Distributed Safe Control Design and Safety Verification for Multi-Agent Systems [⋆]

Han Wang, Antonis Papachristodoulou, Kostas Margellos

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

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. The proposed distributed algorithm addresses infeasibility issues of existing schemes by dynamically allocating auxiliary variables across iterations. The resulting control input is guaranteed to be optimal, and renders the system safe. 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 a multi-agent system probabilistically, using a certain locally Lipschitz continuous feedback controller 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.

Key words: Distributed Control, Safe Control, Multi-Agent Systems, Scenario Approach, Nonlinear Systems

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 [2,3], vehicle platooning [4,5], vehicle merging control [6], 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 [7–9]. 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 [7]. By solving online QP problems for every state, the system can be guaranteed to be safe [10,11]. Most of the existing results in this direction involve a centralized approach; however, multi-agent considerations call for distributed solution regimes. In this paper we address the distributed safety problem for multi-agent systems.

Related to the problem considered in this paper, CBFs for multi-robot systems were studied in [12–14]. These works propose to split the CBF constraints into two components for neighbouring agents: the computation is therefore distributed as every agent solves a local optimisation problem. An improved constraint sharing mechanism is developed in [15], where the CBF constraints are dynamically tuned for compatibility. Optimality is further considered in [16], and a dynamical constraint allocation scheme among agents based on a consensus protocol is proposed. In our work, we aim at dealing with the problem of feasibility and optimality simultaneously, as well as considering multiple CBF constraints for safety. In essence, the distributed CBF-based safe control design problem can be seen under the lens of distributed optimisation.

Distributed optimisation for a multi-agent system aims to design a distributed protocol that involves solving an optimisation problem locally for every agent. Algorithms can be divided into two types, dual decomposition [17–20] and primal decomposition-based [21–25]. 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 [21, 24, 25] or updating auxiliary variables [22, 23], 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 the same structure with the one considered in [22, 23], primal decomposition methods are leveraged.

Our first contribution is to provide a method for constructing a distributed, safe controller. To parallelize the computation, we leverage the primal decomposition method presented in [23] to decompose the coupling constraints via the introduction of auxiliary variables. We also introduce additional relaxation variables for every CBF constraint to overcome incompatibility issues of multiple safety certificates, and avoid compromising the control ability. Compared with other methods in the literature, our approach offers feasibility and optimality guarantees, and exhibits a sublinear convergence rate.

To reduce the communication and computation burden, a truncation mechanism is proposed to allow us to terminate the algorithm prior to convergence. To give a probabilistic guarantee for safety over the state space, we leverage scenario approach [26–30], which samples a number of independent states from the state space and enforces the constraint only at these realizations.

Another problem of interest in this work is safety verification. For a dynamical system, safety requires the trajectory to be within a safe set. Given the vector field, a target set and an unsafe set, solving a reach-avoid game [31, 32] yields a set from which all trajectories start can reach the target set without entering the unsafe set. In this sense, safety verification lies in the scope of reachability analysis. The main challenge here is how to solve the underlying Hamilton Jacobi partial differential equation. To bypass this difficulty, the barrier certificates method was proposed in a convex programming framework [33, 34]. A barrier certificate identifies an invariant set inside the safe set. System trajectories cannot escape from the underlying invariant set, and this directly leads to safety. Numerical methods for verifying safety using barrier certificates with convex programs entails sum-of-squares (SOS) programs [35, 36], which are equivalent to semi-definite programs. In real applications, the system model and control input are usually not precisely known, or are even unknown. In this setup, another type of verification method [37] using sampled data was proposed recently. Probabilistically guaranteed safety is ensured using the so called scenario approach [26–30].

A further contribution is that of constructing a distributed safety verification algorithm. Here we address the problem of certifying safety for a multi-agent system. We propose a scenario-based verification algorithm for a probabilistic quantification of safety. A sequential sampling algorithm is proposed to sample scenarios efficiently in a distributed fashion. For the probabilistic result, we extend the state-of-the-art result [30, Theorem 1] to the multi-agent setting. Both lower and upper bounds on the probability of being unsafe are established, while the safety verification program is also shown to be amenable to parallelization.

1.2 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 $\mathbb{R}$, $\mathbb{R}^N$, $\mathbb{R}_+$ to represent the set of one-dimensional, N-dimensional and nonnegative real numbers, respectively. $\mathbb{N}$ is the set of natural numbers. For matrices A and B, $A \preceq B$ implies $B - A$ is positive semi-definite. A continuous function $\alpha(\cdot) : (-b, a) \rightarrow (-\infty, +\infty)$ 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 $\mathcal{E}$. Throughout the paper $\mathcal{S}$ is used for a safe set, $\mathcal{B}$ is used for an invariant set. Boldface symbols are used as stacked vectors for scalar or vector elements, e.g., $\boldsymbol{x} = [x_1^\top, \ldots, x_N^\top]^\top$. Specifically, $\mathbf{0}$ is vector whose elements are all zero, and I is an identity matrix, with their dimensions being clear from the context. For a set $\mathcal{K}$, $|\mathcal{K}|$ denotes its cardinality. For a function $s(x) : \mathbb{R}^n \to \mathbb{R}$, we use the calligraphic font to represent the corresponding zero-super level set, i.e., $\mathcal{S} := \{\boldsymbol{x} | s(\boldsymbol{x}) \geq 0\}$.

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 $\mathcal{X} \subset \mathbb{R}^n$. The existence and uniqueness of solutions $x(u(\cdot), t, x_0)$ is assumed where x_0 is the initial state.

The safe set $\mathcal{S}$ is represented by the zero-super level set of a continuously differentiable function $s(x)$. Dually, the unsafe set $\bar{\mathcal{S}}$ can be defined as the complementary set. We denote by $\mathcal{S}$ the safe set, by $\partial\mathcal{S}$ the boundary of the safe set, by $\text{Int}(\mathcal{S})$ the interior of the safe set, and by $\bar{\mathcal{S}}$ the unsafe set, respectively. 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 [7].

Control barrier functions are an extension to barrier certificates [33] for safety verification. It has been revealed in these papers that safety is closely related to the notion of *control invariance*.

Definition 1 *A set $\mathcal{B} \subset \mathbb{R}^n$ is said to be control invariant with respect to* (1), *if for any $x_0 \in \mathcal{B}$, there exists $u \in \mathcal{U}$ such that $\psi(u, t, x_0) \in \mathcal{B}$.*

The relationship between safety and control invariance is demonstrated in the following equivalence lemma.

Lemma 2 ([38]) *System* (1) *is able to maintain safety under $\mathcal{S}$, if and only if there exists a control invariant set $\mathcal{B} \subseteq \mathcal{S}$.*

Clearly, given a control invariant set $\mathcal{B}$, a safe control input $u(x)$ always exists for any $x \in \mathcal{B}$. The control barrier function approach answers the question of how to design a closed loop safe control input $u(x)$ inside $\mathcal{B}$. The notion of control barrier functions is related to the notion of extended class-$\mathcal{K}$ functions.

