Automatica 179 (2025) 112393

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Distributed safe control design and probabilistic safety verification for multi-agent systems ☆

Han Wang *, Antonis Papachristodoulou, Kostas Margellos

OX1 3PJ, Department of Engineering Science, University of Oxford, Oxford, United Kingdom

ARTICLE INFO

Article history:
Received 17 January 2024
Received in revised form 10 March 2025
Accepted 20 April 2025
Available online 7 June 2025

Keywords:
Distributed control
Safe control
Multi-agent systems
Scenario approach
Nonlinear systems

ABSTRACT

We propose distributed iterative algorithms for safe control design and safety verification for networked multi-agent systems. These algorithms rely on distributing a control barrier function (CBF) related quadratic programming (QP) problem assuming the existence of CBFs. The proposed distributed algorithm addresses infeasibility issues of existing schemes via a cooperation mechanism between agents. The resulting control input is guaranteed to be optimal, and satisfies CBF constraints of all agents. Furthermore, a truncated algorithm is proposed to facilitate computational implementation. The performance of the truncated algorithm is evaluated using a distributed safety verification algorithm. The algorithm quantifies safety for multi-agent systems probabilistically by means of CBFs. Both upper and lower bounds on the probability of safety are obtained using the so called scenario approach. Both the scenario sampling and safety verification procedures are fully distributed. The efficacy of our algorithms is demonstrated by an example on multi-robot collision avoidance.

1. Introduction

Safety of a dynamical system requires the system state to remain in a safe set for all time. This property is important in many applications such as collision avoidance (Ding, He, Ren, Wang, & Zheng, 2022; Wang, Li, Yu, He, & Guan, 2019), vehicle platooning (Alam, Gattami, Johansson, & Tomlin, 2014; Axelsson, 2016), vehicle merging control (Xiao & Cassandras, 2021), etc. For a single agent system, safety is usually captured by introducing constraints on the state of the agent and the environment. For a multi-agent system, the meaning of safety extends to capture the interactions among agents. In this case, safety is encoded by coupling constraints over the states of a group of agents. For a networked multi-agent system, where agents cooperate to satisfy safety constraints, we consider designing distributed algorithms to ensure safety for all agents.

Another problem of interest is to validate the proposed control law. For a single agent system, an agent can evaluate the system behaviour to characterize its risk of being unsafe under the employed control input. Similarly, for a multi-agent safety verification problem, cooperation among agents is necessary since safety involves multiple agents. In summary, this paper focuses on designing a distributed protocol for safe control input design and developing a distributed safety verification algorithm.

1.1. Related work

Safety in control systems is often certified by control barrier functions (CBF), which is a type of control Lyapunov-like functions (Ames, Xu, Grizzle, & Tabuada, 2016; Primbs, Nevistić, & Doyle, 1999; Sontag, 1989). By enforcing the inner product of the CBF derivative and vector field of the controlled system to be bounded, safety is rigorously guaranteed at any time. CBF is shown to be powerful and scalable in control input design for control-affine systems, as this condition can be encoded as a linear constraint in a quadratic programming (QP) problem (Ames et al., 2016). By solving online QP problems at every state, the system is guaranteed to be safe (Ames, Grizzle, & Tabuada, 2014; Hsu, Xu, & Ames, 2015). Most of the existing results in this direction involve a centralized approach; however, multi-agent considerations call for distributed solution regimes.

CBF-based distributed algorithms have been proposed in Borrmann, Wang, Ames, and Egerstedt (2015), Chen, Singletary, and Ames (2020), Wang, Ames, and Egerstedt (2017). In these papers, the CBF constraints are decomposed and allocated to neighbouring agents to facilitate a distributed implementation. Under the assumption that each local optimization problem is feasible, the overall CBF constraints are satisfied. However, this assumption is

☆ Antonis Papachristodoulou was supported in part by the EPSRC Programme Grant EEBio (EP/Y014073/1). Han Wang was supported by the EPSRC IAA Technology Fund. Kostas Margellos would like to acknowledge partial support by MathWorks. The material in this paper was partially presented at the 62nd IEEE Conference on Decision and Control (CDC) December,13-15, 2023, Marina Bay Sands, Singapore. This paper was recommended for publication in revised form by Associate Editor Dimitra Panagou under the direction of Editor , Christos G. Cassandras.

* Corresponding author.
E-mail addresses: han.wang@eng.ox.ac.uk (H. Wang), antonis@eng.ox.ac.uk (A. Papachristodoulou), kostas.margellos@eng.ox.ac.uk (K. Margellos).

https://doi.org/10.1016/j.automatica.2025.112393
0005-1098/

usually much stronger than that of feasibility of the nominal centralized problem. Moreover, optimality of the nominal centralized problem by the distributed controller is not guaranteed. An improved constraint sharing mechanism is developed in Xu (2018), where the CBF constraints are dynamically tuned for feasibility, but for single-agent systems. Optimality is further considered in Tan and Dimarogonas (2021) , but for multi-agent systems with only one CBF constraint. A dynamical constraint allocation scheme among agents based on a consensus protocol is proposed. In our work, we deal with the problem of guaranteeing feasibility of local problems across iterations while preserving optimality, under multiple CBF safety constraints. In essence, the distributed CBF-based safe control design problem can be seen under the lens of distributed optimization.

Distributed optimization for a multi-agent system aims to design a distributed protocol that involves solving an optimization problem locally for every agent. Algorithms can be divided into two types, dual decomposition (Duchi, Agarwal, & Wainwright, 2011; Falsone, Margellos, Garatti, & Prandini, 2017; Falsone, Notarnicola, Notarstefano, & Prandini, 2020; Shi, Ling, Yuan, Wu, & Yin, 2014) and primal decomposition-based (Camisa, Farina, Notarnicola, & Notarstefano, 2021; Li, Feng, & Xie, 2020; Margellos, Falsone, Garatti, & Prandini, 2017; Nedic, Ozdaglar, & Parrilo, 2010; Notarnicola & Notarstefano, 2019). Dual decomposition methods consider the dual problem, where each agent maintains a local copy of the dual variables. Constraint satisfaction is achieved by consensus over the dual variables. Primal decomposition methods directly decompose the primal problem into local problems. By local projection (Li et al., 2020; Margellos et al., 2017; Nedic et al., 2010) or updating auxiliary variables (Camisa et al., 2021; Notarnicola & Notarstefano, 2019), algorithms converge to centralized optimum under convexity assumptions. Such methods guarantee near feasibility as far as the constraints of the primal problem are concerned. As our problem has similar structure as the one considered in Camisa et al. (2021), Notarnicola and Notarstefano (2019), primal decomposition structure is applied to develop our algorithm.

To reduce the communication and computation burden, a truncation mechanism is proposed to allow us to terminate the algorithm before reaching convergence. To give a probabilistic guarantee for safety over the state space, we leverage scenario approach (Calafiore & Campi, 2005, 2006; Campi & Garatti, 2008, 2018; Garatti & Campi, 2019), which samples a number of independent states from the state space and enforces the constraint only at these realizations.

1.2. Contributions

Our contributions can be summarized as follows:

(1) We provide a distributed algorithm for designing safe controllers for multi-agent systems. Under the assumption of the existence of feasible CBFs, a centralized safe control design problem is formulated. Our distributed algorithm parallelizes computation by decomposing the centralized problem into local problems, while guaranteeing feasibility of every local problem across iterations. The optimal solution returned by our algorithm is guaranteed to be the same as that of the nominal centralized problem, therefore satisfying all the CBF constraints.
(2) In view of practical implementation, and since the convergence guarantees of the proposed algorithm are asymptotic, we propose a truncation mechanism for early termination. This comes at the cost of sacrificing strict guarantees of satisfying the CBF constraints, however, it reduces the communication and computation burden of an asymptotic algorithm. Moreover, it is accompanied with a verification scheme that provides probabilistic guarantees on safety constraint violation.
(3) The proposed verification scheme can be applied more generally to verify safety for multi-agent systems. In particular, instead of verifying safety over the whole state–space, which is challenging for multi-agent systems, we propose a scenario-based verification algorithm for a probabilistic quantification of safety by means of satisfying CBF constraints. A sequential sampling algorithm is proposed to sample scenarios efficiently in a distributed fashion. We accompany our solution with a probabilistic safety certificate; to achieve this, we extend the state-of-the-art result (Garatti & Campi, 2019, Theorem 1) to a multi-agent setting. Both lower and upper bounds on the probability of violating CBF constraints are established, while the safety verification program is also shown to be amenable to parallelized computation.

1.3. Organization

Section 3 proposes our distributed safe control design algorithm, including a truncated version and the associated mathematical analysis. Section 4 provides the distributed safety verification scheme, and the distributed scenario sampling algorithm. Section 5 demonstrates the control design and safety verification algorithms on a multi-robot system collision avoidance case study. Section 6 concludes the paper and provides some directions for future research.

2. Preliminaries

2.1. Notation

We use $\mathbb{R}$, $\mathbb{R}^N$ to represent the set of one-dimensional, and N-dimensional real numbers, respectively. A continuous function $\alpha(\cdot) : (-b, a) \to (-\infty, +\infty)$ is said to be an extended class-$\mathcal{K}$ function for positive a and b, if it is strictly increasing and $\alpha(0) = 0$. $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes a graph with a nodes set $\mathcal{V}$ and an edge set $\mathcal{E}$. Boldface symbols are used as stacked vectors for scalar or vector elements, e.g., $\boldsymbol{x} = [x_1^\top, \dots, x_N^\top]^\top$. For matrices $g_1, \dots, g_N$, $\text{diag}(g_1, \dots, g_N)$ denotes the corresponding block diagonal matrix. I is an identity matrix, with its dimension being clear from the context. For a set $\mathcal{K}$, $|\mathcal{K}|$ denotes its cardinality. For a set $\mathcal{S}$, $\text{Int}(\mathcal{S})$ denotes the interior.

2.2. Control barrier functions

Consider a nonlinear control-affine system

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

with $x(t) \in \mathcal{X} \subset \mathbb{R}^n$, $u(t) \in \mathcal{U} \subset \mathbb{R}^m$, $f(x) : \mathcal{X} \to \mathbb{R}^n$, and $g(x) : \mathcal{X} \to \mathbb{R}^{n\times m}$. Both f and g are further assumed to be locally Lipschitz continuous on a compact set $\mathcal{X} \subset \mathbb{R}^n$. We denote by $x(u(\cdot), t, x_0)$ the state of the system at time t starting from x_0, under a local Lipschitz continuous control law $u(\cdot)$.

The safe set $\mathcal{S}$ is represented by the zero-super level set of a function $s(x)$. Dually, the unsafe set $\bar{\mathcal{S}}$ can be defined as the complementary set. With this formulation, the safe control design problem boils down to finding $u(\cdot) \in \mathcal{U}$, such that $s(x(u(\cdot), t, x_0)) \geq 0$ for any t. To achieve this, a control barrier function-based quadratic programming approach was proposed (Ames et al., 2016).

Definition 2.1. For the control-affine dynamical system (1), a continuously differentiable function $b(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is said to be a control barrier function, if there exists an extended class-$\mathcal{K}$ function $\alpha(\cdot)$, such that for any $x \in \mathcal{B}$,

$$\sup_{u\in\mathcal{U}}[\mathcal{L}_f b(x) + \mathcal{L}_g b(x)u + \alpha(b(x))] \geq 0. \tag{2}$$

Here $\mathcal{L}_f b(x)$ and $\mathcal{L}_g b(x)$ are Lie derivatives, which are defined by $\mathcal{L}_f b(x) := \frac{\partial b(x)}{\partial x} f(x)$ and $\mathcal{L}_g b(x) := \frac{\partial b(x)}{\partial x} g(x)$, respectively.

Given a control barrier function $b(x)$, the control admissible set corresponding to (2) is defined by

$$K_{cbf}(x) := \{u \in \mathcal{U} : \mathcal{L}_f b(x) + \mathcal{L}_g b(x)u + \alpha(b(x)) \geq 0\}. \tag{3}$$

Theorem 1 (*Ames et al. (2016, Corollary 2)*)**.** *Consider a control barrier function $b(x)$. Then for any $x \in \mathcal{B}$, any locally Lipschitz continuous controller $u(x)$ such that $u(x) \in K_{cbf}(x)$ will render the set $\mathcal{B}$ forward invariant.*

2.3. Scenario optimization

Robust optimization offers a methodology to immunize decisions against uncertainty. An uncertain optimization problem is formulated as

$$\begin{aligned} &\min_{z\in\mathcal{Z}} c^\top z \\ &\text{subject to } z \in \mathcal{Z}_x, \text{ for all } x \in \mathcal{H}, \end{aligned} \tag{4}$$

where $z \in \mathbb{R}^n$ is a decision variable constrained by a set $\mathcal{Z} \subseteq \mathbb{R}^n$ and, $c \in \mathbb{R}^n$ is a constant vector. The uncertain constraint set $\mathcal{Z}_x$ is parameterized by an uncertain parameter x, which is a random variable defined on a probability space $(\mathcal{H}, \mathcal{F}, \mathbb{P})$. Even in the case where $\mathcal{Z}_x$ is convex for any $x \in \mathcal{H}$, if the uncertain parameters' domain $\mathcal{H}$ is continuous or even unknown, the robust optimization problem is usually hard (or even impossible) to solve. The so-called scenario approach, on the other hand, proposes to solve the problem over finite empirical records, named *scenarios*, and accompany the resulting solution with probabilistic guarantees on its feasibility properties. The corresponding scenario optimization problem can be formulated as

$$\begin{aligned} &\min_{z\in\mathcal{Z}} c^\top z \\ &\text{subject to } z \in \bigcap_{r=1,\ldots,R} \mathcal{Z}_{x^{(r)}}, \end{aligned} \tag{5}$$

where $x^{(r)}$, $r = 1, \ldots, R$ are *scenarios* sampled independently from the set $\mathcal{H}$. If $\mathcal{Z}_x$ is convex for any $x \in \mathcal{H}$, the scenario optimization (5) is a convex optimization problem which can be solved efficiently.

