Distributed Control Design and Safety Verification for Multi-Agent Systems

Han Wang, Antonis Papachristodoulou, Kostas Margellos

***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, with probabilistically guaranteed constraint satisfication, while generating a Lipschitz continuous control input. We further develop a distributed safety verification algorithm to quantify safety for a multi-agent system by means of CBFs in probability. 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.**

***Index Terms*—Distributed Optimisation, Scenario Approach, Safe Control, Multi-Agent Systems, Nonlinear Systems**

I. 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.

The authors are with the Department of Engineering Science, University of Oxford, Oxford, United Kingdom. E-mails: {han.wang, antonis, kostas.margellos}@eng.ox.ac.uk.

Part of the results of this manuscript is submitted to the 62nd IEEE Conference on Decision and Control [1]. We significantly extend the conference version by additionally evaluating a lower bound on the probability of safety, proposing a distributed safe controller design algorithm, and a truncated algorithm with rigorous safety analysis.

A. 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]. Higher-order derivative based methods for high relative degree systems are proposed in [12]–[14]. In [15], adaptive coefficients are introduced to improve the feasibility of the CBF-QP. For the case where multiple CBFs exist, an optimal decay based method is proposed to tune the CBF constraints [16]. CBFs for discrete time systems are proposed in [17]. For the case where model uncertainty and system noise are added, robust CBF with worst case analysis [18], [19] can be considered. 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 [20]–[22]. 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 [23], where the CBF constraints are dynamically tuned for compatibility. Optimality is further considered in [24], 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 based [25]–[28] and primal decomposition based ones [29]–[33]. Dual decomposition methods consider the dual problem, where

each agent maintains a local copy of the dual variables. Constraint satisfication is achieved by consensus over the dual variables. Primal decomposition methods directly decompose the primal problem into local problems. By local projection [29], [32], [33] or updating auxiliary variables [30], [31], 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 [30], [31], primal decomposition methods are leveraged.

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 [34], [35] 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 [36], [37]. 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 [38], [39], 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 [40] using sampled data was proposed recently. Probabilistically guaranteed safety is ensured using the so called scenario approach [41]–[45].

B. Contributions

Our first contribution is to provide a method for constructing a distributed, safe controller. The proposed algorithm offers a distributed implementation of solving a quadratic programming problem with CBF-based affine constraints [7]. To parallelize the computation, we leverage the primal decomposition method presented in [31] to decompose the coupling constraints via the introduction of auxiliary variables. In this way, each agent can solve a local optimisation problem with auxiliary variables iteratively, while the auxiliary variables are dynamically updated using the dual variables associated with the constraints. 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.

In the original framework [30], [31], asymptotic convergence is achieved for primal variables. Our method exhibits a linear convergence rate for the control input. This is realized by a properly determined quadratic term in every local cost function to enhance convexity and smoothness. The term is designed based only on a local tolerance parameter and local control ability for distributed computation. Moreover, the relaxation variables decrease monotonically across iterations until they reach the certain tolerance threshold. At a given state, the minimal number of iterations for reaching a given threshold can be calculated. The monotonicity and linear convergence results are rigorously analyzed.

To reduce the communication and computation burden, an truncation mechanism is proposed to allow us to terminate the algorithm prior to convergence. The main challenge here is to determine the truncation parameter under the promise of safety. Using the obtained monotonicity result, the upper bound of the truncation parameter can be calculated by solving an uncertain optimisation problem. Such a problem poses difficulties in computation, and parallelization. To address these issues, we leverage the so called scenario approach [41]–[45], which samples a number of independent states from the state space and enforces the constraint only at these realizations. The original program is therefore approximated by a certain convex optimisation problem. We show that the obtained upper bound is also feasible for unrealized states with certain probabilistic guarantees.

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 to quantify safety by means of CBFs. At a given state, if the CBF constraints are violated, then the system is under the risk of being unsafe. Following this principle, we propose a scenario-based verification algorithm for a probabilistic quantification of safety. The scenarios, however, are hard to be sampled independently for a multi-agent system, due to the large cardinality of the uncertain set and the high coupling on each dimension. 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 [45, Theorem 1] to a 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.

C. Organization

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

D. Notation

We use $\mathbb{R}$, $\mathbb{R}^N$, $\mathbb{R}_+$ to represent the space 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 $\alpha(\cdot) : (-b, a) \to (-\infty, +\infty)$ 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 nodes set $\mathcal{V}$ and edge set $\mathcal{E}$. Throughout the paper $\mathcal{S}$ is used for a safe set, $\mathcal{X}$, $\mathcal{U}$ are used for general constraint

sets over state and input, and $\mathcal{B}$ 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-supper level set, i.e., $\mathcal{S} := \{\boldsymbol{x} | s(\boldsymbol{x}) \geq 0\}$.