Definition 3 *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 4 *[7, 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 $\mathcal{B}$ control invariant.*

3 Distributed Safe Control Law

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

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \tag{4}$$

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. 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 $f(\boldsymbol{x})$, $g(\boldsymbol{x})$ stack the dynamics for each agent. The domain and control admissible set for the multi-agent system are then defined by

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

where $\prod$ represents the Cartesian product for the state space of all the agents. Given that all $\mathcal{X}_i$, $i = 1, \ldots, N$, are assumed to be compact, compactness of $\mathcal{X}$ is assured using Tychonoff's theorem [39]. 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 graph $\mathcal{G}$, with nodes set $\mathcal{V} = \{1, \ldots, N\}$, and edges set $\mathcal{E}$ such that $\{i, j\} \in \mathcal{E}$ if agent j communicates with agent i. Agents are grouped in E sub-networks with specific safety requirement. For each sub-network $\mathcal{G}_e$, $e = 1, \ldots, E$, the set of grouped 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 group e. Each

agent i can communicate and cooperate with its neighbour $j \in \mathcal{N}_i$ to stay safe inside group e by ensuring

$$\boldsymbol{x}_e \in \mathcal{S}_e := \{\boldsymbol{x}_e : s_e(\boldsymbol{x}_e) \geq 0\}, \tag{5}$$

where $s_e(\cdot) \in \mathbb{R}$. We let $\mathcal{C}_i$ be the set of constraints agent i participates in; then we have $\mathcal{V}_e = \{i | e \in \mathcal{C}_i\}$.

Assumption 5 (Connectivity) *For each $e = 1, \ldots, E$, sub-network $\mathcal{G}_e$ is connected and undirected.*

Assumption 6 *Suppose $\mathcal{S} = \bigcap_{e=1}^{E} \mathcal{S}_e \neq \emptyset$. There exists E control barrier functions $b_e(\cdot)$, such that $\mathcal{B}_e := \{\boldsymbol{x}_e : b_e(\boldsymbol{x}_e) \geq 0\} \subseteq \mathcal{S}_e$, and $\mathcal{B} = \bigcap_{e=1}^{E} \mathcal{B}_e \neq \emptyset$. Moreover, $\mathcal{H} := \mathcal{B} \cap \mathcal{X} \neq \emptyset$.*

Assumption 7 *For the multi-agent system (4) and CBFs $b_e(\boldsymbol{x}_e)$, $e = 1, \ldots, E$. For every $\boldsymbol{x} \in \mathcal{B}$, there exists $\boldsymbol{u} = [u_1^\top \in \mathcal{U}_1, \ldots, u_N^\top \in \mathcal{U}_N]^\top \in \mathcal{U}$, such that for any $e = 1, \ldots, E$:*

$$\sum_{k \in \mathcal{V}_e} \left(\frac{\partial b_e}{\partial x_k}(f_k(x_k) + g_k(x_k)u_k) + \alpha_{ke}(b_e) \right) \geq 0. \tag{6}$$

Following [7, Theorem 3], safety constraints can be incorporated in the CBF-QP formulation given by

$$J^* = \min_{\boldsymbol{u} \in \mathcal{U}} \sum_{i=1}^{N} ||u_i - u_i^{\text{des}}(x_i)||_2^2$$
$$\text{s.t.} \sum_{k \in \mathcal{V}_e} \left(\frac{\partial b_e}{\partial x_k}(f_k(x_k) + g_k(x_k)u_k) + \alpha_{ke}(b_e) \right) \geq 0,$$
$$\forall e = 1, \ldots, E, \tag{7}$$

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

$$J_i(u_i) = ||u_i - u_i^{\text{des}}(x_i)||_2^2,$$
$$h_{ie}(u_i) = -\left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right). \tag{8}$$

Assumption 8 *For every $\boldsymbol{x} \in \mathcal{B}$, there exists $\boldsymbol{u}(\boldsymbol{x}) \in \mathcal{U}$, such that $h_{ie} < 0$ for all $e = 1, \ldots, E$, $i \in \mathcal{V}_e$.*

Notice that, even not shown explicitly, $h_{ie}(u_i)$ depends on $x_i, i \in \mathcal{V}_e$. We also highlight that (7) is parameterized in $\boldsymbol{x}$, which can be thought of as constant as for the optimisation problem in (7) is concerned. Under Assumption 8, problem (7) is always feasible for all $x \in \mathcal{B}$. To begin with our analysis, we propose a relaxed version of (7) to guarantee feasibility of the local problems in the proposed distributed algorithm. This will be clarified in the sequel.

$$H^* = \min_{\boldsymbol{u} \in \mathcal{U}, \boldsymbol{\rho} \geq 0} H(\boldsymbol{u}, \boldsymbol{\rho}) :=$$
$$\sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \right\}$$
$$\text{subject to} \sum_{i \in \mathcal{V}_e} h_{ie}(u_i) \leq \rho_{ie}, e = 1, \ldots, E. \tag{9}$$

Let μ_e be the optimal dual solution for the constraint $\sum_{i \in \mathcal{V}_e} h_{ie}(u_i) \leq \rho_{ie}$. Feasibility of problem (9) is clear, as the positive variable $\boldsymbol{\rho}$ relaxes the linear constraints. Optimality is analyzed in the following lemma.

Lemma 9 *Denote the minimizer of problem (7) by $\boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$, and the minimizer of problem (9) by $\boldsymbol{u}_{\text{rel}}^*(\boldsymbol{x})$. Then $\boldsymbol{u}_{\text{rel}}^*(\boldsymbol{x}) = \boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$, and $\boldsymbol{\rho}^* = 0$ if*

$$M_i \geq \mu_e, \forall i \in \mathcal{V}_e, \forall e = 1, \ldots, E. \tag{10}$$

Besides, the cost function $H(\boldsymbol{u}, \boldsymbol{\rho})$ is strongly convex and has a Lipschitz continuous gradient.

PROOF. See Appendix.

3.1   Full Control Law

We now design an algorithm to solve the centralized CBF-QP problem (7) in a distributed manner; see Algorithm 1. Since $h_{ie}(u_i)$ also depends on x_k for $k \in \mathcal{V}_e \backslash \{i\}$, an additional communication round at the beginning of the algorithm is designed. For all $i = 1, \ldots, N$, and $e \in \mathcal{C}_i$, agent i is to receive x_k for any $k \in \mathcal{V}_e \backslash \{i\}$ 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 sub-networks $e \in \mathcal{C}_i$. Then, for any $e \in \mathcal{C}_i$, functions $h_{ie}(u_i)$ can be constructed as in (8).

There are two main computation and two communication steps in the algorithm. In the first computation step (Step 3), agent i solves the optimisation problem (11) to obtain the optimal primal-dual solution $((u_i^{k+1}, \boldsymbol{\rho}_i^{k+1}), \boldsymbol{\mu}_i^{k+1})$, 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 $e \in \mathcal{C}_i$ and $l \in \mathcal{N}_i \cap \mathcal{V}_e$. 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 (11) are relaxed by an additional non-negative relaxation variable ρ_{ie}. This guarantees the feasibility of the local optimisation problem by loosing the restriction of the original constraints. However, this does not necessarily imply satisfaction

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_k for any $k \in \mathcal{V}_e \backslash i$ from $l \in \mathcal{N}_i \cap \mathcal{V}_e$, for any $e \in \mathcal{C}_i$

Send x_i to any $l \in \mathcal{N}_i \cap \mathcal{V}_e$, for any $e \in \mathcal{C}_i$.