Definition 2.2 (*Violation Probability*)**.** The violation probability of a given $z \in \mathcal{Z}$ is defined as $V(z) = \mathbb{P}\{x \in \mathcal{H} : z \notin \mathcal{Z}_x\}$.

Clearly, the optimal solution of (5) satisfies $z^* \in \bigcap_{r=1,\ldots,R} \mathcal{Z}_{x^{(r)}}$, but is not necessarily within $\mathcal{Z}_x$ for an arbitrary new $x \in \mathcal{H}$. i.e., we do not necessarily have $V(z^*) = 0$. In fact, z^* is itself a random variable as it depends on the choice of the scenarios $x^{(r)}$, $r = 1, \ldots, R$. To align with our subsequent developments, we will characterize $V(z^*)$ for a slightly more general scenario program; to this end, consider the following scenario optimization problem with relaxed constraints:

$$\begin{aligned} &\min_{z\in\mathcal{Z},\xi^{(r)}\geq 0, r=1,\ldots,R} c^\top z + \rho \sum_{r=1}^{R} \xi^{(r)} \\ &\text{subject to } h(z, x^{(r)}) \leq \xi^{(r)}, r = 1, \ldots, R, \end{aligned} \tag{6}$$

where $x^{(r)}, r = 1, \ldots, R$ are independently sampled from $(\mathcal{H}, \mathcal{F}, \mathbb{P})$. Notice that here we consider the explicit characterization of the constraint set $\mathcal{Z}_{x^{(r)}}$ through functions $h(z, x^{(r)})$, $r = 1, \ldots, R$.

A constraint $z \in \mathcal{Z}_{x^{(r)}}$ is called a support constraint if its removal (while the other constraints are maintained) changes the solution z^*. We impose the following assumption.

Assumption 2.1 (*Garatti and Campi (2019, Assumption 2)*)**.** Consider problem (6) and assume that a unique optimal solution $(z^*, \{\xi^{*,(r)}\}_{r=1}^R)$ exists almost surely with respect to the choice of $\{x^{(r)}\}_{r=1}^R$. We further assume that the optimal solution $(z^*, \{\xi^{*,(r)}\}_{r=1}^R$ of (6) coincides almost surely with respect to the choice of the scenarios $x^{(r)}$, $r = 1, \ldots, R$ with the solution that is obtained after eliminating all the constraints that are not of support.

The violation probability $V(z^*) = \mathbb{P}\{x \in \mathcal{H} : f(z^*, x) > 0\}$ can be then characterized by the following theorem.

Theorem 2 (*Garatti and Campi (2019, Theorem 4)*)**.** *Consider the optimization problem (6). Suppose that its optimal solution $(z^*, \{\xi^{*,(r)}\}_{r=1}^R$ satisfies Assumption 2.1. Given a confidence parameter $\beta \in (0, 1)$, for any $k = 0, 1, \ldots, R-1$ consider the polynomial equation in the t variable*

$$\begin{aligned} &\binom{R}{k} t^{R-k} - \frac{\beta}{2R} \sum_{j=k}^{R-1} \binom{j}{k} t^{j-k} \\ &- \frac{\beta}{6R} \sum_{j=R+1}^{4R} \binom{j}{k} t^{j-k} = 0, \end{aligned} \tag{7}$$

and for $k = R$ consider the polynomial equation

$$1 - \frac{\beta}{6R} \sum_{i=R+1}^{4R} \binom{j}{k} t^{j-R} = 0. \tag{8}$$

For any $k = 0, \ldots, R-1$, (7) has exactly two solutions in $[0, +\infty)$, which we denote with $\underline{t}(k)$ and $\bar{t}(k)$ ($\underline{t}(k)) \leq \bar{t}(k)$. Instead, (8) has only one solution in $[0, +\infty)$, which we denote with $\bar{t}(R)$, while we define $\underline{t}(R) = 0$. Let $\underline{\epsilon}(k) := \max\{0, 1 - \bar{t}(k)\}$ and $\bar{\epsilon}(k) := 1 - \underline{t}(k)$, $k = 0, 1, \ldots, R$. We then have that

$$\mathbb{P}^R\{\underline{\epsilon}(s^*) \leq V(z^*) \leq \bar{\epsilon}(s^*)\} \geq 1 - \beta, \tag{9}$$

where s^* is the number of $x^{(r)}$'s for which $h(z^*, x^{(r)}) \geq 0$.

3. Distributed safe control law

Consider an N-agent system with the dynamics of the ith agent described by

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \tag{10}$$

where $x_i(t) \in \mathcal{X}_i \subset \mathbb{R}^{n_i}$ denotes its state, $u_i \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ denotes its control input, and $\mathcal{U}_i$ is a convex set. The dynamics $f_i(x_i) : \mathcal{X}_i \to \mathbb{R}^{n_i}$ and $g_i(x_i) : \mathcal{X}_i \to \mathbb{R}^{n_i} \times \mathbb{R}^{m_i}$ are both locally Lipschitz-continuous on a compact set $\mathcal{X}_i \subset \mathbb{R}^{n_i}$, which represents the domain of each agent. Vector $\boldsymbol{x} = [x_1^\top, \ldots, x_N^\top]^\top$ stacks the states of all systems, $\boldsymbol{u} = [u_1^\top, \ldots, u_N^\top]^\top$ stacks the control inputs, while $f(\boldsymbol{x}) = [f_1(x_1)^\top, \ldots, f_N(x_N)^\top]^\top$, $g(\boldsymbol{x}) = \text{diag}(g_1(x_1), \ldots, g_N(x_N))$ stack the dynamics for each agent. The domain and control admissible set for the multi-agent system are then defined by

$$\mathcal{X} := \prod_{i=1}^{N} \mathcal{X}_i, \quad \mathcal{U} := \prod_{i=1}^{N} \mathcal{U}_i,$$

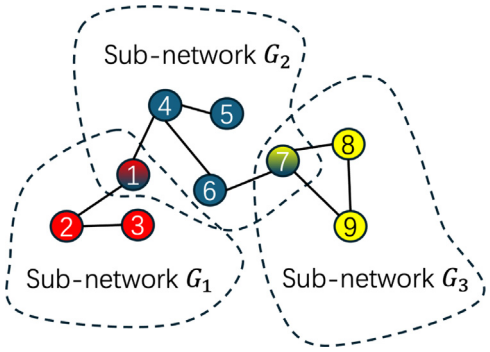


Fig. 1. Pictorial illustration of a connected network $\mathcal{G}$ with 9 agents, where agents 1, 2, and 3 form the sub-network $\mathcal{G}_1$ with safe set $\mathcal{S}_1$, agents 1, 4, 5, 6 and 7 form the sub-network $\mathcal{G}_2$ with a safe set $\mathcal{S}_2$, and agents 7, 8, and 9 form the sub-network $\mathcal{G}_3$ with a safe set $\mathcal{S}_3$. The set of agents in each sub-network is given by $\mathcal{V}_1 = \{1, 2, 3\}$, $\mathcal{V}_2 = \{1, 4, 5, 6, 7\}$, and $\mathcal{V}_3 = \{7, 8, 9\}$. It can be observed that agent 1 belongs to two sub-networks, $\mathcal{G}_1$ and $\mathcal{G}_2$, and thus $\mathcal{C}_1 = \{1, 2\}$. Similarly, agent 2 belongs only to $\mathcal{G}_1$, and agent 7 belongs to both $\mathcal{G}_2$ and $\mathcal{G}_3$, giving $\mathcal{C}_2 = \{1\}$ and $\mathcal{C}_7 = \{2, 3\}$.

where $\prod$ represents the Cartesian product for the state space of all the agents. Given that all $\mathcal{X}_i$, $i = 1, \dots, N$, are assumed to be compact, compactness of $\mathcal{X}$ is assured using Tychonoff's theorem (Wright, 1994). In this way, the system dynamics of the whole multi-agent system can be compactly modelled by $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$.

The networked system is described by an undirected and connected graph $\mathcal{G}$, with nodes set $\mathcal{V} = \{1, \dots, N\}$, and edges set $\mathcal{E}$ such that $\{i, j\} \in \mathcal{E}$ if agent j communicates with agent i. Agents are partitioned in E sub-networks with specific safety requirement. For each subgraph $\mathcal{G}_e$, $e \in \{1, \dots, E\}$, the set of agents is $\mathcal{V}_e \subseteq \mathcal{V}$. Let $\boldsymbol{x}_e = [x_i^\top]_{i\in\mathcal{V}_e}^\top$ be the stacked states in the eth sub-network. Each agent i can communicate and cooperate with its neighbour $j \in \mathcal{N}_i$ to stay safe inside sub-network e by ensuring

$$\boldsymbol{x}_e(t) \in \mathcal{S}_e := \{\boldsymbol{x}_e : s_e(\boldsymbol{x}_e) \geq 0\}, \ \forall t \geq 0, \tag{11}$$

where $s_e(\cdot) \in \mathbb{R}$. Define $\mathcal{S} := \prod_{i=1}^E \mathcal{S}_e$, and let $\mathcal{C}_i$ denote the set of indices representing the safety constraints associated with agent $i \in \{1, \dots, N\}$. Specifically, for a given agent $i \in \{1, \dots, N\}$, $\mathcal{C}_i$ contains all indices e for which constraint e applies to agent i. Given that each safety constraint involves a sub-network of agents, $\mathcal{C}_i$ also describes the set of indices of sub-networks that agent i belongs to. As a result, agent i belongs to sub-networks $\mathcal{G}_e$, $e \in \mathcal{C}_i$. Fig. 1 illustrates pictorially the relationship between $\mathcal{V}_e$ and $\mathcal{C}_i$.

Assumption 3.1. For each $e = 1, \dots, E$, sub-network $\mathcal{G}_e$ is connected and undirected.

Connectivity allows communication among agents in every sub-network $\mathcal{G}_e$, $e \in \{1, \dots, E\}$. Agents in $\mathcal{G}_e$ are then able to cooperatively design a controller $\boldsymbol{u}_e(\boldsymbol{x})$ for safety, satisfying $s_e(\boldsymbol{x}_e) \geq 0$, for all $e = 1, \dots, E$.

Assumption 3.2. Given sets $\mathcal{X}$ and $\mathcal{S}_e$, $e = 1, \dots, E$, we assume there exist control barrier functions $b_e(\cdot)$, such that $\mathcal{B}_e := \{\boldsymbol{x}_e : b_e(\boldsymbol{x}_e) \geq 0\} \subseteq \mathcal{S}_e$, $e = 1, \dots, E$. Define $\mathcal{B} := \prod_{e=1}^E \mathcal{B}_e$, and $\mathcal{H} := \mathcal{B} \cap \mathcal{X}$. We further assume that $\text{Int}(\mathcal{H}) \neq \emptyset$.

Assumption 3.2 directly implies that $\text{Int}(\mathcal{S}) \neq \emptyset$ and $\text{Int}(\mathcal{B}) \neq \emptyset$. This is essential for using CBF methods to design safe controllers. However, checking emptiness of these sets is a challenging task. When $s_e(\boldsymbol{x}_e)$, $b_e(\boldsymbol{x}_e)$, $e = 1, \dots, E$, are polynomial functions, and $\mathcal{X}$ is defined by polynomial functions as well, emptiness can be checked via sum-of-squares programming. We refer the reader to Parrilo (2000) for further details.

Assumption 3.3. Consider the multi-agent system (10) and CBFs $b_e(\boldsymbol{x}_e)$, $e = 1, \dots, E$, and class-$\mathcal{K}$ functions $\alpha_{ie}(\cdot)$, $i = 1, \dots, N$, $e \in \mathcal{C}_i$. For every $\boldsymbol{x} \in \mathcal{B}$, we assume there exists a locally Lipschitz $\boldsymbol{u} = [u_1^\top \in \mathcal{U}_1, \dots, u_N^\top \in \mathcal{U}_N]^\top \in \mathcal{U}$, such that for any $e \in \{1, \dots, E\}$:

$$\sum_{i\in\mathcal{V}_e} \left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right) \geq 0. \tag{12}$$

The summation in (12) follows from applying the chain rule and considering the partial derivative of $b_e(\boldsymbol{x}_e)$ with respect to the state x_i of every agent $i \in \mathcal{V}_e$.

Assumption 3.3 guarantees the existence of one controller $\boldsymbol{u}$ that satisfies all CBF constraints. This property is also known as the control sharing property (Xu, 2018, Definition 2). CBFs that satisfy Assumptions 3.2 and 3.3 can be designed using sum-of-squares programming (Schneeberger, Dörfler, & Mastellone, 2023).

Following Ames et al. (2016, Theorem 3), safety constraints can be incorporated in the CBF-QP formulation given by

$$\begin{aligned} J^* = \min_{u_i\in\mathcal{U}} & \sum_{i=1}^N \|u_i - u_i^{\text{des}}(x_i)\|_2^2 \\ \text{s.t.} & \sum_{i\in\mathcal{V}_e} \left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right) \geq 0, \\ & \forall e \in \{1, \dots, E\}, \end{aligned} \tag{13}$$

where $\alpha_{ie}(\cdot)$'s are class-$\mathcal{K}$ functions, and hence also $\sum_{i\in\mathcal{V}_e} \alpha_{ie}(\cdot)$ is also a class-$\mathcal{K}$. $u_i^{\text{des}}(x_i)$ is a nominal stabilizing control input.