II. Distributed Safe Control Law

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

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \tag{1}$$

where $x_i(t) \in \mathcal{X}_i \subset \mathbb{R}^{n_i}$ denotes its state, $u_i \in \mathcal{U}_i \subset \mathbb{R}^{m_i}$ denotes its control input, while $\mathcal{X}_i$ and $\mathcal{U}_i$ are the state space and control admissible set, respectively. The dynamics $f_i(x_i) : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i}$ and $g_i(x_i) : \mathbb{R}^{n_i} \to \mathbb{R}^{n_i} \times \mathbb{R}^{m_i}$ are both locally Lipchistz-continuous. 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. 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 $\mathcal{V}_e \subseteq \mathcal{V}$. Each agent i can communicate and cooperate with its neighbour $j \in \mathcal{N}_i$ to stay safe inside group e by ensuring $s_e(\boldsymbol{x}_e) \geq 0$, where $s_e(\cdot) \in \mathbb{R}$. We let $\mathcal{C}_i$ to be the set of group constraints agent i participates in; then we have $\mathcal{V}_e = \{i | e \in \mathcal{C}_i\}$.

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

The overall invariant (safe) set is defined by the Cartesian intersection of all the individual safe (invariant) sets

$$\mathcal{B} = \bigcap_{e=1}^{E} \mathcal{B}_e. \tag{2a}$$

$$\mathcal{S} = \bigcap_{e=1}^{E} \mathcal{S}_e. \tag{2b}$$

The following proposition guarantees the existence of the invariant set $\mathcal{B}$ retrieved from $\mathcal{S}$ by the way in (2a).

Proposition 1. *For the networked system* (1) *and the safe set* (2b)*, there exists an invariant set $\mathcal{B} \subseteq \mathcal{S}$. Furthermore, for any invariant set $\mathcal{B} \subseteq \mathcal{S}$, there exists a series of invariants set $\mathcal{B}_e \in \mathcal{S}_e$, for $e = 1, \ldots, E$, such that* (2a) *holds.*

Proof. From [46, Lemma 1], there always exists a invariant set $\mathcal{B} \subseteq \mathcal{S}$. From the definition of $\mathcal{S}$ in (2b), we have that $\mathcal{B} \subseteq \mathcal{S}_e$, for any $e = 1, \ldots, E$. Using [46, Lemma 1] again, it is always possible to retrieve a maximal invariant set $\mathcal{B}_e$, from $\mathcal{S}_e$, with $\mathcal{B}_e \subseteq \mathcal{S}_e$ for any $e = 1, \ldots, E$. Suppose there exists an invariant set $\mathcal{B} \not\subseteq \mathcal{B}_e$. This establishes a contradiction as in this case $\mathcal{B} \cup \mathcal{B}_e \supset \mathcal{B}_e$ would also be invariant, contradicting the assumption that $\mathcal{B}_e$ is the maximal invariant set. Then we conclude that $\forall e = 1, \ldots, E$, $\mathcal{B} \subseteq \mathcal{B}_e$, thus establishing (2a). □

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

$$\begin{aligned} \min_{u_i \in \mathcal{U}_i} & \sum_{i=1}^{N} ||u_i - u_i^{\mathrm{des}}(x_i)||_2^2 \\ \text{s.t.} & \sum_{k \in \mathcal{V}_e} \left\{ \frac{\partial b_e}{\partial x_k}(f_k(x_k) + g_k(x_k)u_k) + \alpha_{ke}(b_e) \right\} \geq 0, \\ & \forall e = 1, \ldots, E, \end{aligned} \tag{3}$$

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 $u_i^{\mathrm{des}}(x_i)$ is a nominal stabilizing control input. Let

$$\begin{aligned} J_i(u_i) &= ||u_i - u_i^{\mathrm{des}}(x_i)||_2^2, \\ h_{ie}(u_i) &= -\left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right). \end{aligned} \tag{4}$$

Notice that, even not shown explicitly, $h_{ie}(u_i)$ depends on $x_i, i \in \mathcal{V}_e$. Communication between neighbouring agents is thus necessary for constructing $h_{ie}(u_i)$ in the optimisation problem. We also highlight that (3) is parameterized in $\boldsymbol{x}$, which can be thought of as constant as for the optimisation in (3) is concerned. For any $e = 1, \ldots, E$, and $i \in \mathcal{V}_e$, there exists class-$\mathcal{K}$ functions $\alpha_{ie}(\cdot)$ such that the CBF-QP problem (3) is feasible for any $\boldsymbol{x} \in \mathcal{B}$ [7, Proposition 3].

A. Full Control Law

We now design an algorithm to solve the centralized CBF-QP problem (3) 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 (4).