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** $((u_i^{k+1}, \boldsymbol{\rho}_i^{k+1}), \boldsymbol{\mu}_i^{k+1})$ as a primal-dual solution of the following optimisation problem

$$\min_{u_i, \boldsymbol{\rho}_i} J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie})$$

$$\text{s.t. } u_i \in \mathcal{U}_i, \rho_{ie} \geq 0,$$

$$h_{ie}(u_i) + \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \leq \rho_{ie}, \forall e \in \mathcal{C}_i.$$

$$(11)$$

4:     **Receive** μ_{le}^{k+1} from agent $l \in \mathcal{N}_i \cap \mathcal{V}_e$.
5:     **Update** λ_{il} by

$$\lambda_{il}^{k+1} = \lambda_{il}^k - \gamma^k (\mu_{ie}^{k+1} - \mu_{le}^{k+1}). \qquad (12)$$

6: **end while**

---

of the CBF constraints in (7) by using $\boldsymbol{u}^{k+1}$. The interpretation of this kind of infeasibility in CBF-QP application is that, there is no admissible control input that renders the agent system safe with the CBFs and given auxiliary variables.

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 11, 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 by agent i via communication in Step 2. We note here that for all $l \in \mathcal{N}_i \cap \mathcal{V}_e$, λ_{il} and λ_{li} are all scalars, hence the communication burden will not be high. The second computation step is to update the local auxiliary variables by means of (5). Part of the dual variables used in the update are received from the neighbours in the second communication round, i.e. Step 4. Here the update is a gradient-like procedure, with stepsize $\gamma > 0$. The dual variables will be bounded provided that the auxiliary variables are also bounded.

Algorithm 1 is fully distributed, where the two computation and communication steps can be carried out locally by each agent. Differently from the setting in [23, Algorithm RSDD], the relaxation penalty in the cost includes now a quadratic term. This renders the cost function strongly convex, allowing for superior convergence

properties and ensuring the minimizer u_i^* is unchanged.

We directly have the following *optimality* and *safety* results.

Theorem 10 *For every $\boldsymbol{x} \in \mathcal{B}$, the optimal distributed control input $\boldsymbol{u}^*(\boldsymbol{x})$ returned by Algorithm 1 coincides with the optimal centralized control input solved using (7). Besides, the optimal distributed control input renders $\mathcal{B}$ invariant.*

Among different types of distributed optimisation algorithms, [23, Algorithm RSDD] is selected here for its ability to guarantee almost-safety in 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 optimisation problem for feasibility. In high-frequency applications, the algorithm may stop before reaching convergence. When the relaxation variables $\boldsymbol{\rho}^k = 0$ for a $k > 0$, then for any $e = 1, \ldots, E$ 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 any control input solving (11) at iteration k. The next theorem gives the convergence result.

Theorem 11 *Consider Assumption 8, and let $M_i \geq \mu_e$ for every $i = 1, \ldots, N$, $e = 1, \ldots, E$. For every $i = 1, \ldots, N$, and any initialized $\boldsymbol{\lambda}^0$,*

(a) *if $\mathcal{U}_i \subset \mathbb{R}^{m_i}$. Choose the sequence $\{\gamma^k\}_{k \geq 0}$, with each $\gamma^k \geq 0$, 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 (7).*

(b) *if $\mathcal{U}_i = \mathbb{R}^{m_i}$. Let the step size $\gamma^k = \gamma$ be small as (A.7). $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k)$ converges to the optimal cost J^* in (7) 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 (7).*

PROOF. See Appendix.

Under certain regularity condition, [23, Algorithm RSDD] only guarantees local sublinear convergence [22] since $\nabla d(\boldsymbol{\lambda})$ is only guaranteed to be bounded [22, Proposition 5.2]. Lipschitz continuity of $\nabla d(\boldsymbol{\lambda})$ is important to establish a global sublinear convergence rate.

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 actuation 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_k for any $k \in \mathcal{V}_e \backslash i$ from $l \in \mathcal{N}_i \cap \mathcal{V}_e$, for any $e \in \mathcal{C}_i$

    Send x_i to any $l \in \mathcal{N}_i \cap \mathcal{V}_e$, for any $e \in \mathcal{C}_i$

    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 $\boldsymbol{\rho}^\eta$. In general, it is challenging to provide an explicit bound for η, under which $\boldsymbol{\rho}^\eta = 0$, as the distributed algorithm converges asymptotically as per Theorem 11. Moreover, $\boldsymbol{\rho}^\eta$ depends on the state $\boldsymbol{x} \in \mathcal{H} := \mathcal{X} \cap \mathcal{B}$, which parameterizes the optimization problem (9). To address this challenge, we study the problem that *given η, we characterize the confidence (probability) with which* $\mathbb{P}\{x \in \mathcal{H} : \boldsymbol{\rho}^\eta(\boldsymbol{x}) \neq 0\}$. 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, using a given feedback controller $\boldsymbol{u}(\boldsymbol{x})$. The verification is conducted by checking the *risk* of becoming unsafe along the current trajectories by means of the CBFs using the so called scenario approach. We tend to measure the violations of the CBF constraints to estimate the trend of being unsafe for the system. 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.

Consider an N-agent system (4) and a safe invariant set $\mathcal{B}$. Our objective is to verify whether for the designed $\boldsymbol{u}(\boldsymbol{x}(t))$, $\boldsymbol{x}(t) \in \mathcal{S}$, for all $t > 0$ and for all $\boldsymbol{x}(0) \in \mathcal{H} = \mathcal{B} \cap \mathcal{X}$.

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

$$\min_{\boldsymbol{z} \leq 0, \boldsymbol{\zeta} \geq 0} \sum_{i=1}^{N} \sum_{e \in \mathcal{C}_i} \left(z_{ie}^2 + M \sum_{r=1}^{\bar{N}} \zeta_{ie}^{(r)} \right)$$

(SC-Verification)

$$\text{s.t.} \sum_{i \in \mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x}^{(r)})) \leq \sum_{i \in \mathcal{V}_e} (z_{ie} + \zeta_{ie}^{(r)}),$$

$$\forall e = 1, \ldots, E, \forall r = 1, \ldots, \bar{N}, \qquad (13)$$

where scenarios $\boldsymbol{x}^{(r)} \in \mathcal{H}$ for any $r = 1, \ldots, \bar{N}$ are extracted according to some probability distribution to be clarified in the sequel. Throughout the section $\bar{X} = \{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(\bar{N})}\}$ denotes the set of scenarios, where $\boldsymbol{x}^{(r)} = [(x_1^{(r)})^\top, \ldots, (x_N^{(r)})^\top]^\top \in \mathbb{R}^{\sum_{i=1}^{N} n_i}$, for $r = 1, \ldots, \bar{N}$. Relaxation variables $\boldsymbol{\zeta}$ are introduced to ensure feasibility, while $M > 0$ is a large enough penalty coefficient.

Program (SC-Verification) is a data-driven QP, where all the constraints are linear based on the samples. Roughly speaking, if for any $\boldsymbol{x} \in \mathcal{H}$ and 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. A new set $\mathcal{Z}(\mathcal{H})$ for optimal solution $\boldsymbol{z}^*$ is defined as follow

$$\boldsymbol{z}^* \in \mathcal{Z}(\mathcal{H}) \Longleftrightarrow$$
$$\sum_{i \in \mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x})) \leq \sum_{i \in \mathcal{V}_e} z_{ie}^*, \forall e = 1, \ldots, E, \forall \boldsymbol{x} \in \mathcal{H}. \ (14)$$

Then $\mathcal{Z}(\mathcal{H})$ is constituted of N individual set $\mathcal{Z}_i(\mathcal{H})$ as

$$\mathcal{Z}(\mathcal{H}) = \bigcap_{i=1}^{N} \mathcal{Z}_i(\mathcal{H}). \qquad (15)$$

The argument of $\mathcal{Z}$ and $\mathcal{Z}_i$ is dropped in the sequel for simplicity.