The CBF constraints in (13) are defined on the control inputs for multiple agents. If every agent regards the variables of other agents as stationary, (13) decomposes to a family of problems, one for each $i = 1, \dots, N$,

$$\begin{aligned} & \min_{u_i\in\mathcal{U}_i} \|u_i - u_i^{\text{des}}(x_i)\|_2^2 \\ & \text{s.t.} \frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \geq 0, \forall e \in \mathcal{C}_i. \end{aligned} \tag{14}$$

Under Assumptions 3.2 and 3.3, (13) is guaranteed to be feasible, but feasibility of (14) is not ensured for every $i \in \{1, \dots, N\}$. In this work, we propose an improved distributed framework for solving (13) with guaranteed feasibility.

Let

$$\begin{aligned} J_i(u_i) &= \|u_i - u_i^{\text{des}}(x_i)\|_2^2, \\ h_{ie}(u_i) &= -\left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right). \end{aligned} \tag{15}$$

We then have the following *safety* results.

Proposition 3.1 (*Tan and Dimarogonas (2022, Proposition 1)*)**.** *Consider Assumptions 3.2, 3.3. Let $\boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$ be the optimal solution of (13). Suppose $\boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$ is locally Lipschitz continuous for every $\boldsymbol{x} \in \mathcal{B}$, then the set $\mathcal{B}$ is forward invariant under the vector field $f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$.*

Remark 3.1. Local Lipschitz continuity of $\boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$ is important for forward invariance of $\mathcal{B}$ under the vector field $f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$. This can be guaranteed if the CBF constraints are linearly independent, and there are no input constraints. For more general cases, (strong) forward invariance can be guaranteed for a discontinuous vector field, under certain regularity conditions on the different CBFs. Interested readers are referred to Isaly, Ghanbarpour, Sanfelice, and Dixon (2024), Usevitch, Garg, and Panagou (2020), and Garg et al. (2024, Section 9) for a comprehensive review. As this is tangential to the focus of our work, we will concentrate on the distributed implementation of the QP induced from multi-CBFs (13).

Notice that, even not shown explicitly, $h_{ie}(u_i)$ depends on x_i, $i \in \mathcal{V}_e$. We also highlight that (13) is parameterized in $\boldsymbol{x}$, which can be thought of as constant as for the optimization problem in (13) is concerned. Under Assumptions 3.2, 3.3, problem (13) is always feasible for all $x \in \mathcal{B}$. To begin with our analysis, we propose a relaxed version of (13) to guarantee feasibility of the local problems in the proposed distributed algorithm. This will be clarified in the sequel.

$$\begin{aligned} H^* &= \min_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{\rho}\geq 0} H(\boldsymbol{u},\boldsymbol{\rho}) \\ &:= \sum_{i=1}^{N}\left\{J_i(u_i)+\sum_{e\in\mathcal{C}_i}(\rho_{ie}^2+M_i\rho_{ie})\right\} \\ \text{subject to } &\sum_{i\in\mathcal{V}_e} h_{ie}(u_i) \leq \sum_{i\in\mathcal{V}_e}\rho_{ie},\ \forall e\in\{1,\dots,E\}. \end{aligned} \tag{16}$$

Feasibility of problem (16) is clear, as the positive variable $\boldsymbol{\rho}$ relaxes the linear constraints.

In view of an optimality analysis, we further impose the following constraint qualification assumption.

Assumption 3.4. For every $\boldsymbol{x} \in \mathcal{B}$, there exists $\boldsymbol{u}(\boldsymbol{x})\in \mathcal{U}$, such that $\sum_{i\in\mathcal{V}_e} h_{ie} < 0$ for all $e = 1,\dots,E$.

Assumption 3.4 ensures strong duality for the nominal problem (13). As a result, there also exists $\boldsymbol{u}(\boldsymbol{x}) \in \mathcal{U}$ and $\boldsymbol{\rho} = 0$, such that $\sum_{i\in\mathcal{V}_e} h_{ie} < \sum_{i\in\mathcal{V}_e}\rho_{ie}$, for all $e \in \{1,\dots,E\}$. This demonstrates strong duality for the relaxed problem (16).

Optimality is analysed in the following lemma.

Lemma 3.1. *Consider Assumptions 3.2, 3.3 and 3.4. Denote the minimizers of (13) and (16), by $\boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$ and $\left(\boldsymbol{u}^*_{\text{rel}}(\boldsymbol{x}),\boldsymbol{\rho}^*\right)$, respectively. Let $\tilde{\boldsymbol{\mu}}^*$ be an optimal dual variable associated with the CBF constraint in (13). If*

$$M_i \geq \tilde{\mu}^*_e,\ \forall i\in\mathcal{V}_e,\ \forall e\in\{1,\dots,E\}, \tag{17}$$

where $\tilde{\mu}^*_e$ is the eth element of $\tilde{\boldsymbol{\mu}}^*$, then $\boldsymbol{u}^*_{\text{rel}}(\boldsymbol{x}) = \boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$, $\boldsymbol{\rho}^* = 0$, and $\tilde{\boldsymbol{\mu}}^*$ is also an optimal dual solution of (16).

Proof. See the Appendix. □

Lemma 3.1 establishes a lower bound for $M_i, i = 1,\dots,N$, under which the optimal primal–dual solution of (16) coincides with that of (13). The lower bound is determined by the optimal dual solution $\tilde{\boldsymbol{\mu}}$ of the unrelaxed problem (13). Under Assumption 3.4, $\tilde{\boldsymbol{\mu}}^*$ is also bounded following Nedić and Ozdaglar (2009, Lemma 1). In practice, one can select a large enough $M_i, i = 1,\dots,N$ to satisfy (17).

3.1. Full control law

We now design an algorithm to solve the centralized CBF-QP problem (13) in a distributed manner with guaranteed feasibility across iterations; see Algorithm 1.

Since $h_{ie}(u_i)$ also depends on x_l for $l \in \mathcal{V}_e\backslash\{i\}$, an additional communication round at the beginning of the algorithm is designed. For all $i = 1,\dots,N$, and $e \in \mathcal{C}_i$, agent i is to receive x_l from agent $l \in \mathcal{N}_i\cap\mathcal{V}_e$. Within a finite number of communication rounds, agent i can gather all the other agents' states in subnetworks $e \in \mathcal{C}_i$. Then, for any $e \in \mathcal{C}_i$, functions $h_{ie}(u_i)$ can be constructed as in (15).

There are two main computation and two communication steps in the algorithm. At the first computation step (Step 3), agent i solves the optimization problem 18 to obtain the optimal primal–dual solution $((u_i^k,\boldsymbol{\rho}_i^k),\boldsymbol{\mu}_i^k)$, where $\boldsymbol{\rho}_i$ includes relaxation variables denoted by ρ_{ie} (penalized in the cost by M_i), and $\boldsymbol{\mu}_i$ includes the dual variables μ_{ie}, for all $e \in \mathcal{C}_i$. In practice, μ_{ie} corresponds to the constraints allocated to agent i, i.e. $h_{ie}(x_i) + \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e}(\lambda_{il}^k - \lambda_{li}^k) \leq \rho_{ie}$. Moreover, the constraints in the distributed problem 18 are relaxed by an additional non-negative relaxation variable ρ_{ie}. This guarantees the feasibility of the local optimization problem. However, this does not necessarily imply satisfaction of the CBF constraints in (13) by using $\boldsymbol{u}^{k+1}$.

The first computation step uses auxiliary variables λ_{il}^k and λ_{li}^k. The difference $\lambda_{il}^k - \lambda_{li}^k$ constitutes estimates of the neighbouring terms $h_{le}(u_l)$. λ_{il}^0 is initialized arbitrarily. As we will show in Theorem 3, the initialization will not influence convergence to the optimizer. Among all these variables, λ_{le}^k for $l \in \mathcal{N}_i\cap\mathcal{V}_e$ are updated and stored by neighbours. They are available to agent i via communication at Step 2. The second computation step is to update the local auxiliary variables at Step 5. Part of the dual variables used in the update are received from the neighbours at Step 4. Here the update is a gradient-like procedure, with stepsize $\gamma^k > 0$.

Algorithm 1 Distributed Safe Control Design Algorithm for agent i at x_i

Initialization Arbitrary $\lambda_{il}^0, \forall l \in \mathcal{N}_i\cap\mathcal{V}_e, \forall e \in \mathcal{C}_i$.
Receive x_l for any $l \in \mathcal{N}_i\cap\mathcal{V}_e, e \in \mathcal{C}_i$
Send x_i to any $l \in \mathcal{N}_i\cap\mathcal{V}_e$, for $e \in \mathcal{C}_i$.
Output: Optimal control input u_i^*

1: **while** Not reaching convergence **do**
2: **Receive** λ_{il}^k from $\forall l \in \mathcal{N}_i\cap\mathcal{V}_e, \forall e \in \mathcal{C}_i$.
3: **Solve** $((u_i^k,\boldsymbol{\rho}_i^k),\boldsymbol{\mu}_i^k)$ as a primal-dual solution of the following optimization problem

$$\begin{aligned} &\min_{u_i,\boldsymbol{\rho}_i} J_i(u_i)+\sum_{e\in\mathcal{C}_i}(\rho_{ie}^2+M_i\rho_{ie}) \\ &\text{s.t. } u_i\in\mathcal{U}_i,\ \rho_{ie}\geq 0, \\ &\quad h_{ie}(u_i)+\sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e}(\lambda_{il}^k-\lambda_{li}^k)\leq\rho_{ie},\forall e\in\mathcal{C}_i. \end{aligned} \tag{18}$$

4: **Receive** μ_{le}^k from agent $l \in \mathcal{N}_i\cap\mathcal{V}_e$.
5: **Update** λ_{il} by

$$\lambda_{il}^{k+1} = \lambda_{il}^k - \gamma^k(\mu_{ie}^k - \mu_{le}^k). \tag{19}$$

6: **end while**

Remark 3.2. Algorithm 1 capitalizes on the primal-decomposition algorithm in Notarnicola and Notarstefano (2019, Algorithm RSDD), however, with several key extensions. First, the relaxation penalty in the cost includes a new quadratic term. This renders the cost function strongly convex, allowing for superior convergence properties and ensuring uniqueness of the minimizer across iterations. Moreover, for every agent $i \in \{1,\dots,N\}$, each CBF constraint $e \in \mathcal{C}_i$ is relaxed by an individual relaxation variable ρ_{ie}. On the contrary, Notarnicola and Notarstefano (2019, Algorithm RSDD) uses one relaxation variable for all the constraints. Multiple relaxation variables enable stricter satisfaction of CBF constraints across iterations. This is especially important when a particular $\rho_{ie_1}^k$ is significantly larger than the other ones $\rho_{ie_2}^k, e_2 \in \mathcal{C}_i\backslash e_1$. It should also be noted that Algorithm 1 is applicable to the case where $\mathcal{G}$ is divided into several sub-networks $\mathcal{G}_e, e \in \{1,\dots,E\}$, while Notarnicola and Notarstefano (2019, Algorithm RSDD) only deals with a single network. This becomes of importance for multi-agent applications where safety constraints are typically defined on several sub-networks.

Among different types of distributed optimization algorithms, primal-decomposition methods, firstly proposed by Notarnicola

and Notarstefano (2019, Algorithm RSDD) is selected here for its ability to guarantee almost-safety across iterations. This is realized by allocating the auxiliary variables $\boldsymbol{\lambda}$, while balancing the safety requirement to every agent. We say "almost" here since additional relaxation variables are introduced in every local optimization problem for feasibility. In applications that require high control frequency, the algorithm may stop before reaching convergence. When the relaxation variables $\boldsymbol{\rho}^k = 0$ for a given $k > 0$, then for any $e \in \{1, \dots, E\}$ we have that

$$\sum_{i\in\mathcal{V}_e} h_{ie}(u_i^k) = \sum_{i\in\mathcal{V}_e} \underbrace{\left\{ h_{ie}(u_i^k) + \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right\}}_{\leq 0} \leq 0,$$

which implies that the CBF constraints are satisfied with $\boldsymbol{u}^k$. The next theorem gives the convergence result.

Theorem 3. *Consider Assumptions 3.1, 3.2, 3.3, 3.4, and let $M_i \geq \mu_e$ for every $i = 1, \dots, N$, $e \in \mathcal{C}_i$. For every agent $i = 1, \dots, N$, and any bounded $\boldsymbol{\lambda}^0$,*

(a) if $\mathcal{U}_i \subset \mathbb{R}^{m_i}$. Choose the sequence $\{\gamma^k\}_{k\geq 0}$, with each $\gamma^k > 0$, and $\sum_{k=0}^{\infty} \gamma^k = \infty$, $\sum_{k=0}^{\infty} (\gamma^k)^2 < \infty$. Then we have $\lim_{k\to\infty} H(\boldsymbol{u}^k, \boldsymbol{\rho}^k) - J^* \to 0$, and $\boldsymbol{u}^k$ converges to the primal optimal solution of (13).

(b) if $\mathcal{U}_i = \mathbb{R}^{m_i}$, and for every $e \in \{1, \dots, E\}$, $\sum_{i\in\mathcal{V}_e} h_{ie}(u_i)$ are linearly independent in $\boldsymbol{u}$. Let the step size $\gamma^k = \gamma > 0$ be a small constant. $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k)$ converges to the optimal cost J^* in (13) sublinearly, i.e. $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k) - J^* \leq \frac{2\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_2^2}{\gamma^k}$, and $\boldsymbol{u}^k$ converges to the primal optimal solution of (13).