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 (5) to obtain the optimal primal-dual solution $((x_i^{k+1}, \boldsymbol{\rho}_i^{k+1}), \boldsymbol{\mu}_i^{k+1})$, where $\boldsymbol{\rho}_i$ includes relaxation variables denoted by ρ_{ie} (penalized in the cost by $M_i > 0$), 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 group 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 group constraints in the distributed problem (5) are relaxed by an additional nonnegative relaxation variable ρ_{ie}. This guarantees the feasibility of the local optimisation problem by loosing the restriction of the original constraints. The possible infeasibility without relaxations happens if $\sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) > \sup_{x_i \in \mathcal{X}_i} -h_{ie}(x_i)$, the constraint $h_{ie}(x_i) + \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \leq 0$ might not be satisfied. The interpretation of this kind of infeasibility in CBF-QP application is that, there is no

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

Initialization $\lambda_{il}^0, \forall l \in \mathcal{N}_i \cap \mathcal{V}_e$, $\forall e \in \mathcal{C}_i$ pre-defined.

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

$$\begin{aligned} \min_{u_i, \boldsymbol{\rho}_i} \ & J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \\ \text{s.t. } & u_i \in \mathcal{U}_i, \rho_{ie} \geq 0, \\ & h_{ie}(u_i) + \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \leq \rho_{ie}, \forall e \in \mathcal{C}_i. \end{aligned} \tag{5}$$

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

5: **Update** λ_{il} by

$$\lambda_{il}^{k+1} = \lambda_{il}^k - \gamma(\mu_{ie}^{k+1} - \mu_{le}^{k+1}). \tag{6}$$

6: **end while**

admissible control input that renders the system safe with the CBFs and auxiliary variables.

The first computation step uses auxiliary variables λ_{il}^k and λ_{li}^k which constitute estimates of the neighbouring terms $h_{le}(x_l)$. 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 [31, 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. These properties are illustrated in the following lemma.

Lemma 1. *Consider problem* (3) *and denote its minimizer by* $\boldsymbol{u}_{\text{cen}}^*(\boldsymbol{x})$. *The optimal control input returned by Algorithm 1 is denoted by* $\boldsymbol{u}_{\text{dis}}^*(\boldsymbol{x})$. *Then* $\boldsymbol{u}_{\text{dis}}^*(\boldsymbol{x}) = \boldsymbol{u}_{\text{cen}}^*(\boldsymbol{x})$ *if*

$$M_i \geq \mu_e, \forall i \in \mathcal{V}_e, \forall e = 1, \ldots, E. \tag{7}$$

Besides, the overall cost function $J(\boldsymbol{u}, \boldsymbol{\rho}) = \sum_{i=1}^N \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \right\}$ *is strongly convex and has a Lipschitz continuous gradient.*

We directly have the following *optimality* and *safety* results; the proof follows from Proposition 1.

Theorem 1. *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* (3). *Besides, the optimal distributed control input renders* $\mathcal{B}$ *invariant.*

Among different types of distributed optimisation algorithms, [31, 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 (5) at iteration k.

For each state $\boldsymbol{x}$, let

$$\rho_{\text{sum}}^k = \sum_{i=1}^N \sum_{e \in \mathcal{C}_i} \left((\rho_{ie}^k)^2 + M_i \rho_{ie}^k \right), \tag{8a}$$

$$P_i = \sup_{u_i \in \mathcal{U}_i} ||u_i - u_i^{\text{des}}(\boldsymbol{x})||^2. \tag{8b}$$

The next theorem gives the convergence result, and further points out a monotonicity property of the relaxation variables.

Theorem 2. *Consider Algorithm 1. There exists* $\theta \in (0, 1)$, *such that for any* $\epsilon_1, \ldots, \epsilon_N > 0$, *and* $k \in \mathbb{N}$, *if*

$$\sum_{e \in \mathcal{C}_i} \left(\frac{(\rho_{ie}^k)^2}{M_i} + \rho_{ie}^k \right) > \epsilon_i, \forall i = 1, \ldots, N, \tag{9a}$$

$$\frac{P_i}{M_i} < \frac{1 - \theta}{\theta} \epsilon_i, \forall i = 1, \ldots, N, \tag{9b}$$

then $0 \leq \rho_{\text{sum}}^{k+1} < \rho_{\text{sum}}^k$.

Note that $\theta, \rho_{\text{sum}}^k$, ρ_{sum}^{k+1}, u_i^k, u_i^{k+1} are functions of $\boldsymbol{x}$. We drop the arguments for simplicity. Theorem 2 establishes that under tolerance $\epsilon_1, \ldots, \epsilon_N$, ρ_{sum} decreases monotonically when parameters M_i, $i = 1, \ldots, N$ satisfy (9). The requirement of M_i in Lemma 1 is consistent with that of (9). With large enough M_i, (9a) is approximated by $\sum_{e \in \mathcal{C}_i} \rho_{ie}^k > \epsilon_i$, which represents the case that the local control law u_i violates the CBF constraint by at least ϵ_i, while (9b) and (7) hold.