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 6. Then, $\boldsymbol{x}$ can be sampled $\bar{N}$ 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 results of [30, Section 3.1]. Although the uniform distribution here is well-defined, the set $\mathcal{H}$ is defined implicitly as an intersection of multiple sets. Sampling a point from the proposed uniform distribution is rather arduous in practice, and agents may not have access to $\mathcal{H}$. Here, we provide a sequential algorithm to sample scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, \bar{N}$.

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 $\pi_1(x)$.
2: **for** $i = 2, \ldots, N$ **do**
3:     Construct a set $\mathcal{H}_i = \cap_{e \in \mathcal{C}_i} \mathcal{H}_{ie}$ following (16).
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**

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.

$$\mathcal{H}_{ie} = \begin{cases} \mathcal{X}_i, \text{ if } \exists l \in \mathcal{V}_e, \text{such that } l > i \\ \{x_i \in \mathcal{X}_i | b_{ie}(x_i, \{x_l^{(r)}\}) \geq 0)\}, \text{ otherwise} \end{cases} \tag{16}$$

We then have that $\mathcal{H}_i = \bigcap_{e \in \mathcal{C}_i} \mathcal{H}_{ie}$. The parameters in (16) can all be collected by local communication, since only states of agents in the same sub-network are required. Note here $\mathcal{H}_i$ is possibly empty with some parameters $\{x_1^{(r)}, \ldots, x_{i-1}^{(r)}\}$.

At 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

$i = 2, \ldots, N$, before sampling the scenario $x_i^{(r)}$, we first check whether $\mathcal{H}_i$ is empty (Step 4). By Assumption 6, there exists $\{x_1^{(r)}, \ldots, x_{i-1}^{(r)}\}$ such that $\mathcal{H}_i \neq \emptyset$. Therefore, if $\mathcal{H}_i = \emptyset$ (Step 5), then go back to the sampling-construction of agent $i - F$, $F \neq 1$ is to avoid a deadlock on step i. The deadlock happens when for given scenarios $x_1^{(r)}, \ldots, x_{i-2}^{(r)}$, 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 $x_1^{(r)}, \ldots, x_{i-1}^{(r)}$, 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 $\mathcal{H}_i \cap \mathcal{X} \neq \emptyset$.

Proposition 12 *The scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, \bar{N}$, are feasible, i.e., $\boldsymbol{x}^{(r)} \in \mathcal{H}$, and independent.*

PROOF. The feasibility result holds directly from the definition of every set $\mathcal{H}_i$ in (16) that $x_i^{(r)}$ is sampled from. As a result, we have $b_{ie}(x_i^{(r)}, \{x_k^{(r)}\}) \geq 0$ for any $i = 1, \ldots, N$, $e \in \mathcal{C}_i$, and $k \in \mathcal{V}_e$. Therefore, $\boldsymbol{x}^{(r)} \in \mathcal{H}$. $\boldsymbol{x}^{(r)}$ for $r = 1, \ldots, \bar{N}$ are independent since for $r = 1, \ldots, \bar{N}$, $x_1^{(r)}$ are independently sampled from distribution π_1.

We note here that the elements in $\boldsymbol{x}^{(r)}$ are correlated, but this will not influence the independence results in Proposition 12 since we seek independence across r.

4.3 Distributed Safety Verification

After sampling scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, \bar{N}$ using Algorithm 3, we are at the stage of solving the safety verification program (SC-Verification).

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

$$
\begin{aligned}
J_i(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) &= \sum_{e \in \mathcal{C}_i} \left(z_{ie}^2 + \sum_{r=1}^{\bar{N}} \zeta_{ie}^{(r)} \right), \\
\hat{h}_{ie}^{(r)}(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) &= h_{ie}(u_i(\boldsymbol{x}^{(r)})) - z_{ie} - \zeta_{ie}^{(r)}, r = 1, \ldots, \bar{N},
\end{aligned}
\tag{17}
$$

Algorithm 1 can be applied to solve the distributed scenario optimisation problem (SC-Verification). The relaxation variables in Algorithm 1 are unnecessary, since every optimisation sub-problem in iteration is solvable. In the sequel, we use $\boldsymbol{z}^*$ and $\boldsymbol{\zeta}^*$ to represent the optimal solution to (SC-Verification), with scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, \bar{N}$. We then have the following theorem as the main result on probabilistic safety.

Theorem 13 *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 $0 \leq s_i^* \leq \bar{N} - 1$, consider the polynomial equation in t_i*

$$
\binom{\bar{N}}{s_i^*} t_i^{\bar{N}-s_i^*} - \frac{\beta_i}{2\bar{N}} \sum_{j=s_i^*}^{\bar{N}-1} \binom{j}{s_i^*} t_i^{j-s_i^*}
$$
$$
- \frac{\beta_i}{6\bar{N}} \sum_{j=\bar{N}+1}^{4\bar{N}} \binom{j}{s_i^*} t_i^{j-s_i^*} = 0, \tag{18}
$$

while for $s_i^* = \bar{N}$ consider the polynomial equation

$$
1 - \frac{\beta}{6N} \sum_{j=\bar{N}+1}^{4\bar{N}} \binom{j}{s_i^*} t_i^{j-\bar{N}} = 0. \tag{19}
$$

For any $i = 1, \ldots, N$, this equation has exactly two solutions in $[0, +\infty)$ denoted by $\underline{t}_i(s_i^*)$ and $\bar{t}_i(s_i^*)$, where $\underline{t}_i(s_i^*) \leq \bar{t}_i(s_i^*)$. Let $\underline{\epsilon}_i(s_i^*) := \max\{0, 1 - \bar{t}_i(s_i^*)\}$, $\bar{\epsilon}_i(s_i^*) := 1 - \underline{t}_i(s_i^*)$, 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

$$
\mathbb{P}^{\bar{N}} \left\{ \frac{\underline{\epsilon}(s^*)}{N} \leq \mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : 0 \notin \mathcal{Z}\} \leq \bar{\epsilon}(s^*) \right\} \geq 1 - \beta, \tag{20}
$$

where s_i^* is the number of non-zero $\zeta_{ie}^{(r)*}$, $e \in \mathcal{C}_i$.

PROOF. See Appendix.

Note that Theorem 13 constitutes a generalization of [30, Theorem 2] to a multi-agent setting. It also extends [21] by determining the lower bound $\frac{\underline{\epsilon}(s^*)}{N}$. Theorem 13 states that with confidence $1 - \beta$, the system tends to be unsafe by means of the CBFs with probability within the interval $[\frac{\underline{\epsilon}(s^*)}{N}, \bar{\epsilon}(s^*)]$.

Furthermore, for a given r, (SC-Verification) can be split into $\sum_{i=1}^{E} |\mathcal{V}_e|$ sub-problems, each one with its own CBF constraint. Each sub-problem is solved at the agent level and has only $\bar{N}$ constraints. Then, the probability that one of the CBF constraints is violated can be bounded as shown in the following corollary.