Proof. See the Appendix. □

Given that (13) is guaranteed to be feasible under Assumption 3.3, the optimal controller designed by Algorithm 1 is guaranteed to satisfy all the CBF constraints. However, this does not necessarily hold for $\boldsymbol{u}^k(\boldsymbol{x})$ with arbitrary k, if $\boldsymbol{\rho}^k(\boldsymbol{x}) \neq 0$. However, terminating the algorithm early, and considering $\boldsymbol{u}^k(\boldsymbol{x})$ at the time of termination has many benefits in terms of reducing computation and communication complexity. This motivates the analysis of a truncated algorithm as presented in the next section.

3.2. Truncated control law

Algorithm 1 can be implemented in a distributed fashion with ensured safety and optimality properties, however, it may not be suitable for control tasks that require high control frequency, i.e. multi-robot system control, as its theoretical properties are established in an asymptotic manner. This motivates the use of a *truncated algorithm*, Algorithm 2, where the algorithm terminates after a finite number of iterations, denoted by η.

Algorithm 2 Truncated Distributed Safe Control Design Algorithm for agent i

Initialization Predefined $\lambda_{il}^0, \forall l \in \mathcal{N}_i \cap \mathcal{V}_e, \forall e \in \mathcal{C}_i$, truncated parameter $\eta \in \mathbb{N}$
Receive x_l for any $l \in \mathcal{N}_i \cap \mathcal{V}_e$,$e \in \mathcal{C}_i$
Send x_i to any $l \in \mathcal{N}_i \cap \mathcal{V}_e$, $e \in \mathcal{C}_i$
Output: Optimal control input u_i^*
1: **while** $k \leq \eta$ **do**
2: steps 2, 3, 4 in Algorithm 1
3: step 5 in Algorithm 1
4: **end while**

Algorithm 2 is computationally more efficient compared to Algorithm 1, at the cost of potentially violating the control barrier function constraints. The violations are reflected in the non-zero relation variables $\boldsymbol{\rho}^\eta(\boldsymbol{x})$. In general, it is challenging to provide an explicit bound for η, under which $\boldsymbol{\rho}^\eta(\boldsymbol{x}) = 0$, as the distributed algorithm converges asymptotically as per Theorem 3. Moreover, $\boldsymbol{\rho}^\eta(\boldsymbol{x})$ depends on the state $\boldsymbol{x} \in \mathcal{H} := \mathcal{X} \cap \mathcal{B}$, which parameterizes the optimization problem (16). To quantify safety of the multi-agent system (10) with $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}^\eta(\boldsymbol{x})$, we study the problem of safety verification by means of CBFs. This is established in the following section.

4. Distributed safety verification

In this section we show how to verify safety for a multi-agent system for any $\boldsymbol{x} \in \mathcal{H}$, using the truncated controller $\boldsymbol{u}^\eta(\boldsymbol{x})$ designed by Algorithm 2. The verification is conducted by checking the *risk* of becoming unsafe along the current trajectories by means of CBFs. We would like to measure the violations of the CBF constraints for the multi-agent system (10), under the control law $\boldsymbol{u}^\eta(\boldsymbol{x})$. However, this problem becomes challenging as $\text{Int}(\mathcal{H}) \neq \emptyset$, and one would need to verify a safety property for an uncountable number of points. Instead of verifying this for any $\boldsymbol{x} \in \mathcal{H}$, we propose to verify over *finite* scenarios, i.e. samples of $\boldsymbol{x}$, from $\mathcal{H}$. Notice that the multi-agent system under consideration (see (10)) is deterministic; however, we draw scenarios as a discrete approximation of $\mathcal{H}$. The scenario approach (Garatti & Campi, 2019) then provides the theoretical foundation for quantifying the probability that the solution that satisfies our safety property for a finite number of scenarios, satisfies this property when it comes to yet another realization of $\boldsymbol{x} \in \mathcal{H}$. Such a generalization property is in turn probabilistic, with a probability measure implicitly defined using the mechanism employed to draw scenarios (see Section 4.2).

We note here the analysis conducted in this section can be applied to, but not limited to the controller designed using Algorithm 2. The only requirement for the verified controller $\boldsymbol{u}(\boldsymbol{x})$ is locally Lipschitz continuous, which is necessary for the solution of the multi-agent system to be unique. We also highlight that in this section a CBF is only regarded as a verification criterion but not necessarily as a control design principle.

4.1. Scenario based safety verification

Consider an N-agent system (10) and a safe invariant set $\mathcal{B}$. Our objective is to verify whether all the CBF constraints are satisfied for the multi-agent system (10) using $\boldsymbol{u}(\boldsymbol{x})$, for any $\boldsymbol{x} \in \mathcal{H}$. A new set $\mathcal{Z}_{\boldsymbol{x}}$ is introduced to represent the satisfaction of all the CBF constraints.

$$\mathcal{Z}_{\boldsymbol{x}} := \left\{ \boldsymbol{z} : \sum_{i\in\mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x})) \leq \sum_{i\in\mathcal{V}_e} z_{ie}, \forall e \in \{1, \dots, E\} \right\}. \tag{20}$$

Here, $\boldsymbol{z} := [z_{ie}]$, $\forall e \in \{1, \dots, E\}$, $\forall i \in \mathcal{V}_e$. Then, if $0 \in \mathcal{Z}_{\boldsymbol{x}}$, $\forall \boldsymbol{x} \in \mathcal{H}$, we conclude that all CBF constraints are satisfied using $\boldsymbol{u}(\boldsymbol{x})$, for any $\boldsymbol{x} \in \mathcal{H}$. With a slight abuse of notation, we define $\mathcal{Z}_{\boldsymbol{x}}^i$ as

$$\mathcal{Z}_{\boldsymbol{x}}^i := \left\{ \boldsymbol{z} : \sum_{k\in\mathcal{V}_e} h_{ke}(u_k(\boldsymbol{x})) \leq \sum_{k\in\mathcal{V}_e} z_{ke}, \forall e \in \mathcal{C}_i \right\} \tag{21}$$

to represent the satisfaction of CBF constraints that involve agent i, for every $i \in \{1, \dots, N\}$. Here, $u_k(\boldsymbol{x})$ denotes the control input of agent k. If $0 \in \mathcal{Z}_{\boldsymbol{x}}^i$, $\forall \boldsymbol{x} \in \mathcal{H}$, the CBF constraints that involve agent i are satisfied using $\boldsymbol{u}(\boldsymbol{x})$. Conversely, if $0 \notin \mathcal{Z}_{\boldsymbol{x}}^i$, at least one CBF constraint that involves agent i is violated, for some $\boldsymbol{x} \in \mathcal{H}$. Therefore, $\mathcal{Z}_{\boldsymbol{x}}$ can be expressed as

$$\mathcal{Z}_{\boldsymbol{x}} = \bigcap_{i=1}^{N} \mathcal{Z}_{\boldsymbol{x}}^i. \tag{22}$$

We propose a scenario-based safety verification program as follows.

$$\begin{aligned}\min_{\boldsymbol{z}\geq 0,\boldsymbol{\zeta}\geq 0} \ & \sum_{i=1}^{N}\sum_{e\in\mathcal{C}_i}\left(z_{ie}^2+H_i\sum_{r=1}^{R}\zeta_{ie}^{(r)}\right)\\ \text{s.t.} \ & \sum_{i\in\mathcal{V}_e}h_{ie}(u_i(\boldsymbol{x}^{(r)}))\leq\sum_{i\in\mathcal{V}_e}(z_{ie}+\zeta_{ie}^{(r)}),\\ & \forall e\in\{1,\dots,E\},\forall r\in\{1,\dots,R\},\end{aligned}\tag{23}$$

where scenarios $\boldsymbol{x}^{(r)}\in\mathcal{H}$ for any $r=1,\dots,R$ are extracted according to some probability distribution to be clarified in the sequel. Throughout the section $\bar{X}=\{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(R)}\}$ denotes the set of scenarios, where $\boldsymbol{x}^{(r)}=[(x_1^{(r)})^\top,\dots,(x_N^{(r)})^\top]^\top\in\mathbb{R}^{\sum_{i=1}^N n_i}$, for $r=1,\dots,R$, and R is the number of scenarios. Relaxation variables $\boldsymbol{\zeta}$ are introduced, while $H_i>0$ is a penalty coefficient for every $i\in\{1,\dots,N\}$. Let $(\boldsymbol{z}^*(\boldsymbol{x}),\boldsymbol{\zeta}^*(\boldsymbol{x}))$ denote the optimal solution of (23). In the sequel, we drop the dependency of $\boldsymbol{x}$ for simplicity.

Program (23) is a data-driven QP, where all the constraints are linear based on the samples. If for any scenario $\boldsymbol{x}^{(r)}$, $r=1,\dots,R$, and the corresponding control input $\boldsymbol{u}(\boldsymbol{x})$, all the CBF constraints are satisfied, then $\boldsymbol{\zeta}^*=0$. Conversely, $\boldsymbol{\zeta}^*\neq 0$ represents a CBF constraint violation, and indicates the risk of being unsafe by means of CBF, up to level $\boldsymbol{z}^*$. Following Definition 2.2, the violation probability for (23) is defined by

$$V(\boldsymbol{z}):=\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}\notin\mathcal{Z}_{\boldsymbol{x}}\right\}.\tag{24}$$

Then, $V(\boldsymbol{z}^*)=\mathbb{P}\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}\}$ represents the probability that at least one CBF constraint is violated up to $\boldsymbol{z}^*$, for any $\boldsymbol{x}\in\mathcal{H}$. Our goal is to *distributedly* characterize the violation probability $V(\boldsymbol{z}^*)$ using a *finite* number of scenarios, i.e. samples of $\boldsymbol{x}$ from $\mathcal{H}$.

4.2. Sampling the scenarios

The scenarios are sampled independently from the set $\mathcal{H}$. For sampling we define a probability density $\pi(\boldsymbol{x})$ associated with set $\mathcal{H}$ that satisfies $\int_{\mathcal{H}}\pi(\boldsymbol{x})d\boldsymbol{x}=1$. One typical choice of $\pi(\boldsymbol{x})$ is to set it according to the density of the uniform distribution, i.e., $\pi(x)=\pi^{\text{uni}}(\boldsymbol{x})=\frac{1}{\int_{\mathcal{H}}d\boldsymbol{x}}$.

The existence of $\pi^{\text{uni}}(\boldsymbol{x})$ is assured as $\mathcal{H}$ is a non-empty and compact set, due to Assumption 3.2. Then, $\boldsymbol{x}$ can be sampled R times independently from the distribution $\pi^{\text{uni}}(\boldsymbol{x})$. Note that the choice of the probability distribution does not affect the probabilistic results established in the sequel due to the distribution-free nature of the scenario approach (Garatti & Campi, 2019, Section 3.1). Although the uniform distribution here is well-defined, the set $\mathcal{H}$ is defined implicitly as the intersection of multiple sets. Sampling a point from the proposed uniform distribution is rather arduous in practice, and every agent may not have access to $\mathcal{H}$. Here, we provide a sequential algorithm to sample scenarios $\boldsymbol{x}^{(r)}$, $r=1,\dots,R$.

The algorithm constructs the densities from which samples are extracted sequentially for each agent. We first define the sets from which samples are extracted for agent i with part of the states of agents in the same sub-network $\mathcal{G}_e$ fixed.

$$\mathcal{H}_{ie}=\begin{cases}\mathcal{X}_i, & \text{if } \exists l\in\mathcal{V}_e, \text{such that } l>i\\ \{x_i\in\mathcal{X}_i|b_e(x_i,\{x_l^{(r)}\})\geq 0\}, & \text{otherwise}\end{cases}\tag{25}$$

Let $\mathcal{H}_i:=\bigcap_{e\in\mathcal{C}_i}\mathcal{H}_{ie}$. The parameters in (25) can all be collected by local communication, since only states of agents in the same sub-network are required.

In Step 1, the first scenario $x_1^{(r)}$ associated with Agent 1 is sampled from distribution $\pi_1=\frac{1}{\int_{\mathcal{X}}dx}$, since now there are no other agents involved to restrict the set for Agent 1. Then, the sampling-construction procedures repeat sequentially from Agent 2 to Agent N. For $i=2,\dots,N$, before sampling the scenario $x_i^{(r)}$, we first check whether $\mathcal{H}_i$ is empty (Step 4). If $\mathcal{H}_i=\emptyset$ (Step 5), then we return to the sampling-construction of agent $i-F$, $F\neq 1$ to avoid a deadlock on step i. The deadlock happens when for given scenarios $x_1^{(r)},\dots,x_{i-2}^{(r)}$, the set $\mathcal{H}_{i-1}$ is such that for any $x_{i-1}^{(r)}\in\mathcal{H}_{i-1}$, $\mathcal{H}_i=\emptyset$. It is guaranteed that $F\leq i-1$ for $i\geq 2$, since $\mathcal{H}_1=\mathcal{X}_1\neq\emptyset$. After finding feasible scenarios $x_1^{(r)},\dots,x_{i-1}^{(r)}$, we sample the scenario $x_i^{(r)}$ for the ith agent from the uniform distribution π_i (Step 8). The sampled scenario is then checked at Step 9. If $x_i^{(r)}\notin\mathcal{H}_i$, it will be sampled again following π_1. The loop will terminate in finite time since $\text{Int}(\mathcal{H}_i\cap\mathcal{X})\neq\emptyset\ \forall i\in\{1,\dots,N\}$.