For a general strongly convex and smooth cost function, Algorithm 1 only guarantees asymptotic convergence [31]. However, our problem is a strictly convex quadratic program, where the dual function, and the dual of the dual function are both strongly convex and smooth. Thus, the cost $\sum_{i=1}^N J_i(x_i) + \sum_{e \in \mathcal{C}_i} (M_i \rho_{ie} + \rho_{ie}^2)$ converges to the optimum $\sum_{i=1}^N J_i(u_i^*) + \sum_{e \in \mathcal{C}_i} (M_i \rho_{ie}^* + (\rho_{ie}^*)^2) = \sum_{i=1}^N J_i(u_i^*)$ with linear convergence rate. However, this does not necessarily

lead to a decrease of the relaxation variables across iterations, since $J_i(u_i^k)$ also appears in the cost. The following corollary further investigates the evolution of ρ_{sum}^k.

Corollary 2.1. *Assume* (9b) *holds for* $\epsilon_1, \dots, \epsilon_N > 0$*, and* $M_1, \dots, M_N > 0$ *and in addition:* Case I*:*

$$\sum_{e \in \mathcal{C}_i} \left(\frac{(\rho_{ie}^0)^2}{M_i} + \rho_{ie}^0\right) > \epsilon_i, \forall i = 1, \dots, N. \tag{10}$$

Then, for all $k \in \mathbb{N}$*,* ρ_{sum}^k *satisfies*

$$\rho_{\text{sum}}^k \leq \max\left\{\sum_{i=1}^N M_i \epsilon_i, \theta^k \rho_{\text{sum}}^0 + \frac{\theta - \theta^{k+1}}{1-\theta} \sum_{i=1}^N P_i\right\}. \tag{11}$$

Moreover, $\theta^k \rho_{\text{sum}}^0 + \frac{\theta - \theta^{k+1}}{1-\theta} \sum_{i=1}^N P_i$ *decreases monotonically and*

$$\lim_{k \to \infty} \theta^k \rho_{\text{sum}}^0 + \frac{\theta - \theta^{k+1}}{1-\theta} \sum_{i=1}^N P_i < \sum_{i=1}^N M_i \epsilon_i. \tag{12}$$

Case II*:*

$$0 \leq \sum_{e \in \mathcal{C}_i} \left(\frac{(\rho_{ie}^0)^2}{M_i} + \rho_{ie}^0\right) \leq \epsilon_i, \forall i = 1, \dots, N. \tag{13}$$

Then for all $k \in \mathbb{N}$*,* ρ_{sum}^k *satisfies*

$$\rho_{\text{sum}}^k \leq \sum_{i=1}^N M_i \epsilon_i. \tag{14}$$

Conditions (9), (10), (13) are all distributed, hence M_i, $i = 1, \dots, N$ can be designed by agents locally.

B. 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 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 is interrupted 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: **end while**
4: step 5 in Algorithm 1

The following theorem shows how to decide on a value for η to meet the desired tolerances $\epsilon_1, \dots, \epsilon_N$. To reduce conservatism we do this in a probabilistic manner, where we sample scenarios $\boldsymbol{x}^{(r)}, r = 1, \dots, \bar{N}$, independently from some distribution $\mathbb{P}$ and choose η on the basis of these samples. Moreover, we support this choice with a probabilistic certificate on its effect on $\rho_{\text{sum}}^\eta(\boldsymbol{x})$.

Theorem 3. *Consider* $\bar{N}$ *i.i.d. samples* $\boldsymbol{x}^{(r)} \in \mathcal{B}$*,* $r = 1 \dots, \bar{N}$*, extracted from some probability distribution* $\mathbb{P}$*, and let* $\bar{X} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(r)}\}$*. For all* $i = 1, \dots, N$*, let*

$$\tilde{\epsilon_i} = \sup_{r=1,\dots,\bar{N}} \sum_{e \in \mathcal{C}_i} \left(\frac{(\rho_{ie}^0(\boldsymbol{x}^{(r)}))^2}{M_i} + \rho_{ie}^0(\boldsymbol{x}^{(r)})\right). \tag{15}$$