Corollary 14 *Consider the multi-agent system (4), and let $\underline{\epsilon}_i(s_i^*)$, $\bar{\epsilon}_i(s_i^*)$, and β_i as in Theorem 13. We then have that*

$$
\mathbb{P}^{\bar{N}} \left\{ \underline{\epsilon}_i(s_i^*) \leq \mathbb{P}\left\{ \boldsymbol{x} \in \mathcal{H} : \sum_{i \in \mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x}^{(r)})) > 0 \right\} \leq \bar{\epsilon}_i(s_i^*) \right\}
$$
$$
\geq 1 - \beta_i. \tag{21}
$$

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 [13].

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 group of ten robots, indexed by $i = 1, \ldots, 10$ are considered, with double integrator dynamics

$$
\begin{bmatrix} \dot{\boldsymbol{p}}_i \\ \dot{\boldsymbol{v}}_i \end{bmatrix} = \begin{bmatrix} 0 & I_{2\times2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_i \\ \boldsymbol{v}_i \end{bmatrix} + \begin{bmatrix} 0 \\ I_{2\times2} \end{bmatrix} \boldsymbol{a}_i, \tag{22}
$$

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 $||\boldsymbol{a}_i||_\infty \leq a_i^{\max}$. $a_i^{\max}$ will be cleared in the sequel. Each robot is regarded as a disk centered 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

$$
s_{ij}(\boldsymbol{p}, \boldsymbol{v}) = ||\Delta \boldsymbol{p}_{ij}||_2^2 - D_{ij}, \tag{23}
$$

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 [13], the control barrier function for invariance certificates is then defined pair-wisely, as

$$
b_{ij}(\boldsymbol{p}, \boldsymbol{v}) = \sqrt{2(a_i^{\max} + a_j^{\max})(||\Delta \boldsymbol{p}_{ij}||_2^2 - D_{ij})}
$$
$$
+ \frac{\Delta \boldsymbol{p}_{ij}^\top}{||\Delta \boldsymbol{p}_{ij}||_2^2} \Delta \boldsymbol{v}_{ij}, \tag{24}
$$

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 $\mathcal{B} = \bigcap_{\{i,j\} \in \mathcal{E}} \mathcal{B}_{ij}$ is possibly empty. Intuitively, this is since robots cannot utilize 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 (24).

5.2 Distributed Control: Asymptotic Algorithm

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

(a) Time 1   (b) Time 100

(c) Time 200   (d) Time 300
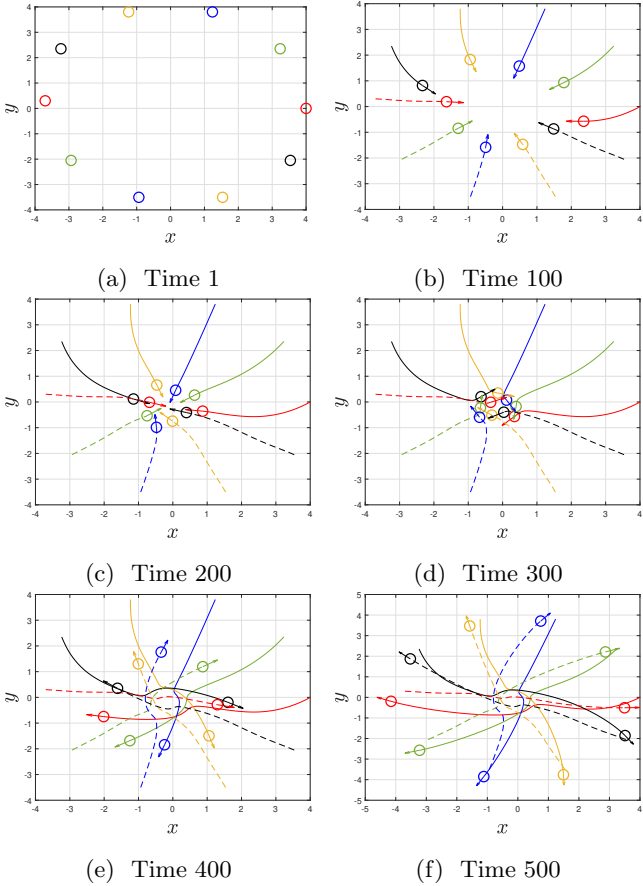
(e) Time 400   (f) Time 500

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

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 Figure 2. Define

$$\rho_{\text{sum}}^k = \sum_{i=1}^N \sum_{e \in \mathcal{C}_i} \left((\rho_{ie}^k)^2 + M_i \rho_{ie}^k \right). \qquad (25)$$

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 Figures 3a and 3b. It can be seen that ρ_{sum}^{30} is close to zero at every time step, even ρ_{sum}^0 is relatively

(a) Time 1   (b) Time 100

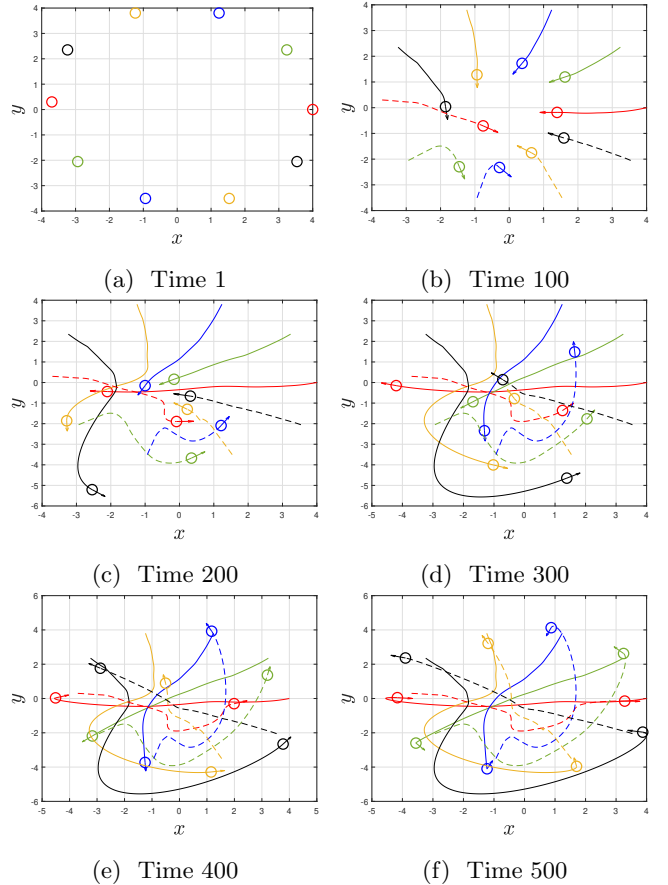(c) Time 200   (d) Time 300

(e) Time 400   (f) Time 500

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

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.
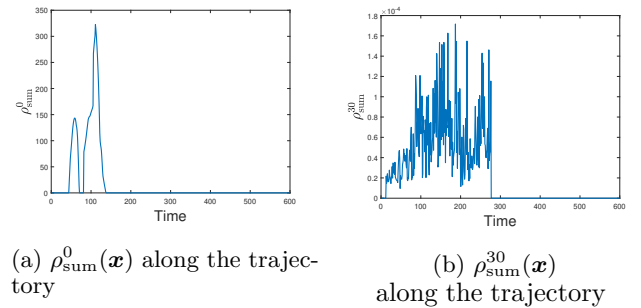

(a) $\rho_{\text{sum}}^0(\boldsymbol{x})$ along the trajectory   (b) $\rho_{\text{sum}}^{30}(\boldsymbol{x})$ along the trajectory

Fig. 3. Evolution of the relaxation parameters $\rho_{\text{sum}}^0(\boldsymbol{x})$ and $\rho_{\text{sum}}^{30}(\boldsymbol{x})$ evaluated at the state trajectory, across algorithm iterations.