Algorithm 3 Scenarios Sampling Algorithm

Initialization Set $\mathcal{H}=\mathcal{B}\cap\mathcal{X}$, failed times $F=0$.
Output: Scenario $\boldsymbol{x}^{(r)}$.
1: Sample $x_1^{(r)}$ from $\pi_1(\boldsymbol{x})$.
2: **for** $i=2,\dots,N$ **do**
3: Construct a set $\mathcal{H}_i=\bigcap_{e\in\mathcal{C}_i}\mathcal{H}_{ie}$ following (25).
4: **if** $\mathcal{H}_i=\emptyset$ **then**
5: $F\leftarrow F+1$.
6: go to $i=i-F$ ($i=1$ is step 1).
7: **end if**
8: Sample $x_i^{(r)}$ from distribution $\pi_i=\frac{1}{\int_{\mathcal{X}_i}dx}$.
9: **while** $x_i^{(r)}\notin\mathcal{H}_i$ **do**
10: Sample $x_i^{(r)}$ from distribution π_i.
11: **end while**
12: **end for**

Proposition 4.1. *Consider Assumptions 3.1, 3.2, and assume scenarios $\boldsymbol{x}^{(r)}$, $r=1,\dots,R$ are sampled using Algorithm 3. We then have that $\boldsymbol{x}^{(r)}\in\mathcal{H}$, for all $r=1,\dots,R$. Moreover, all scenarios are independently and identically sampled.*

Proof. The feasibility result holds directly from the definition of every set $\mathcal{H}_i$ in (25) that $x_i^{(r)}$ is sampled from. As a result, we have $b_{ie}(x_i^{(r)},\{x_k^{(r)}\})\geq 0$ for any $i=1,\dots,N$, $e\in\mathcal{C}_i$, and $k\in\mathcal{V}_e\backslash i$. Therefore, $\boldsymbol{x}^{(r)}\in\mathcal{H}$. Moreover, for all $r=1,\dots,R$, $\boldsymbol{x}^{(r)}$ are independent since $x_1^{(r)}$, $r=1,\dots,R$ are independently sampled from the distribution π_1.

At Step 6, when $F=i-1$, it returns Step 1 to resample $x_1^{(r)}$. This happens when there exists $e\in\mathcal{C}_2$, and b_e is defined only on Agent 1 and 2, such that $\mathcal{H}_{2e}=\{x_2\in\mathcal{X}_2|b_e(x_2,x_1^{(r)})\geq 0\}=\emptyset$. $x_1^{(r)}$ will then be resampled from the distribution π_1 to make $\mathcal{H}_{2e}\neq\emptyset$. Therefore, the actual distribution $\tilde{\pi}_1$ from which $x_1^{(r)}$ is sampled is defined on a set $\tilde{\mathcal{X}}_1\subseteq\mathcal{X}_1$, which satisfies

$$\{x_2\in\mathcal{X}_2:b_e(x_2,x_1^*)\geq 0\}\neq\emptyset,\forall x_1^*\in\tilde{\mathcal{X}}_1.\tag{26}$$

It trivially holds that $\text{Int}(\tilde{\mathcal{X}}_1)\neq\emptyset$ since $\text{Int}(\mathcal{H})\neq\emptyset$, from Assumption 3.2. $\tilde{\pi}_1$ can be different from π_1, but is identical for every $r=1,\dots,R$. Similarly, the resampling mechanism implicitly defines distributions $\tilde{\pi}_2,\dots,\tilde{\pi}_N$ that may be different from $\pi_2,\dots,\pi_N$. But these distributions are identical for scenarios $\boldsymbol{x}^{(r)}$, $r=1,\dots,R$. □

We note here that the elements in $\boldsymbol{x}^{(r)}$ are correlated, but this will not influence the independence results in Proposition 4.1 since we seek independence across r.

4.3. Distributed safety verification

After sampling scenarios $\boldsymbol{x}^{(r)}$, $r=1,\dots,R$ using Algorithm 3, we are at the stage of solving the safety verification program (23).

Letting the local cost function $J_i(\boldsymbol{z}_i, \boldsymbol{\zeta}_i)$, and constraint function $\hat{h}_{ie}(\boldsymbol{z}_i, \boldsymbol{\zeta}_i)$ be

$$J_i(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) = \sum_{e\in\mathcal{C}_i}\left(z_{ie}^2 + H_i\sum_{r=1}^{R}\zeta_{ie}^{(r)}\right),$$
$$\hat{h}_{ie}^{(r)}(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) = h_{ie}(u_i(\boldsymbol{x}^{(r)})) - z_{ie} - \zeta_{ie}^{(r)}, r = 1, \ldots, R, \tag{27}$$

Algorithm 1 can be applied to solve the distributed scenario optimization problem (23). The relaxation variables in Algorithm 1 are unnecessary, since every optimization sub-problem across iterations is solvable. We then have the following theorem as the main result on distributed probabilistic safety. The following theorem constitutes the multi-agent counterpart of Theorem 2. Using the density functions constructed in Algorithm 3 and considering Assumption 3.2, there will be no repeated scenarios for $r = 1, \ldots, R$. Therefore, eliminating all the constraints that are not in the support set for (23) will not change the optimal solution $\boldsymbol{z}^*$, and hence due to Assumption 3.2, the non-degeneracy requirement of Assumption 2.1 is satisfied.

Theorem 4. *Let Assumptions 3.1 and 3.2 hold. Consider the optimization problem (23) and let $(\boldsymbol{z}^*, \{\boldsymbol{\zeta}^{*,(r)}\}_{r=1}^R)$ be the optimal solution. Choose $\beta_i \in (0, 1), i = 1, \ldots, N$, and set $\beta = \sum_{i=1}^N \beta_i$. For $i = 1, \ldots, N$, and $0 \le k_i \le R-1$, consider the polynomial equation in the t_i variable*

$$\binom{R}{k_i} t_i^{R-k_i} - \frac{\beta_i}{2R}\sum_{j=k_i}^{R-1}\binom{j}{k_i} t_i^{j-k_i} - \frac{\beta_i}{6R}\sum_{j=R+1}^{4R}\binom{j}{k_i} t_i^{j-k_i} = 0, \tag{28}$$

while for $k_i = R$ consider the polynomial equation

$$1 - \frac{\beta}{6N}\sum_{j=R+1}^{4R}\binom{j}{k_i} t_i^{j-R} = 0. \tag{29}$$

For every $i = 1, \ldots, N$ and any $k_i = 0, \ldots, R-1$, Eq. (28) has exactly two solutions in $[0, +\infty)$ denoted by $\underline{t}_i(k_i)$ and $\bar{t}_i(k_i)$, where $\underline{t}_i(k_i) \le \bar{t}_i(k_i)$. Instead, Eq. (29) has only one solution in $[0, +\infty)$, which we denote with $\bar{t}_i(R)$, while we define $\underline{t}_i(R) = 0$. Let $\underline{\epsilon}_i(k_i) := \max\{0, 1 - \bar{t}_i(k_i)\}$, $\bar{\epsilon}_i(k_i) := 1 - \underline{t}_i(k_i)$, and $\underline{\epsilon}(s^*) = \sum_{i=1}^N \underline{\epsilon}_i(s_i^*)$, $\bar{\epsilon}(s^*) = \min\{\sum_{i=1}^N \bar{\epsilon}_i(s_i^*), 1\}$. We then have that

$$\mathbb{P}^R\left\{\frac{\underline{\epsilon}(s^*)}{N} \le V(\boldsymbol{z}^*) \le \bar{\epsilon}(s^*)\right\} \ge 1 - \beta, \tag{30}$$

where s_i^* is the number of $\boldsymbol{x}^{(r)}$'s for which there exists $e \in \mathcal{C}_i$, such that $\sum_{k\in\mathcal{V}_e} h_{ke}(u_k(\boldsymbol{x}^{(r)})) \ge \sum_{k\in\mathcal{V}_e} z_{ke}^*$. Recalling Eq. (24), the violation probability $V(\boldsymbol{z}^*)$ is defined by $V(\boldsymbol{z}^*) = \mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\}$.

Proof. See the Appendix. □

Theorem 4 is a generalization of Garatti and Campi (2019, Theorem 2) to a multi-agent setting. It also extends Margellos et al. (2017), Wang, Papachristodoulou, and Margellos (2023) by determining the lower bound $\frac{\underline{\epsilon}(s^*)}{N}$. Theorem 4 states that with confidence at least $1-\beta$, the probability that the CBF constraints of the multi-agent system are violated by more than $\boldsymbol{z}^*$, lies within the interval $[\frac{\underline{\epsilon}(s^*)}{N}, \bar{\epsilon}(s^*)]$.

5. Simulation results

The distributed safe control input design and safety verification algorithms are numerically validated on a multi-robot positions swapping problem. To facilitate comparison, we adopt a similar setup as in Wang et al. (2017).

5.1. Multi-robot position swapping

Robots are assigned different initial positions and are required to navigate towards target locations. In a distributed framework, robots are equipped with sensing and communication modules for collision detection and information sharing. A network of ten robots, indexed by $i = 1, \ldots, 10$ are considered, with double integrator dynamics

$$\begin{bmatrix}\dot{\boldsymbol{p}}_i \\ \dot{\boldsymbol{v}}_i\end{bmatrix} = \begin{bmatrix}0 & I_{2\times2} \\ 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_i \\ \boldsymbol{v}_i\end{bmatrix} + \begin{bmatrix}0 \\ I_{2\times2}\end{bmatrix}\boldsymbol{a}_i, \tag{31}$$

where $\boldsymbol{p}_i \in \mathbb{R}^2$, $\boldsymbol{v}_i \in \mathbb{R}^2$ represent positions and velocities, and $\boldsymbol{a}_i \in \mathbb{R}^2$ is the control input, representing accelerations. The acceleration is limited as $\|\boldsymbol{a}_i\|_\infty \le a_i^{\max}$. $a_i^{\max}$ will be cleared in the sequel. Each robot is regarded as a disc centred at $\boldsymbol{p}_i$ with radius $D_i \in \mathbb{R}_+$. The safety certificate $s_{ij}(\boldsymbol{p}, \boldsymbol{v})$ for collision avoidance between robot i and j is defined by

$$s_{ij}(\boldsymbol{p}, \boldsymbol{v}) = \|\Delta\boldsymbol{p}_{ij}\|_2^2 - D_{ij}, \tag{32}$$

where $\Delta\boldsymbol{p}_{ij} = \boldsymbol{p}_i - \boldsymbol{p}_j$, $D_{ij} = D_i + D_j$. Note here that the system is heterogeneous as different robots have different mobility. Following Wang et al. (2017), the control barrier function for invariance certificates is then defined pair-wisely, as

$$b_{ij}(\boldsymbol{p}, \boldsymbol{v}) = \sqrt{2(a_i^{\max} + a_j^{\max})(\|\Delta\boldsymbol{p}_{ij}\|_2^2 - D_{ij})} + \frac{\Delta\boldsymbol{p}_{ij}^\top}{\|\Delta\boldsymbol{p}_{ij}\|_2^2}\Delta\boldsymbol{v}_{ij}, \tag{33}$$

where $\Delta\boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$. The function $b_{ij}(\boldsymbol{p}, \boldsymbol{v})$ is guaranteed to be a CBF since when $b_{ij}(\boldsymbol{p}, \boldsymbol{v}) > 0$, collision can be avoided with maximum braking acceleration $\boldsymbol{a}_i^{\max} + \boldsymbol{a}_j^{\max}$ applied to robots i and j. For $i = 1, \ldots, 5$, $\boldsymbol{a}_i^{\max} = 1$, while for $i = 6, \ldots, 10$, $\boldsymbol{a}_i^{\max} = 10$. Note that although $b_{ij}(\boldsymbol{p}, \boldsymbol{v})$ is guaranteed to be a CBF for safety certificate $s_{ij}(\boldsymbol{p}, \boldsymbol{v})$, the corresponding invariant set $\mathcal{B} = \prod_{\{i,j\}\in\mathcal{E}} \mathcal{B}_{ij}$ is possibly empty. Intuitively, this is since robots cannot utilize the maximum braking force to avoid collision with multiple other robots simultaneously. This problem is beyond the scope of this paper, and we still adopt the CBF as in (33).

5.2. Distributed control: Asymptotic algorithm

The distributed safe control design procedure of Algorithm 1 that exhibits asymptotic convergence and optimality guarantees that it is implemented for robots to swap positions with the opposite robots while avoiding collision. The resulting simulation results are shown in Fig. 2.

5.3. Distributed control: Truncated algorithm

The truncated Algorithm 2 is then implemented for the same setting, the truncation parameter $\eta = 30$.

The resulting swapping trajectories are shown in Fig. 3. Define

$$\rho_{\text{sum}}^k = \sum_{i=1}^N\sum_{e\in\mathcal{C}_i}\left((\rho_{ie}^k)^2 + M_i\rho_{ie}^k\right). \tag{34}$$

The evolution of the relaxation parameters $\rho_{\text{sum}}^0(\boldsymbol{x})$ and $\rho_{\text{sum}}^{30}(\boldsymbol{x})$ at each time step along the trajectory is shown in Figs. 4(a) and 4(b). It can be seen that ρ_{sum}^{30} is close to zero at every time step, even ρ_{sum}^0 is relatively large at some time steps. This empirically demonstrates the safety guarantees performance of the proposed distributed algorithm. From our experience, η could be much smaller for a practical implementation.