For $\epsilon_1, \dots, \epsilon_N$ *with* $\epsilon = \sum_{i=1}^N M_i \epsilon_i$*, choose* $K_1, \dots, K_N$ *such that* (9) *holds. For each* $r = 1, \dots, \bar{N}$*,* $i = 1, \dots, r$ *select*

$$\eta_i(x_i^{(r)}) \in \arg\min_{\eta_i \in \mathbb{N}} \left\{\theta^{\eta_i} M_i \tilde{\epsilon_i} + \frac{\theta - \theta^{\eta_i + 1}}{1-\theta} P_i \leq M_i \epsilon_i\right\}, \tag{16}$$

where θ *is a function of* $\boldsymbol{x}$*. Let* $\eta = \max_{\boldsymbol{x}^{(r)} \in \bar{X}} \{\eta_1, \dots, \eta_N\}$*. Fix confidence parameters* $\beta_1, \dots, \beta_N \in (0, 1)$ *with* $\sum_{i=1}^N \beta_i = 1$*. Then we have:*

$$\begin{aligned} &\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\{\boldsymbol{x} \in \mathcal{B} : \rho_{\text{sum}}^\eta(\boldsymbol{x}) > \epsilon\} \geq 1 - \sum_{i=1}^N \sqrt[\bar{N}-1]{1-\beta_i}\right\} \\ &\geq 1 - \sum_{i=1}^N \beta_i. \end{aligned} \tag{17}$$

Scenarios $\boldsymbol{x}^{(r)}, r = 1, \dots, \bar{N}$ can be sampled in a distributed manner; this is shown in Algorithm 3 in the next section. Another property of interest is whether the control input $\boldsymbol{u}^\eta(\boldsymbol{x})$ is Lipschitz continuous in $\boldsymbol{x}$. The continuity of the control input is important both from view of theory and application. A Lipschitz continuous control input $\boldsymbol{u}^\eta(\boldsymbol{x})$ together with Lipschitz continuous $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ in (1) guarantees the Lipschitz continuity of the whole vector field $f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})$. From Peano's Uniqueness Theorem, the solution of $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})$ is then unique locally. In applications, continuous control input avoids system chattering, and guarantees steady motion performance for kinetic systems.

Proposition 2. *For system* (1)*, fix* η *and let* $\boldsymbol{u}^\eta(\boldsymbol{x})$ *be the truncated control input solved by Algorithm 2 for the CBF-QP problem* (3)*. Then* $\boldsymbol{u}^\eta(\boldsymbol{x})$ *is locally Lipschitz continuous in* $\boldsymbol{x}$*.*

Proof. We will show this by means of induction. For agent i, consider the first iteration with $k = 1$ in Algorithm 2. From [47, Theorem 2] we have that u_i^1 and μ_{ie}^1 are functions in the form of

$$u_i^1 = u_i(\boldsymbol{x}, \{\lambda_{il}^0, \lambda_{li}^0\}_{l \in \mathcal{N}_i \cap \mathcal{V}_e, e \in \mathcal{C}_i}), \tag{18a}$$

$$\mu_{ie}^1 = \mu_{ie}(\boldsymbol{x}, \{\lambda_{il}^0, \lambda_{li}^0\}_{l \in \mathcal{N}_i \cap \mathcal{V}_e}), \forall e \in \mathcal{C}_i, \tag{18b}$$

where $\{\lambda_{il}^0, \lambda_{li}^0\}_{l \in \mathcal{N}_i \cap \mathcal{V}_e, e \in \mathcal{C}_i}$ represents all the auxiliary variables in the sub-problem (5). In the sequel, we will use $\boldsymbol{\lambda}$ in stead of $\{\lambda_{il}, \lambda_{li}\}$ for simplicity. By regarding the state $\boldsymbol{x}$ and the auxiliary variable $\boldsymbol{\lambda}$ as parameters, we obtain that $u_i(\boldsymbol{x}, \boldsymbol{\lambda})$, and $\mu_{ie}(\boldsymbol{x}, \boldsymbol{\lambda})$ are both piece-wise locally Lipschitz continuous functions over $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ [47, Section III A]. Following this

result, we have the expression for λ_{il}^1 from the update (6)

$$\begin{aligned}\lambda_{il}^1 &= \lambda_{il}^0 - \gamma(\mu_{ie}^1 - \mu_{le}^1),\\ &= \lambda_{il}(\boldsymbol{x}, \boldsymbol{\lambda}^0).\end{aligned} \tag{19}$$

Clearly, $\lambda_{il}(\boldsymbol{x}, \boldsymbol{\lambda}^0)$ is Lipschitz continuous in the first argument $\boldsymbol{x}$, and the second argument $\boldsymbol{\lambda}^0$. Recursively repeating the explicit update procedure (18) and (19) for the auxiliary variables until $k \in \mathbb{N} < \eta$, we obtain that

$$u_i^{k+1} = u_i(\boldsymbol{x}, \boldsymbol{\lambda}^k), \mu_{ie}^{k+1} = \mu_{ie}(\boldsymbol{x}, \boldsymbol{\lambda}^k), \lambda_{il}^{k+1} = \lambda_{il}(\boldsymbol{x}, \boldsymbol{\lambda}^k). \tag{20}$$