5.4 Distributed Safety Verification

Safety verification is performed for a four-robot system, within the working space $\mathcal{X}$, defined as $\{p : ||p|| \le p^{\max} = 6\} \times \{v : ||v|| \le v^{\max} = 1\}$. Each

robot is using Algorithm 2 to safely move towards the origin. We sample 200 scenarios via Algorithm 3. Theorem 13 yields then that with confidence at least 0.9, $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : 0 \notin \mathcal{Z}\} \in [0, 0.146]$. We repeat this procedure 300 times, each time using 300 scenarios, and construct the empirical cumulative distribution function of $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : 0 \notin \mathcal{Z}\}$. This is shown in Figure 4; it can be observed that the empirical probability that $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : 0 \notin \mathcal{Z}\} \in [0, 0.146] \approx 1$, thus satisfying the theoretical confidence lower bound of 0.9.
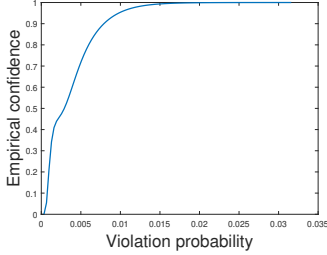

Fig. 4. Cumulative distribution function for safety violation.

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 establish a sublinear convergence rate. We also addressed the problem of distributed safety verification for given control inputs. A scenario-based verification program is formulated and can be solved locally by each agent. The scenarios are sampled independently by a sequential algorithm from the controlled invariant set. 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 determines the efficacy of our result. Current work concentrates in accounting for communication delays and model uncertainty in real systems.

A Appendix

PROOF. [Proof of Lemma 9] The dual function of (9) is given by

$$q(\boldsymbol{\mu}) = \inf_{\{u_i \in \mathcal{U}_i\}, \boldsymbol{\rho} \geq 0} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \right\}$$
$$+ \sum_{e=1}^{E} \mu_e \left\{ \sum_{i \in \mathcal{V}_e} h_{ie}(u_i) - \sum_{i \in \mathcal{V}_e} \rho_{ie} \right\}$$

$$= \inf_{\{u_i \in \mathcal{U}_i\}, \boldsymbol{\rho} \geq 0} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} \mu_e h_{ie}(u_i) \right\}$$
$$+ \sum_{e=1}^{E} \sum_{i \in \mathcal{V}_e} \left\{ \rho_{ie}^2 + (M_i - \mu_e)\rho_{ie} \right\} \qquad (A.1)$$

If there exists $e = 1, \ldots, E$, and $i \in \mathcal{V}_e$, such that $M_i - \mu_e < 0$, then the minimizer of $\sum_{e=1}^{E} \sum_{k \in \mathcal{V}_e} \{\rho_{ie}^2 + (M_i - \mu_e)\rho_{ie}\}$ is given by $\boldsymbol{\rho}^* > 0$. Notice that this is an unconstrained QP, so its minimizer can be determined by setting its gradient with respect to ρ_{ie} to zero. On the other hand if $M_i - \mu_e \geq 0$, for all $e = 1, \ldots, E, i \in \mathcal{V}_e$, the unconstrained minimizer for $\boldsymbol{\rho}$ is negative, hence the constrained minimizer $\boldsymbol{\rho}^* = 0$ as $\boldsymbol{\rho} \geq 0$. For this case, we obtain

$$q(\boldsymbol{\mu}) = \inf_{\{u_i \in \mathcal{U}_i\}} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} \mu_e h_{ie}(u_i) \right\}, \quad (A.2)$$

which is the dual function for the original unrelaxed problem (7). Recall that (7) is a strictly convex quadratic problem, then $\boldsymbol{u}_{\text{rel}}^*(\boldsymbol{x}) = \boldsymbol{u}_{\text{nom}}^*(\boldsymbol{x})$ if $M_i \geq \mu_e$, for all $e = 1, \ldots, E, i \in \mathcal{V}_e$ since $\boldsymbol{\rho}^* = 0$. The cost function $H(\boldsymbol{u}, \boldsymbol{\rho})$ is strongly convex with Lipschitz continuous gradient since every $J_i(u_i)$ is a strongly convex quadratic function with respect to u_i, and $\sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie})$ is strongly convex with respect to ρ_{ie}.

PROOF. [Proof of Theorem 11] We begin with (a). The proof follows a similar idea as [40, Theorem II.6], we only scratch the primal-dual exploration here. Under Assumption 8, strongly duality holds for the primal problem (7) and the dual problem (A.1). To solve the dual problem $\max_{\boldsymbol{\mu} \geq 0} q(\boldsymbol{\mu})$ in a distributed manner, an equivalent decomposed problem [40, Section 3.1.3] is formulated as

$$\max_{\boldsymbol{\mu_i} \geq 0} \sum_{i=1}^{N} q_i(\boldsymbol{\mu_i}) \qquad (A.3)$$
$$\text{subject to } \boldsymbol{\mu}_i = \boldsymbol{\mu}_j, (i, j) \in \mathcal{E},$$

where $\boldsymbol{\mu}_i$ is a local copy of $\boldsymbol{\mu}$ for agent i. The dual problem of (A.3) is given by

$$d^* = \min_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}) := \sum_{i=1}^{N} \sup_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \boldsymbol{\mu}_i^\top \sum_{j \in \mathcal{N}_i} (\lambda_{ij} - \lambda_{ji}) \right),$$
$$(A.4)$$

where λ_{ij} is a free dual variable for the constraint $\boldsymbol{\mu}_i = \boldsymbol{\mu}_j$. Strong duality holds between problem (A.3) and (A.4) since (A.3) is an equality constrained concave problem. Using the gradient descent method to solve (A.4) at each iteration k, each agent i performs two

steps:

- (i) calculates the gradient $\nabla d(\boldsymbol{\lambda}^k)$: receive $\lambda_{ji}^k, j \in \mathcal{N}_i$, and compute $\boldsymbol{\mu}_i^{k+1}$ by solving

$$\max_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \boldsymbol{\mu}_i^\top \sum_{j \in \mathcal{N}_i} (\lambda_{ij}^k - \lambda_{ji}^k) \right) \qquad (A.5)$$

- (ii) uses gradient descent: receive $\boldsymbol{\mu}_j^{k+1}$, and update λ_{ij} by:

$$\lambda_{ij}^{k+1} = \lambda_{ij}^k - \gamma^k (\boldsymbol{\mu}_i^{k+1} - \boldsymbol{\mu}_j^{k+1}). \qquad (A.6)$$

Diminishing step-size is used here as [23]. Specifically, (A.5) is the dual problem of (11). Strong duality holds for large enough $\boldsymbol{\rho}$ as the relaxed CBF constraints hold strictly. Updating (A.6) is the same as (12) for every agent in iterations. Given that $d(\boldsymbol{\lambda})$ is convex, gradient descent guarantees that $d(\boldsymbol{\lambda}^k)$ convergence to the optimal value $d^* = J^*$ since strong duality holds between (9) and (A.3), as well as (A.3) and (A.4). Moreover, the relaxed problem (9) is strongly (hence also strictly) convex, which indicates uniqueness of the optimal solution $(\boldsymbol{u}_{\text{rel}}^*, \boldsymbol{\rho}^*)$. Using Lemma 9, we obtain $\boldsymbol{u}_{\text{rel}}^* = \boldsymbol{u}_{\text{nom}}^*$, which is the optimal solution of (7).