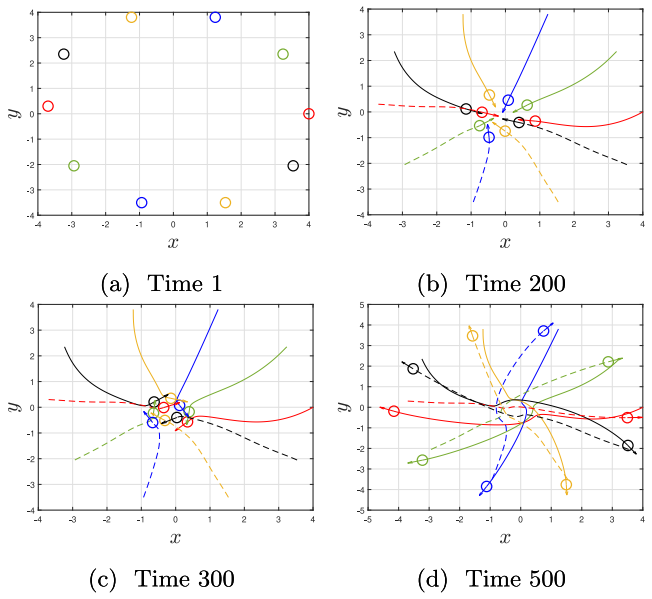


Fig. 2. Trajectory of ten robots swapping positions according to Algorithm 1. Robots with the same colour are swapping positions, and avoiding collision with the others.

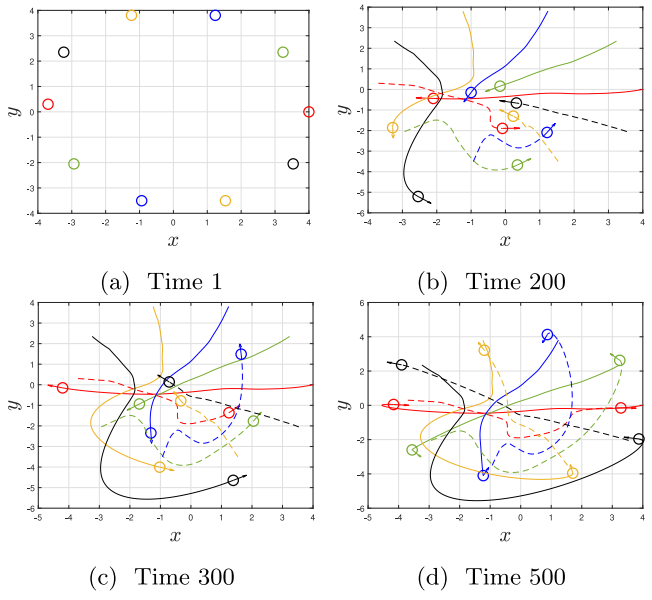
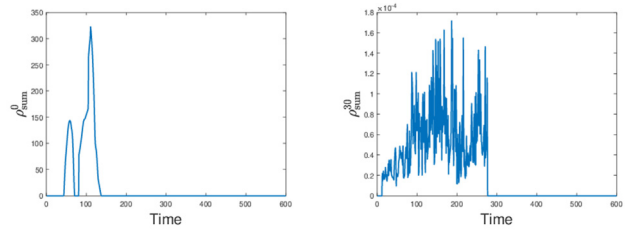


Fig. 3. Trajectory of ten robots swapping positions while avoiding collision by means of Algorithm 2, with $\eta = 30$.



(a) $\rho^0_{\text{sum}}(\boldsymbol{x})$ along the trajectory (b) $\rho^{30}_{\text{sum}}(\boldsymbol{x})$ along the trajectory

Fig. 4. Evolution of the relaxation parameters $\rho^0_{\text{sum}}(\boldsymbol{x})$ and $\rho^{30}_{\text{sum}}(\boldsymbol{x})$ evaluated at the state trajectory, across algorithm iterations.

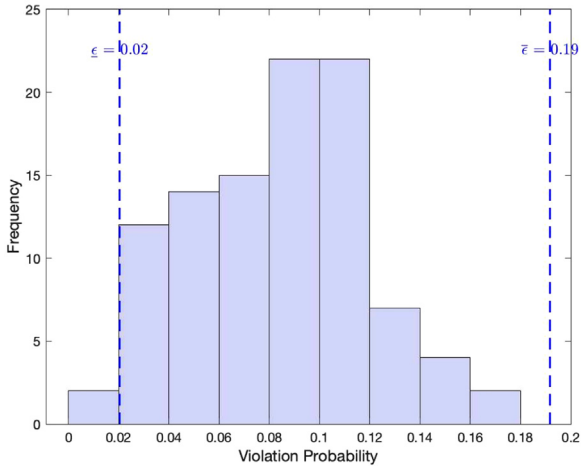


Fig. 5. Bar graph for the violation probability, and (with dashed lines) the theoretical bounds $[\underline{\epsilon}, \bar{\epsilon}]$.

5.4. Distributed safety verification

The proposed safety verification procedure is illustrated on a four-robot system within the working space

$$\mathcal{X} = \{\boldsymbol{p} \in \mathbb{R}^2 : \|\boldsymbol{p}\| \leq p^{\max} = 4\} \times \{\boldsymbol{v} \in \mathbb{R}^2 : \|\boldsymbol{v}\| \leq v^{\max} = 3\}.$$

Each robot employs Algorithm 2 to safely move towards the origin.

We first examine the effect of the number of scenarios R on the verification result. Let $\eta = 30$ for the four agents and consider $R = 2000$. Each agent maintains a confidence level $\beta_i = 0.025$ for $i \in \{1, \ldots, 4\}$. By solving the verification program (23) and applying Theorem 4, we establish that with confidence at least $1 - \beta = 0.9$, the violation probability satisfies $\underline{\epsilon} = 0.02 \leq \mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* = 0 \notin \mathcal{Z}_{\boldsymbol{x}}\} \leq \bar{\epsilon} = 0.19$. In practice, an even smaller β_i can be chosen; this would only have a mild effect on the interval $[\underline{\epsilon}, \bar{\epsilon}]$ due to the way this depends on the confidence.

To validate the obtained probabilistic result, we run 100 independent experiments, each with $R_v = 50$ scenarios $\boldsymbol{x}^{(1),j}, \ldots, \boldsymbol{x}^{(R_v),j}$, for $j \in \{1, \ldots, 100\}$. In each experiment $j \in \{1, \ldots, 100\}$, we monitor the frequency of violation f_v^j by

$$f_v^j = \sum_{i=1}^{R_v} I_{\mathcal{Z}^c_{\boldsymbol{x}^{(i),j}}}(\boldsymbol{z}^*). \tag{35}$$

In the above equation, $I_{\mathcal{Z}^c_{\boldsymbol{x}^{(i),j}}}(\boldsymbol{z}^*)$ is the indicator function that equals one if $\boldsymbol{z}^*$ belongs to the set complement of $\mathcal{Z}_{\boldsymbol{x}^{(i),j}}$, i.e., if $\boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}^{(i),j}}$, and zero otherwise. For each experiment $j \in \{1, \ldots, 100\}$, the violation probability $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\}$ is empirically calculated as

$$\hat{\mathbb{P}}^j\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\} = \frac{f_v^j}{R_v}. \tag{36}$$

Fig. 5 illustrates the bar graph of f_v^j, $j \in \{1, \ldots, 100\}$, while with dashed lines we highlight the theoretical bounds $[\underline{\epsilon}, \bar{\epsilon}]$. It can be observed that most of the empirical mass of the violation probability lies between $[\underline{\epsilon}, \bar{\epsilon}]$. This in turn implies that our bound for this case study offers a tight estimate of the violation probability.

The empirical cumulative distribution function (CDF) of $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\}$ can be constructed using $\hat{\mathbb{P}}^j\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\}$, $j \in \{1, \ldots, 100\}$. These results are shown in Fig. 6; it can be observed that the empirical probability implies that

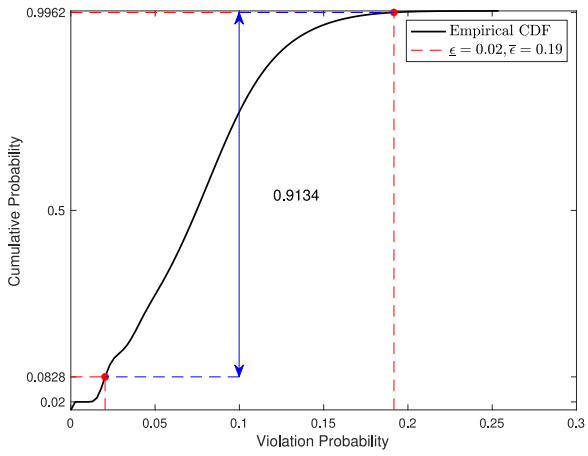


Fig. 6. Empirical Cumulative distribution function (CDF) for $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\}$, theoretic bounds $[\underline{\epsilon}, \overline{\epsilon}]$ and the corresponding empirical probability. The horizontal axis represents the violation probability while the vertical axis represents the associated empirical cumulative probability.

$\mathbb{P}\{\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\} \in [\underline{\epsilon}, \overline{\epsilon}]\} \approx 0.9134 \geq 1 - \beta = 0.9$, thus demonstrating numerically the confidence of the theoretical result of Theorem 4.

6. Conclusion

In this paper we presented distributed safe control design and safety verification algorithms for multi-agent systems. The proposed control algorithms introduce auxiliary and relaxation variables to allow feasibility across iterations. We guaranteed convergence to an optimal solution and established a sublinear convergence rate under certain conditions. We also addressed the problem of distributed safety verification for given control inputs. A scenario-based verification program was formulated and can be solved locally by each agent. The scenarios are sampled independently by a sequential algorithm. The distributed scenario program characterizes the probability of being unsafe, with both lower and upper bounds being determined. Simulation on a multi-robot swapping position problem demonstrated the efficacy of our result. Current work concentrates in accounting for communication delays and model uncertainty in real systems.

Appendix

Proof of Lemma 3.1. The dual function of the relaxed problem (16) is given by

$$\begin{aligned} q(\boldsymbol{\mu}) \coloneqq & \inf_{\{u_i \in \mathcal{U}_i\}_{i=1}^N, \boldsymbol{\rho} \geq 0} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \right\} \\ & + \sum_{e=1}^{E} \mu_e \left\{ \sum_{i \in \mathcal{V}_e} h_{ie}(u_i) - \sum_{i \in \mathcal{V}_e} \rho_{ie} \right\} \\ = & \inf_{\{u_i \in \mathcal{U}_i\}_{i=1}^N, \boldsymbol{\rho} \geq 0} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} \mu_e h_{ie}(u_i) \right\} \\ & + \sum_{e=1}^{E} \sum_{i \in \mathcal{V}_e} \left\{ \rho_{ie}^2 + (M_i - \mu_e) \rho_{ie} \right\}. \\ = & \, q_{\text{nom}}(\boldsymbol{\mu}) + \inf_{\boldsymbol{\rho} \geq 0} \sum_{e=1}^{E} \sum_{i \in \mathcal{V}_e} \left\{ \rho_{ie}^2 + (M_i - \mu_e) \rho_{ie} \right\}, \end{aligned} \tag{A.1}$$

where

$$q_{\text{nom}}(\boldsymbol{\mu}) \coloneqq \inf_{\{u_i \in \mathcal{U}_i\}_{i=1}^N} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} \mu_e h_{ie}(u_i) \right\} \tag{A.2}$$

is the dual function of the nominal problem (13). Let $\boldsymbol{\mu}^*$ be the maximizer for $q(\boldsymbol{\mu})$, and $\tilde{\boldsymbol{\mu}}$ be a maximizer of $q_{\text{nom}}(\boldsymbol{\mu})$.

Now consider the value of the second term in $q(\boldsymbol{\mu})$. If $\mu_e \leq M_i$ for some $e \in \{1, \dots, E\}, i \in \mathcal{V}_e$, then

$$\inf_{\rho_{ie} \geq 0} \rho_{ie}^2 + (M_i - \mu_e) \rho_{ie} = 0, \, \rho_{ie}^* = 0. \tag{A.3}$$

Otherwise if $\mu_e > M_i$, then

$$\inf_{\rho_{ie} \geq 0} \rho_{ie}^2 + (\mu_e - M_i) \rho_{ie} = -\frac{(M_i - \mu_e)^2}{4}, \, \rho_{ie}^* = \frac{\mu_e - M_i}{2}. \tag{A.4}$$

We first show that if (17) holds, then

$$M_i \geq \mu_e^*, \forall i \in \mathcal{V}_e, \forall e \in \{1, \dots, E\}, \tag{A.5}$$

where μ_e^* is the eth element of $\boldsymbol{\mu}^*$. Suppose for the sake of contradiction that there exists $e \in \{1, \dots, E\}, i \in \mathcal{V}_e$ such that $\mu_e^* > M_i$. Then, from (A.1), (A.3) and (A.4) we have

$$q(\boldsymbol{\mu}^*) < q_{\text{nom}}(\boldsymbol{\mu}^*) \leq q_{\text{nom}}(\tilde{\boldsymbol{\mu}}^*).$$

The second inequality comes from a fact that $\tilde{\boldsymbol{\mu}}^*$ is a maximizer of $q_{\text{nom}}(\boldsymbol{\mu})$. However, from (A.1), (A.3), and (17) we have

$$q(\tilde{\boldsymbol{\mu}}^*) = q_{\text{nom}}(\tilde{\boldsymbol{\mu}}^*).$$

We conclude that $q(\tilde{\boldsymbol{\mu}}^*) > q(\boldsymbol{\mu}^*)$, thus reach a contradiction as $\boldsymbol{\mu}^*$ maximizes $q(\boldsymbol{\mu})$. By (A.3) and (A.5) we have $\boldsymbol{\rho}^* = 0$, any $\tilde{\boldsymbol{\mu}}^*$ that maximizes $q_{\text{nom}}(\boldsymbol{\mu})$ also maximizes $q(\boldsymbol{\mu})$.