For the induction hypothesis assume that $\boldsymbol{\lambda}^k$ is a Lipschitz continuous in $\boldsymbol{x}$. By (20) then $\boldsymbol{\lambda}^{k+1}$ is also a Lipschitz continuous function in $\boldsymbol{x}$. Together with $\boldsymbol{\lambda}^1$ being Lipschitz continuous, we conclude that $\boldsymbol{\lambda}^{\eta-1}$ is Lipschitz continuous in $\boldsymbol{x}$, and hence also $\boldsymbol{u}^\eta$. □

III. Distributed Safety Verification

In this section we show how to verify safety by checking the *risk* of becoming unsafe along the current trajectories by means of the CBFs using the so called scenario approach. The verification method proposed in this section can be used with any type of time-invariant control inputs; CBF is only regarded as a verification criteria but not necessarily as a control design principle.

A. Scenario Based Safety Verification

Consider an N-agent system (1) 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 any $\boldsymbol{x}(0) \in \mathcal{B}$ and $t > 0$.

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

$$\begin{aligned}\min_{\boldsymbol{z}\le 0, \boldsymbol{\zeta}\ge 0} \quad & \sum_{i=1}^{N}\sum_{e\in\mathcal{C}_i}\left(z_{ie}^2 + M\sum_{r=1}^{\bar{N}}\zeta_{ie}^{(r)}\right) \qquad \text{(SC-Verification)}\\ \text{s.t.} \quad & \sum_{i\in\mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x}^{(r)})) \le \sum_{i\in\mathcal{V}_e}(z_{ie} + \zeta_{ie}^{(r)}),\\ & \forall e = 1,\dots,E, \forall r = 1,\dots,\bar{N},\end{aligned} \tag{21}$$

where scenarios $\boldsymbol{x}^{(r)} \in \mathcal{B}$ for any $r = 1,\dots,\bar{N}$ are extracted according to some probability distribution to be clarified in the sequel. Throughout the section $\bar{X} = \{\boldsymbol{x}^{(1)},\dots,\boldsymbol{x}^{(\bar{N})}\}$ 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 $r = 1,\dots,\bar{N}$. 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 $\boldsymbol{x} \in \mathcal{B}$ 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 $\mathcal{Z}(\mathcal{B})$ for optimal solution $\boldsymbol{z}^*$ is defined as follow

$$\boldsymbol{z}^* \in \mathcal{Z}(\mathcal{B}) \Longleftrightarrow$$

$$\sum_{i\in\mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x})) \le \sum_{i\in\mathcal{V}_e} z_{ie}^*, \forall e = 1,\dots,E, \forall \boldsymbol{x} \in \mathcal{B}. \tag{22}$$

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

$$\mathcal{Z}(\mathcal{B}) = \bigcap_{i=1}^{N} \mathcal{Z}_i(\mathcal{B}). \tag{23}$$

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

B. Sampling the Scenarios

The scenarios are sampled independently from the uncertain set $\mathcal{B}$. For sampling we define a probability density $\pi(\boldsymbol{x})$ associated with set $\mathcal{B}$ that satisfies:

$$\int_{\mathcal{B}} \pi(\boldsymbol{x}) d\boldsymbol{x} = 1.$$

One typical choice of $\pi(\boldsymbol{x})$ is to set it according to the density of the uniform distribution, i.e., $\pi(x) = \pi^{\text{uni}}(\boldsymbol{x}) = \frac{1}{\int_{\mathcal{B}} d\boldsymbol{x}}$.

Assumption 2. *The safe invariant set $\mathcal{B}$ constructed from* (2a) *has non-zero Lebesgue measure.*

Assumption 2 is not a strong assumption since there always exists non-empty $\mathcal{B}$ as $\mathcal{S}$ is assumed to be non-empty. We only omit the case where $\mathcal{B}$ is a single-point set, or a countable number of points set. This guarantees the existence of $\pi^{\text{uni}}(\boldsymbol{x})$. Then, $\boldsymbol{x}$ can be sampled for $\bar{N}$ times independently from the distribution $\pi^{\text{uni}}(\boldsymbol{x})$. We note here that different probability distributions have minor impact on the final violation results, interested readers are referred to [45, Section 3.1]. Although the uniform distribution here is well-defined, the uncertain set $\mathcal{B}$ is defined implicitly as an intersection of multiple sets (2a). Sampling a point from the proposed uniform distribution is rather arduous in practice, and agents may not have access to $\mathcal{B}$. Here, we provide a sequential algorithm to sample scenarios $\boldsymbol{x}^{(r)}$, $r = 1,\dots,\bar{N}$.

Algorithm 3 Scenarios Sampling Algorithm

Initialization Uncertain set $\mathcal{B}$ constructed by (2a), failed times $F = 0$.

Output: Scenario $\boldsymbol{x}^{(r)}$.