We then prove (b). We show that $d(\boldsymbol{\lambda})$ is a convex quadratic function. First we prove that for every $i = 1, \ldots, N$, $q_i(\boldsymbol{\mu}_i)$ in (A.3) is a concave quadratic function. When every $\mathcal{U}_i = \mathbb{R}^{m_i}$, the relaxed CBF-QP (9) is a linearly constrained quadratic problem. Following the example [41, Section 5.2.4, Eq. 5.28] [1], every $q_i(\boldsymbol{\mu}_i)$ is a concave quadratic function. Therefore, problem (A.3) is a linearly constrained concave quadratic problem. Following [42, Example 3.6] $d(\boldsymbol{\lambda})$ is a convex quadratic function, which is necessarily smooth. Using constant step size

$$0 < \gamma < \frac{1}{2L}, \qquad (A.7)$$

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

$$d(\boldsymbol{\lambda}^k) - J^* \leq \frac{2(d(\boldsymbol{\lambda}^0) - J^*)||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2}{2||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2 + k\gamma(2 - L\gamma)(d(\boldsymbol{\lambda}^0) - J^*)}$$

---

[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.

$$\leq \frac{2(d(\boldsymbol{\lambda}^0) - J^*)||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2}{k\gamma(d(\boldsymbol{\lambda}^0) - J^*)} \leq \frac{2||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2}{k\gamma}. \tag{A.8}$$

The first inequality is proved by [43, theorem 2.1.4], the second one comes from eliminating the term $||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2$, and considering $2 - L\gamma \geq 1$ from (A.7). The Lipschitz constant can be calculated by the largest eigenvalue of the quadratic term for $d(\boldsymbol{\lambda})$. Although the expression of $d(\boldsymbol{\lambda}^k)$ is not derived, $d(\boldsymbol{\lambda}^k)$ can be calculated by

$$d(\boldsymbol{\lambda}^k) = \sum_{i=1}^N \sup_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \boldsymbol{\mu}_i^\top \sum_{j \in \mathcal{N}_i} (\lambda_{ij} - \lambda_{ji}) \right)$$
$$= \sum_{i=1}^N \sup_{\boldsymbol{\mu}_i \geq 0} \left(\inf_{u_i, \boldsymbol{\rho}_i \geq 0} \left(J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \right) \right.$$
$$\left. + \sum_{e \in \mathcal{C}_i} \mu_{ie}(h_{ie}(u_i) - \rho_{ie}) + \boldsymbol{\mu}_i^\top \sum_{j \in \mathcal{N}_i} (\lambda_{ij}^k - \lambda_{ji}^k) \right). \tag{A.9}$$

For every $i = 1, \ldots, N$, and $\boldsymbol{\lambda}^k$, there always exists $\boldsymbol{u}_i$ and $\boldsymbol{\rho}_i$, such that the inequality constraint in (4) strictly holds, which indicates strong duality. Therefore, we have

$$d(\boldsymbol{\lambda}^k) =$$
$$\sum_{i=1}^N \inf_{u_i, \boldsymbol{\rho}_i \geq 0} \left(\sup_{\boldsymbol{\mu}_i \geq 0} \left(J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \right) \right.$$
$$\left. + \sum_{e \in \mathcal{C}_i} \mu_{ie}(h_{ie}(u_i) - \rho_{ie}) + \boldsymbol{\mu}_i^\top \sum_{j \in \mathcal{N}_i} (\lambda_{ij}^k - \lambda_{ji}^k) \right)$$
$$= \sum_{i=1}^N \left(J_i(u_i^k + \sum_{e \in \mathcal{C}_i} ((\rho_{ie}^k)^2 + M_i \rho_{ie}^k) \right)$$
$$= \sum_{i=1}^N ||u_i^k - u^{\text{des}}||^2 + \rho_{\text{sum}}^k = H(\boldsymbol{u}^k, \boldsymbol{\rho}^k). \tag{A.10}$$

Hence, by (A.9) and (A.10) we conclude that $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k) - J^* < \frac{2||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2}{\gamma k}$.

PROOF. [Proof of Theorem 13] We have that

$$\mathbb{P}^{\bar{N}} \left\{ \frac{\sum_{i=1}^N \underline{\epsilon}_i(s_i^*)}{N} \leq \mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}\} \leq \sum_{i=1}^N \bar{\epsilon}_i(s_i^*) \right\}$$
$$= \mathbb{P}^{\bar{N}} \left\{ \frac{1}{N} \sum_{i=1}^N \epsilon_i(s_i^*) \leq \mathbb{P}\left\{ \boldsymbol{x} \in \mathcal{H} : \right. \right.$$
$$\left. \left. \exists i \in \{1, \ldots, N\}, \boldsymbol{z}^* \notin \mathcal{Z}_i \right\} \leq \sum_{i=1}^N \bar{\epsilon}_i(s_i^*) \right\}$$

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

$$\left. \bigcap \mathbb{P} \left\{ \bigcup_{i=1}^{N} \{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_i \} \right\} \leq \sum_{i=1}^{N} \bar{\epsilon}_i(s_i^*) \} \right\} \tag{A.11}$$

We separately deal with the inner and upper bounds on the probability. For the upper bound we have

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

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

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

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

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

The equality is achieved if for any $i \neq j$, $\boldsymbol{z}^* \neq \mathcal{Z}_i \Leftrightarrow \boldsymbol{z} \neq \mathcal{Z}_j$ and $\underline{\epsilon}_i(s_i^*) = \underline{\epsilon}_j(s_j^*)$. The right-hand side of (A.11) can be then lower-bounded by

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

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

$$\geq \mathbb{P}^{\bar{N}} \left\{ \bigcap_{i=1}^{N} \left\{ \underline{\epsilon}_i(s_i^*) \leq \mathbb{P} \left\{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_i \right\} \leq \bar{\epsilon}_i(s_i^*) \right\} \right\}$$

$$\geq 1 - \sum_{i=1}^{N} \mathbb{P}^{\bar{N}} \left\{ \bar{\epsilon}_i(s_i^*) < \mathbb{P} \left\{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_i \right\} \right.$$

$$\left. \bigcup \mathbb{P} \left\{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_i \right\} < \underline{\epsilon}_i(s_i^*) \right\}. \tag{A.12}$$

By [30, Theorem 1] we have that for any $i = 1, \ldots, N$

$$\mathbb{P}^{\bar{N}} \left\{ \boldsymbol{x} \in \mathcal{H} : \underline{\epsilon}_i(s_i^*) \leq \mathbb{P} \left\{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_i \right\} \leq \bar{\epsilon}_i(s_i^*) \right\}$$

$$\geq 1 - \beta_i$$

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

$$\bigcup \mathbb{P} \left\{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_i \right\} < \underline{\epsilon}_i(s_i^*) \right\} < \sum_{i=1}^{N} \beta_i. \tag{A.13}$$