As a direct result of (A.3) and (A.4), the second part of $q(\boldsymbol{\mu})$, $\inf_{\boldsymbol{\rho} \geq 0} \sum_{e=1}^{E} \sum_{i \in \mathcal{V}_e} \left\{ \rho_{ie}^2 + (M_i - \mu_e) \rho_{ie} \right\}$, is concave and smooth. This is different from Notarnicola and Notarstefano (2019, Lemma III.2) where the dual function goes to $-\infty$ when $\mu_e > M_i$. Introducing a quadratic term for the relaxation variables enhances convexity of the primal function, hence smoothness of the dual function. □

Proof of Theorem 3. We begin with (a). Under Assumption 3.4, strongly duality holds for the primal problem (13) and the dual problem (A.1). With a slight abuse of notation, we define

$$\begin{aligned} q_i(\boldsymbol{\mu}_i) \coloneqq & \inf_{\{u_i \in \mathcal{U}_i\}, \boldsymbol{\rho} \geq 0} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \right. \\ & \left. + \sum_{e \in \mathcal{C}_i} \mu_{ie} (h_{ie}(u_i) - \rho_{ie}) \right\}. \end{aligned} \tag{A.6}$$

By Assumption 3.1, we have $\mathcal{G}_e$ is undirected and connected for every $e \in \{1, \dots, E\}$. Therefore, suppose $\mu_{ie} = \mu_{le}, \forall e \in \{1, \dots, E\}, i \in \mathcal{V}_e, l \in \mathcal{N}_i \cap \mathcal{V}_e$, then we can deduce that

$$\mu_{ie} = \mu_{le}, \forall i, l \in \mathcal{V}_e. \tag{A.7}$$

Recalling that $i \in \{1, \dots, N\}$ is the numbering of agent, $e \in \mathcal{C}_i$ is the numbering of CBF constraint that involves agent i, $\mathcal{N}_i$ is the set of neighbouring agents for agent i, and $\mathcal{V}_e$ is the set of agents in sub-network $\mathcal{G}_e$. $\mathcal{N}_i \cap \mathcal{V}_e \neq \emptyset$ due to Assumption 3.1. The new variables μ_{ie} and μ_{le} can be regarded as local copies of μ_e by agent i and agent l, which are associated with the eth CBF constraint. Using the decomposed dual function (A.6) and the new constraint (A.7), we come up with an equivalent decomposed dual problem

$$\begin{aligned} & \max_{\boldsymbol{\mu}_i \geq 0} \sum_{i=1}^{N} q_i(\boldsymbol{\mu}_i) \\ & \text{subject to } \mu_{ie} = \mu_{le}, \forall i \in \{1, \dots, N\}, e \in \mathcal{C}_i, l \in \mathcal{N}_i \cap \mathcal{V}_e, \end{aligned} \tag{A.8}$$

If $\mathcal{V}_e = \{1,\dots,N\}, \forall e \in \{1,\dots,E\}$, (A.8) is a generic dual decomposition problem (Notarstefano, Notarnicola, Camisa, et al., 2019, Section 3.1.3).

Consider the dual function of (A.8)

$$d(\boldsymbol{\lambda}) := \sum_{i=1}^{N} \sup_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e\in\mathcal{C}_i} \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e} \lambda_{il}^\top(\mu_{ie} - \mu_{le}) \right), \tag{A.9}$$

where λ_{il} is a free dual variable for the constraint $\mu_{ie} = \mu_{le}$ in (A.8). Recalling that the network $\mathcal{G}$ is undirected, for each $(i,l) \in \mathcal{E}$ we also have $(l,i) \in \mathcal{E}$. This indicates that in (A.9), we have both $\lambda_{il}^\top(\mu_{ie} - \mu_{le})$ and $\lambda_{li}^\top(\mu_{le} - \mu_{ie})$ for every given $i \in \{1,\dots,N\}$, $e \in \mathcal{C}_i$, $l \in \mathcal{N}_i \cap \mathcal{V}_e$. By gathering the terms involving $\boldsymbol{\mu}_i$ together, such as $\lambda_{il}^\top \mu_{ie}$ and $-\lambda_{li}^\top \mu_{ie}$, and doing some algebraic calculations, we obtain

$$d(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \sup_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e\in\mathcal{C}_i} \mu_{ie}^\top \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e} (\lambda_{il} - \lambda_{li}) \right) \tag{A.10}$$

As (A.9) is traversing every $i \in \{1,\dots,N\}$, μ_{le} in (A.9) is also contained in (A.10), for $l \in \mathcal{V}_e$. A procedure similar to (A.9) and (A.10) has been proposed in Notarnicola and Notarstefano (2019, Section III.B) but only for one network $\mathcal{G}$. Our formulation generalizes these results to constraints defined on multiple sub-networks $\mathcal{G}_e$, for $e \in \{1,\dots,E\}$.

The dual problem of (A.8) is then given by

$$d^* = \min_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}). \tag{A.11}$$

Strong duality holds between problem (A.8) and (A.11) since (A.8) is an linear equality constrained concave problem. Therefore, solving problem (A.11) leads to the optimal solution of problem (A.8). Solving problem (A.11) has advantages in terms of distributed computation. This can be seen by applying the gradient descent method to solve (A.11). From (A.9), for every $i \in \{1,\dots,N\}$, $e \in \mathcal{C}_i$, and $l \in \mathcal{N}_i \cap \mathcal{V}_e$, the gradient $\nabla d(\lambda_{il})$ is given by

$$\nabla d(\lambda_{il}) = \mu_{ie} - \mu_{le}. \tag{A.12}$$

At iteration k, each agent i performs two steps:

- (i) for every $e \in \mathcal{C}_i$, $l \in \mathcal{N}_i \cap \mathcal{V}_e$, calculate the gradient $\nabla d(\lambda_{il}^k)$: receive λ_{li}^k, $l \in \mathcal{N}_i \cap \mathcal{V}_e$, and compute μ_{ie} by solving

$$\max_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e\in\mathcal{C}_i} \mu_{ie}^\top \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right). \tag{A.13}$$

- (ii) use gradient descent: for every $e \in \mathcal{C}_e$, $l \in \mathcal{N}_i \cap \mathcal{V}_e$, receive μ_{le}^k and update λ_{il} by (A.12):

$$\lambda_{il}^{k+1} = \lambda_{il}^k - \gamma^k(\mu_{ie}^k - \mu_{le}^k). \tag{A.14}$$

(A.14) is Step 5 of Algorithm 1. We then show that solving (A.13) is equivalent to solving 18 at Step 3. For every $i \in \{1,\dots,N\}$, dualizing the CBF constraints in 18 by $\boldsymbol{\mu}_i \geq 0$ yields a dual problem

$$\begin{aligned}
&\max_{\boldsymbol{\mu}_i\geq 0} \inf_{\{u_i\in\mathcal{U}_i\},\boldsymbol{\rho}\geq 0} \left\{ J_i(u_i) + \sum_{e\in\mathcal{C}_i} (\rho_{ie}^2 + M_i\rho_{ie}) \right. \\
&\left. + \sum_{e\in\mathcal{C}_i} \mu_{ie}(h_{ie}(u_i) - \rho_{ie}) \right\} + \sum_{e\in\mathcal{C}_i} \mu_{ie}^\top \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \\
&\overset{(A.6)}{=} \max_{\boldsymbol{\mu}_i\geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e\in\mathcal{C}_i} \mu_{ie}^\top \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right),
\end{aligned} \tag{A.15}$$

which is (A.13). Therefore, Steps 2–5 in Algorithm 1 involve performing gradient descent to solve problem (A.11) in a distributed manner.

Diminishing step-size is used here as Notarnicola and Notarstefano (2019). Specifically, (A.13) is the dual problem of 18. Strong duality holds for large enough $\boldsymbol{\rho}$ as the relaxed CBF constraints hold strictly. Updating (A.14) is the same as 19 for every agent across iterations. Given that $d(\boldsymbol{\lambda})$ is convex, gradient descent guarantees that $d(\boldsymbol{\lambda}^k)$ converges to the optimal value $d^* = J^*$ since strong duality holds between (16) and (A.8), as well as (A.8) and (A.11). Moreover, the relaxed problem (16) is strongly (hence also strictly) convex, which indicates uniqueness of the optimal solution $(\boldsymbol{u}_{\text{rel}}^*, \boldsymbol{\rho}^*)$. Using Lemma 3.1, we obtain $\boldsymbol{u}_{\text{rel}}^* = \boldsymbol{u}_{\text{nom}}^*$, which is the optimal solution of (13).

We then prove (b). First we prove that $q(\boldsymbol{\mu})$ in (A.1) is a concave quadratic function. When every $\mathcal{U}_i = \mathbb{R}^{m_i}$ and the CBF constraints are linearly independent, the relaxed CBF-QP (16) is a linearly constrained strongly convex quadratic problem. Following the example Boyd, Boyd, and Vandenberghe (2004, Section 5.2.4, Eq. 5.28),[1] $q_{\text{nom}}(\boldsymbol{\mu})$ in (A.2) is a strongly concave quadratic function. Together with (A.1), (A.3) and (A.4), we conclude that $q(\boldsymbol{\mu})$ is a strongly concave and smooth function. From duality between strong concavity (convexity) and smoothness (Kakade, Shalev-Shwartz, Tewari, et al., 2009, Theorem 6), $d(\boldsymbol{\lambda})$ is a smooth and necessarily convex function. Using constant step size

$$0 < \gamma < \frac{1}{2L}, \tag{A.16}$$

where L is Lipschitz constant of $\nabla d(\boldsymbol{\lambda})$, in a gradient descent method to minimize a smooth and convex function $d(\boldsymbol{\lambda})$, the generated iterates converge sublinearly as

$$\begin{aligned}
d(\boldsymbol{\lambda}^k) - J^* &\leq \frac{2(d(\boldsymbol{\lambda}^0) - J^*)\|\boldsymbol{\lambda}^{\mathbf{0}} - \boldsymbol{\lambda}^*\|_2^2}{2\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_2^2 + k\gamma(2 - L\gamma)(d(\boldsymbol{\lambda}^0) - J^*)} \\
&\leq \frac{2(d(\boldsymbol{\lambda}^0) - J^*)\|\boldsymbol{\lambda}^{\mathbf{0}} - \boldsymbol{\lambda}^*\|_2^2}{k\gamma(d(\boldsymbol{\lambda}^0) - J^*)} \leq \frac{2\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_2^2}{k\gamma}.
\end{aligned} \tag{A.17}$$

The first inequality is proved by Nesterov (2003, Theorem 2.1.14), the second one comes from eliminating the term $\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_2^2$ from the denominator, and considering $2 - L\gamma \geq 1$ from (A.16).

Recalling the expression of $d(\boldsymbol{\lambda})$ from (A.6), and the duality result from (A.15), we have

$$\begin{aligned}
&d(\boldsymbol{\lambda}^k) = \\
&\sum_{i=1}^{N} \inf_{u_i,\boldsymbol{\rho}_i\geq 0} \left(\sup_{\boldsymbol{\mu}_i\geq 0} \left(J_i(u_i) + \sum_{e\in\mathcal{C}_i} (\rho_{ie}^2 + M_i\rho_{ie}) \right) \right. \\
&\left. + \sum_{e\in\mathcal{C}_i} \mu_{ie}(h_{ie}(u_i) - \rho_{ie}) + \sum_{e\in\mathcal{C}_i} \mu_{ie}^\top \sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right) \\
&= \sum_{i=1}^{N} \left(J_i(u_i^k) + \sum_{e\in\mathcal{C}_i} ((\rho_{ie}^k)^2 + M_i\rho_{ie}^k) \right) \\
&= \sum_{i=1}^{N} \|u_i^k - u^{\text{des}}\|^2 + \rho_{\text{sum}}^k = H(\boldsymbol{u}^k, \boldsymbol{\rho}^k).
\end{aligned} \tag{A.18}$$

Hence, by (A.17) and (A.18), we conclude that $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k) - J^* < \frac{2\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|_2^2}{\gamma k}$. □

[1] The example demonstrates that the dual function of a convex quadratically constrained quadratic programming problem is a concave quadratic function. Our problem is as a special case where the quadratic terms are zero in the constraints.