1: Sample $x_1^{(r)}$ from $\pi_1(x_1)$.
2: **for** $i = 2,\dots,N$ **do**
3: Construct a distribution $\pi_i(\{x_1^{(r)},\dots,x_{i-1}^{(r)}\}; x_i)$ following (24) (25).
4: **if** $\pi_i = 0$ **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 $\pi_i(\{x_1^{(r)},\dots,x_{i-1}^{(r)}\}; x_i)$.
9: **end for**

The algorithm constructs the densities from which samples are extracted sequentially for each agent. We first show the construction of $\pi_i(\{x_1^{(r)},\dots,x_{i-1}^{(r)}\}; x_i)$. Here $\{x_1^{(r)},\dots,x_{i-1}^{(r)}\}$ are sampled according to the i-th density, π_i. To determine this, we first define the uncertain 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.

$$\tilde{\mathcal{B}}_{ie} = \begin{cases} \mathcal{X}_i, \text{ if } \exists l \in \mathcal{V}_e, \text{such that } l > i \\ \{x_i \in \mathbb{R}^{n_i} | b_{ie}(x_i, \{x_l^{(r)}\}) \geq 0)\}, \text{ otherwise} \end{cases} \tag{24}$$

We then have that $\tilde{\mathcal{B}}_i = \bigcap_{e \in \mathcal{C}_i} \tilde{\mathcal{B}}_{ie}$. The parameters in (24) can all be collected by local communication, since only states of agents in the same sub-network are required. Note here $\tilde{\mathcal{B}}_i$ is possibly empty with some parameters $\{x_1^{(r)}, \ldots, x_{i-1}^{(r)}\}$. The distribution $\pi_i(\{x_1^{(r)}, \ldots, x_{i-1}^{(r)}\}; x_i)$ used in the step 3 of Algorithm 3 is a uniform distribution over $\tilde{\mathcal{B}}_i$, i.e.,

$$\pi_i(\{x_1^{(r)}, \ldots, x_{i-1}^{(r)}\}; x_i) = \begin{cases} \frac{1}{\int_{\tilde{\mathcal{B}}_i} dx_i}, \text{ if } \tilde{\mathcal{B}}_i \neq \emptyset \\ 0, \text{ otherwise} \end{cases} \tag{25}$$

In step 1, the first scenario $x_1^{(r)}$ associated with agent 1 is sampled from distribution $\pi_1(x_1) = \frac{1}{\int_{\mathcal{X}_1} dx}$, since now there are no other agents involved to restrict the uncertain set for agent 1. Then, the sampling-construction procedures repeat sequentially from agent 2 to agent N. One case needs additional attention, where $\pi_i = 0$. This implies that $\tilde{\mathcal{B}}_i = \emptyset$. By Assumption 2, there exists $\{x_1^{(r)}, \ldots, x_{i-1}^{(r)}\}$ such that $\tilde{\mathcal{B}}_i \neq \emptyset$. Therefore, if $\pi_i = 0$ (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 uncertain set $\tilde{\mathcal{B}}_{i-1}$ and distribution π_{i-1} is such that for any $x_{i-1}^{(r)} \in \tilde{\mathcal{B}}_{i-1}$, $\tilde{\mathcal{B}}_i = \emptyset$. It is guaranteed that $F \leq i - 1$ for $i \geq 2$, since $\tilde{\mathcal{B}}_1 = \mathcal{X}_1 \neq \emptyset$.

Proposition 3. *The scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, \bar{N}$, are feasible, i.e. $\boldsymbol{x}^{(r)} \in \mathcal{B}$, and independent.*

Proof. The feasibility result holds directly from the definition of every uncertain set $\tilde{\mathcal{B}}_i$ in (24) 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, $\boldsymbol{x}^{(r)} \in \mathcal{B}$. $\boldsymbol{x}^{(r)}$ 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 3 since we seek independence across r.

C. 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

$$\begin{aligned} J_i(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) &= \sum_{e \in \mathcal{C}_i} \left(z_{ie}^2 + \sum_{r=1}^{\bar{N}} \zeta_{ie}^{(r)} \right), \\ \hat{h}_{ie}^{(r)}(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) &= h_{ie}(u_i(\boldsymbol{x}^{(r)})) - z_{ie} - \zeta_{ie}^{(r)}, r = 1, \ldots, \bar{N}, \end{aligned} \tag{26}$$

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 4. *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*

$$\begin{aligned} &\binom{\bar{N}}{s_i^*} t_i^{\bar{N} - s_i^*} - \frac{\beta_i}{2\bar{N}} \sum_{j=s_i^*}^{\bar{N}-1} \binom{j}{s_i^*} t_i^{j - s_i^*} \\ &- \frac{\beta_i}{6\bar{N}} \sum_{j=\bar{N}+1}^{4\bar{N}} \binom{j}{s_i^*} t_i^{j - s_i^*} = 0, \end{aligned} \tag{27}$$

