loop control systems," in *Tools and Algorithms for the Construction and Analysis of Systems*, vol. 8413, 04 2014, pp. 470–484.

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

[18] CANOPIES project, "A Collaborative Paradigm for Human Workers and Multi-Robot Teams in Precision Agriculture Systems," in *https://www.project-canopies.eu/*, european Commission under the H2020 Framework Programme. [Online]. Available: https://www.project-canopies.eu/.

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

[20] V. Nan Fernandez-Ayala, X. Tan, and D. V. Dimarogonas. Distributed barrier function-enabled human-in-the-loop control for multi-robot systems. Youtube. [Online]. Available: https://youtu.be/2ASw5bkWa1I