Proof of Theorem 4. We have that

$$\mathbb{P}^R\left\{\frac{\sum_{i=1}^N \underline{\epsilon}_i(s_i^*)}{N} \le \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}\right\} \le \sum_{i=1}^N \bar{\epsilon}_i(s_i^*)\right\}$$
$$= \mathbb{P}^R\left\{\frac{1}{N}\sum_{i=1}^N \underline{\epsilon}_i(s_i^*) \le \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\right.\right.$$
$$\left.\left.\exists i\in\{1,\dots,N\},\ \boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\} \le \sum_{i=1}^N \bar{\epsilon}_i(s_i^*)\right\}$$
$$= \mathbb{P}^R\left\{\frac{1}{N}\sum_{i=1}^N \underline{\epsilon}_i(s_i^*) \le \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\bigcup_{i=1}^N\left\{\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\}\right.$$
$$\left.\bigcap\mathbb{P}\left\{\bigcup_{i=1}^N\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\} \le \sum_{i=1}^N \bar{\epsilon}_i(s_i^*)\right\} \tag{A.19}$$

The second equation comes from the fact that $\boldsymbol{z}^*\in\mathcal{Z}_{\boldsymbol{x}}$ is equivalent to $\boldsymbol{z}^*\in\mathcal{Z}_{\boldsymbol{x}}^i\ \forall i\{1,\dots,N\}$. The second equation changes $\exists i\in\{1,\dots,N\}, \boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i$ into $\bigcup_{i=1}^N\{\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\}$. Similar tricks have been used in Margellos et al. (2017, Equation 15) to derive an upper bound for the inner probability. Here we extend the results to both upper and lower bounds, using Theorem 2. We separately deal with the two bounds on the probability. For the upper bound we have

$$\mathbb{P}^R\left\{\mathbb{P}\left\{\bigcup_{i=1}^N\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\} \le \sum_{i=1}^N \bar{\epsilon}_i(s_i^*)\right\}$$
$$\ge \mathbb{P}^R\left\{\sum_{i=1}^N \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\} \le \sum_{i=1}^N \bar{\epsilon}_i(s_i^*)\right\}.$$

The equality is achieved when for any $i\neq j$, $\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i$ and $\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^j$ are mutually exclusive. For the lower bound we have

$$\mathbb{P}^R\left\{\frac{1}{N}\sum_{i=1}^N \underline{\epsilon}_i(s_i^*) \le \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\bigcup_{i=1}^N\left\{\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\}\right\}$$
$$\ge \mathbb{P}^R\left\{N\cdot\frac{1}{N}\sum_{i=1}^N \underline{\epsilon}_i(s_i^*) \le \sum_{i=1}^N \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\}.$$

The equality is achieved if for any $i\neq j$, $\boldsymbol{z}^*\neq\mathcal{Z}_{\boldsymbol{x}}^i \Leftrightarrow \boldsymbol{z}^*\neq\mathcal{Z}_{\boldsymbol{x}}^j$ and $\underline{\epsilon}_i(s_i^*) = \underline{\epsilon}_j(s_j^*)$. The right-hand side of (A.19) can be then lower-bounded by

$$\mathbb{P}^R\left\{N\cdot\frac{1}{N}\sum_{i=1}^N \underline{\epsilon}_i(s_i^*) \le \sum_{i=1}^N \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right.$$
$$\left.\bigcap\sum_{i=1}^N \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\} \le \sum_{i=1}^N \bar{\epsilon}_i(s_i^*)\right\}$$
$$\ge \mathbb{P}^R\left\{\bigcap_{i=1}^N\left\{\underline{\epsilon}_i(s_i^*) \le \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\} \le \bar{\epsilon}_i(s_i^*)\right\}\right\}$$
$$\ge 1-\sum_{i=1}^N \mathbb{P}^R\left\{\bar{\epsilon}_i(s_i^*) < \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right.$$
$$\left.\bigcup\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\} < \underline{\epsilon}_i(s_i^*)\right\}. \tag{A.20}$$

By applying Theorem 2 to every agent $i\in\{1,\dots,N\}$, in the sense that it holds only for the CBF constraints that involve agent i, we have that for any $i\in\{1,\dots,N\}$

$$\mathbb{P}^R\left\{\boldsymbol{x}\in\mathcal{H}:\underline{\epsilon}_i(s_i^*) \le \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\} \le \bar{\epsilon}_i(s_i^*)\right\}$$
$$\ge 1-\beta_i$$
$$\Rightarrow \sum_{i=1}^N \mathbb{P}^R\left\{\bar{\epsilon}_i(s_i^*) < \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right.$$
$$\left.\bigcup\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\} < \underline{\epsilon}_i(s_i^*)\right\} < \sum_{i=1}^N \beta_i. \tag{A.21}$$

Here s_i^* is the number of $\boldsymbol{x}^{(r)}$'s for which there exists $e\in\mathcal{C}_i$, such that $\sum_{k\in\mathcal{V}_e} h_{ke}(u_k(\boldsymbol{x}^{(r)})) \ge \sum_{k\in\mathcal{V}_e} z_{ke}^*$. For a specific r, this means that agent i recognizes that at least one CBF constraint is violated up to level $\sum_{i\in\mathcal{V}_e} z_{ie}^*$, over this scenario $\boldsymbol{x}^{(r)}$. After solving the scenario program (23) and communicating with the neighbouring agents in $\mathcal{G}_e$ in a distributed manner, every individual agent is able to compute $\bar{\epsilon}_i^*(s_i^*)$, $\underline{\epsilon}_i^*(s_i^*)$ by (28), (29). Since $\frac{\underline{\epsilon}(s^*)}{N} < \underline{\epsilon}(s^*) < \bar{\epsilon}(s^*)$, substituting (A.21) into (A.19) with $i=1,\dots,N$ we obtain

$$\mathbb{P}^R\left\{\frac{\underline{\epsilon}(s^*)}{N} \le \mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}\right\} \le \bar{\epsilon}(s^*)\right\} \ge 1-\beta. \quad \square \tag{A.22}$$

References

Alam, A., Gattami, A., Johansson, K. H., & Tomlin, C. J. (2014). Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations. *Control Engineering Practice*, *24*, 33–41.

Ames, A. D., Grizzle, J. W., & Tabuada, P. (2014). Control barrier function based quadratic programs with application to adaptive cruise control. In *53rd IEEE conference on decision and control* (pp. 6271–6278). IEEE.

Ames, A. D., Xu, X., Grizzle, J. W., & Tabuada, P. (2016). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, *62*(8), 3861–3876.

Axelsson, J. (2016). Safety in vehicle platooning: A systematic literature review. *IEEE Transactions on Intelligent Transportation Systems*, *18*(5), 1033–1045.

Borrmann, U., Wang, L., Ames, A. D., & Egerstedt, M. (2015). Control barrier certificates for safe swarm behavior. *IFAC-PapersOnLine*, *48*(27), 68–73.

Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Calafiore, G., & Campi, M. C. (2005). Uncertain convex programs: Randomized solutions and confidence levels. *Mathematical Programming*, *102*(1), 25–46.

Calafiore, G. C., & Campi, M. C. (2006). The scenario approach to robust control design. *IEEE Transactions on Automatic Control*, *51*(5), 742–753.

Camisa, A., Farina, F., Notarnicola, I., & Notarstefano, G. (2021). Distributed constraint-coupled optimization via primal decomposition over random time-varying graphs. *Automatica*, *131*, Article 109739.

Campi, M. C., & Garatti, S. (2008). The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization*, *19*(3), 1211–1230.

Campi, M. C., & Garatti, S. (2018). Wait-and-judge scenario optimization. *Mathematical Programming*, *167*(1), 155–189.

Chen, Y., Singletary, A., & Ames, A. D. (2020). Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach. *IEEE Control Systems Letters*, *5*(1), 127–132.

Ding, F., He, J., Ren, Y., Wang, H., & Zheng, Y. (2022). Configuration-aware safe control for mobile robotic arm with control barrier functions. arXiv preprint arXiv:2204.08265.

Duchi, J. C., Agarwal, A., & Wainwright, M. J. (2011). Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, *57*(3), 592–606.

Falsone, A., Margellos, K., Garatti, S., & Prandini, M. (2017). Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica*, *84*, 149–158.

Falsone, A., Notarnicola, I., Notarstefano, G., & Prandini, M. (2020). Tracking-ADMM for distributed constraint-coupled optimization. *Automatica*, *117*, Article 108962.

Garatti, S., & Campi, M. C. (2019). Risk and complexity in scenario optimization. *Mathematical Programming*, 1–37.

Garg, K., Usevitch, J., Breeden, J., Black, M., Agrawal, D., Parwana, H., et al. (2024). Advances in the theory of control barrier functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems. *Annual Reviews in Control*, *57*, Article 100945.

Hsu, S.-C., Xu, X., & Ames, A. D. (2015). Control barrier function based quadratic programs with application to bipedal robotic walking. In *2015 American control conference* (pp. 4542–4548). IEEE.

Isaly, A., Ghanbarpour, M., Sanfelice, R. G., & Dixon, W. E. (2024). On the feasibility and continuity of feedback controllers defined by multiple control barrier functions. *IEEE Transactions on Automatic Control*.

Kakade, S., Shalev-Shwartz, S., Tewari, A., et al. (2009). *vol. 2, On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization* (1), (p. 35). Unpublished Manuscript, http://ttic.uchicago.edu/shai/papers/kakadeshalevtewari09.pdf.

Li, X., Feng, G., & Xie, L. (2020). Distributed proximal algorithms for multiagent optimization with coupled inequality constraints. *IEEE Transactions on Automatic Control*, *66*(3), 1223–1230.

Margellos, K., Falsone, A., Garatti, S., & Prandini, M. (2017). Distributed constrained optimization and consensus in uncertain networks via proximal minimization. *IEEE Transactions on Automatic Control*, *63*(5), 1372–1387.

Nedić, A., & Ozdaglar, A. (2009). Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, *19*(4), 1757–1780.

Nedic, A., Ozdaglar, A., & Parrilo, P. A. (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, *55*(4), 922–938.

Nesterov, Y. (2003). *vol. 87, Introductory lectures on convex optimization: A basic course*. Springer Science & Business Media.

Notarnicola, I., & Notarstefano, G. (2019). Constraint-coupled distributed optimization: A relaxation and duality approach. *IEEE Transactions on Control of Network Systems*, *7*(1), 483–492.

Notarstefano, G., Notarnicola, I., Camisa, A., et al. (2019). Distributed optimization for smart cyber-physical networks. *Foundations and Trends® in Systems and Control*, *7*(3), 253–383.

Parrilo, P. A. (2000). *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology.

Primbs, J. A., Nevistić, V., & Doyle, J. C. (1999). Nonlinear optimal control: A control Lyapunov function and receding horizon perspective. *Asian Journal of Control*, *1*(1), 14–24.

Schneeberger, M., Dörfler, F., & Mastellone, S. (2023). SOS construction of compatible control Lyapunov and barrier functions. *IFAC-PapersOnLine*, *56*(2), 10428–10434.

Shi, W., Ling, Q., Yuan, K., Wu, G., & Yin, W. (2014). On the linear convergence of the ADMM in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, *62*(7), 1750–1761.

Sontag, E. D. (1989). A 'universal'construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, *13*(2), 117–123.

Tan, X., & Dimarogonas, D. V. (2021). Distributed implementation of control barrier functions for multi-agent systems. *IEEE Control Systems Letters*, *6*, 1879–1884.

Tan, X., & Dimarogonas, D. V. (2022). Compatibility checking of multiple control barrier functions for input constrained systems. In *2022 IEEE 61st conference on decision and control* (pp. 939–944). IEEE.

Usevitch, J., Garg, K., & Panagou, D. (2020). Strong invariance using control barrier functions: A clarke tangent cone approach. In *2020 59th IEEE conference on decision and control* (pp. 2044–2049). IEEE.

Wang, L., Ames, A. D., & Egerstedt, M. (2017). Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, *33*(3), 661–674.

Wang, H., Li, Y., Yu, W., He, J., & Guan, X. (2019). Moving obstacle avoidance and topology recovery for multi-agent systems. In *American control conference* (pp. 2696–2701). IEEE.

Wang, H., Papachristodoulou, A., & Margellos, K. (2023). Distributed safety verification for multi-agent systems. *62nd IEEE Conference on Decision and Control*.

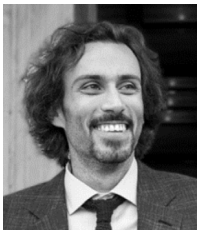
Wright, D. G. (1994). Tychonoff's theorem. *Proceedings of the American Mathematical Society*, *120*(3), 985–987.

Xiao, W., & Cassandras, C. G. (2021). Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees. *Automatica*, *123*, Article 109333.

Xu, X. (2018). Constrained control of input–output linearizable systems using control sharing barrier functions. *Automatica*, *87*, 195–201.

Han Wang received the B.S. degree in Information Security from Shanghai Jiao Tong University, China, in 2020, and the Ph.D. degree in control from the University of Oxford, UK, in 2024. He is currently a Postdoctoral Researcher at ETH Zürich. His research interests include safe and stable control design, learning to control and convex optimization, with applications to industrial automation.

Antonis Papachristodoulou (Fellow, IEEE) received the M.A./M.Eng. degree in Electrical and Information Sciences from the University of Cambridge, United Kingdom, and the Ph.D. degree in Control and Dynamical Systems (with a minor in Aeronautics) from the California Institute of Technology, Pasadena, CA, USA. He is currently a Professor of Engineering Science at the University of Oxford, United Kingdom, and a Tutorial Fellow at Worcester College, Oxford. He was previously an EPSRC Fellow. His research interests include large-scale nonlinear systems analysis, sum-of-squares programming, synthetic and systems biology, networked systems, and flow control.



Kostas Margellos received the Diploma degree in Electrical Engineering from the University of Patras, Greece, in 2008, and the Ph.D. degree in Control Engineering from ETH Zürich, Switzerland, in 2012. From 2013 to 2015, he was a Postdoctoral Researcher with ETH Zürich, UC Berkeley, USA, and Politecnico di Milano, Italy. In 2016, he joined the Control Group, Department of Engineering Science, University of Oxford, United Kingdom, where he is currently an Associate Professor. He is also a Fellow of Reuben College, Oxford, and a Lecturer at Worcester College, Oxford. His research interests include optimization and control of complex uncertain systems, with applications to energy and transportation networks.