Since $\frac{\underline{\epsilon}(s^*)}{N} < \underline{\epsilon}(s^*) < \bar{\epsilon}(s^*)$, substituting (A.13) into (A.11) with $i = 1, \ldots, N$ we obtain

$$\mathbb{P}^{\bar{N}} \left\{ \frac{\underline{\epsilon}(s^*)}{N} \leq \mathbb{P} \left\{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z} \right\} \leq \bar{\epsilon}(s^*) \right\} \geq 1 - \beta. \tag{A.14}$$

We then prove that $\boldsymbol{z}^*$ is unique, and $\boldsymbol{z}^* = 0$. For the case where all the CBF constraints are satisfied, i.e. $\sum_{k \in \mathcal{V}_e} h_{ke}(u_k(\boldsymbol{x}^{(r)})) \leq 0, \forall e = 1, \ldots, E$, $r = 1, \ldots, \bar{N}$, we have that $\boldsymbol{z}^* = 0$ and $\boldsymbol{\zeta}^* = 0$. For the case where there exists violated CBF constraint, i.e. $\sum_{i \in \mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x}^{(r)})) > 0$, we have that $z_{ie}^* = 0$ since $\boldsymbol{z} \leq 0$, and $\zeta_{ie}^* > 0$ for $i \in \mathcal{V}_e$. In summary, we always have $\boldsymbol{z}^* = 0$ for any scenarios, thus (A.14) is equivalent to (20). In addition, we directly obtain that $\zeta_{ie}^* > 0$ shows that $\sum_{i \in \mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x}^{(r)})) > z_{ie}^* = 0$. Thus, for every agent, s_i^* is the number of non-zero ζ_{ie}^*, for $e \in \mathcal{C}_i$.

References

[1] H. Wang, A. Papachristodoulou, and K. Margellos, "Distributed safety verification for multi-agent systems," *62nd IEEE Conference on Decision and Control*, 2023.

[2] F. Ding, J. He, Y. Ren, H. Wang, and Y. Zheng, "Configuration-aware safe control for mobile robotic arm with control barrier functions," *arXiv preprint arXiv:2204.08265*, 2022.

[3] H. Wang, Y. Li, W. Yu, J. He, and X. Guan, "Moving obstacle avoidance and topology recovery for multi-agent systems," in *American Control Conference (ACC)*, pp. 2696–2701, IEEE, 2019.

[4] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1033–1045, 2016.

[5] A. Alam, A. Gattami, K. H. Johansson, and C. J. Tomlin, "Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations," *Control Engineering Practice*, vol. 24, pp. 33–41, 2014.

[6] W. Xiao and C. G. Cassandras, "Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees," *Automatica*, vol. 123, p. 109333, 2021.

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

[8] E. D. Sontag, "A 'universal' construction of artstein's theorem on nonlinear stabilization," *Systems & control letters*, vol. 13, no. 2, pp. 117–123, 1989.

[9] J. A. Primbs, V. Nevistić, and J. C. Doyle, "Nonlinear optimal control: A control lyapunov function and receding horizon perspective," *Asian Journal of Control*, vol. 1, no. 1, pp. 14–24, 1999.

[10] S.-C. Hsu, X. Xu, and A. D. Ames, "Control barrier function based quadratic programs with application to bipedal robotic

walking," in *2015 American Control Conference (ACC)*, pp. 4542–4548, IEEE, 2015.

[11] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, pp. 6271–6278, IEEE, 2014.

[12] Y. Chen, A. Singletary, and A. D. Ames, "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach," *IEEE Control Systems Letters*, vol. 5, no. 1, pp. 127–132, 2020.

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

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

[15] X. Xu, "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, 2018.

[16] X. Tan and D. V. Dimarogonas, "Distributed implementation of control barrier functions for multi-agent systems," *IEEE Control Systems Letters*, vol. 6, pp. 1879–1884, 2021.

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

[18] A. Falsone, I. Notarnicola, G. Notarstefano, and M. Prandini, "Tracking-admm for distributed constraint-coupled optimization," *Automatica*, vol. 117, p. 108962, 2020.

[19] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[20] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Transactions on Automatic control*, vol. 57, no. 3, pp. 592–606, 2011.

[21] K. Margellos, A. Falsone, S. Garatti, and M. Prandini, "Distributed constrained optimization and consensus in uncertain networks via proximal minimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 5, pp. 1372–1387, 2017.

[22] A. Camisa, F. Farina, I. Notarnicola, and G. Notarstefano, "Distributed constraint-coupled optimization via primal decomposition over random time-varying graphs," *Automatica*, vol. 131, p. 109739, 2021.

[23] I. Notarnicola and G. Notarstefano, "Constraint-coupled distributed optimization: A relaxation and duality approach," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 1, pp. 483–492, 2019.

[24] X. Li, G. Feng, and L. Xie, "Distributed proximal algorithms for multiagent optimization with coupled inequality constraints," *IEEE Transactions on Automatic Control*, vol. 66, no. 3, pp. 1223–1230, 2020.

[25] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[26] M. C. Campi and S. Garatti, "The exact feasibility of randomized solutions of uncertain convex programs," *SIAM Journal on Optimization*, vol. 19, no. 3, pp. 1211–1230, 2008.

[27] M. C. Campi and S. Garatti, "Wait-and-judge scenario optimization," *Mathematical Programming*, vol. 167, no. 1, pp. 155–189, 2018.

[28] G. Calafiore and M. C. Campi, "Uncertain convex programs: randomized solutions and confidence levels," *Mathematical Programming*, vol. 102, no. 1, pp. 25–46, 2005.

[29] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, 2006.

[30] S. Garatti and M. C. Campi, "Risk and complexity in scenario optimization," *Mathematical Programming*, pp. 1–37, 2019.

[31] K. Margellos and J. Lygeros, "Hamilton–Jacobi formulation for reach–avoid differential games," *IEEE Transactions on Automatic Control*, vol. 56, no. 8, pp. 1849–1861, 2011.

[32] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, 2004.

[33] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *International Workshop on Hybrid Systems: Computation and Control*, pp. 477–492, Springer, 2004.

[34] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A framework for worst-case and stochastic safety verification using barrier certificates," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

[35] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "Introducing sostools: A general purpose sum of squares programming solver," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 1, pp. 741–746, IEEE, 2002.

[36] S. Prajna, A. Papachristodoulou, P. Seiler, and P. A. Parrilo, "Sostools: Sum of squares optimization toolbox for matlab," 2004.

[37] P. Akella and A. D. Ames, "A barrier-based scenario approach to verifying safety-critical systems," *IEEE Robotics and Automation Letters*, 2022.

[38] A. Taly and A. Tiwari, "Deductive verification of continuous dynamical systems," in *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

[39] D. G. Wright, "Tychonoff's theorem," *Proceedings of the American Mathematical Society*, vol. 120, no. 3, pp. 985–987, 1994.

[40] G. Notarstefano, I. Notarnicola, A. Camisa, *et al.*, "Distributed optimization for smart cyber-physical networks," *Foundations and Trends® in Systems and Control*, vol. 7, no. 3, pp. 253–383, 2019.

[41] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[42] V. Guigues, "On the strong concavity of the dual function of an optimization problem," *arXiv preprint arXiv:2006.16781*, 2020.

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