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

$$1 - \frac{\beta}{6N} \sum_{j=\bar{N}+1}^{4\bar{N}} \binom{j}{s_i^*} t_i^{j - \bar{N}} = 0. \tag{28}$$

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 $\underline{t}_i(s_i^*) \leq \bar{t}(s_i^*)$. 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

$$\mathbb{P}^{\bar{N}} \left\{ \frac{\underline{\epsilon}(s^*)}{N} \leq \mathbb{P}\{\boldsymbol{x} \in \mathcal{B} : 0 \notin \mathcal{Z}\} \leq \bar{\epsilon}(s^*) \right\} \geq 1 - \beta, \tag{29}$$

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

Note that Theorem 4 constitutes a generalization of [45, Theorem 2] to a multi-agent setting. It also extends [29] by determining the lower bound $\frac{\underline{\epsilon}(s^*)}{N}$. Theorem 4 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 4.1. *Consider the multi-agent system* (1)*, and let $\underline{\epsilon}_i(s_i^*)$, $\bar{\epsilon}_i(s_i^*)$, and β_i as in Theorem 4. We then have that*

$$\begin{aligned} &\mathbb{P}^{\bar{N}} \left\{ \underline{\epsilon}_i(s_i^*) \leq \mathbb{P}\left\{ \boldsymbol{x} \in \mathcal{B} : \sum_{i \in \mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x}^{(r)})) > 0 \right\} \leq \bar{\epsilon}_i(s_i^*) \right\} \\ &\geq 1 - \beta_i. \end{aligned} \tag{30}$$

IV. Simulations 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 [21].

A. 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 $i = 1, \dots, 10$ are considered, with double integrator dynamics

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

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 working space is restricted to be $||\boldsymbol{p}_i||_\infty \leq x^{\max}$, $||\boldsymbol{v}_i||_\infty \leq v_i^{\max}$, $||\boldsymbol{a}_i||_\infty \leq a_i^{\max}$. 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

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

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. 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 $i = 1, \dots, 5$, $\boldsymbol{a}_i^{\max} = 1$, while for $i = 6, \dots, 10$, $\boldsymbol{a}_i^{\max} = 10$. 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 (33).

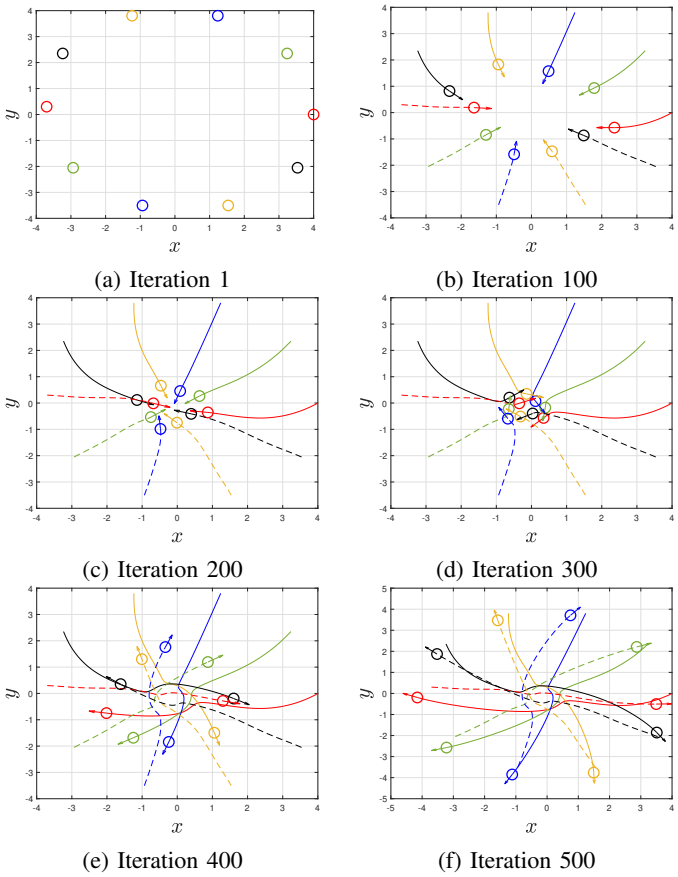
B. Distributed Control: Asymptotic Algorithm

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.

C. Distributed Control: Truncated Algorithm

The truncated Algorithm 2 is then implemented for the same setting. Here we consider the problem in a working space with boundary $x^{\max} = 4$, $v^{\max} = 3$. To employ Theorem 3 we used 50 samples related to the truncation parameter, and set $\beta_1, \dots, \beta_{10} = 0.01$, $\epsilon_1, \dots, \epsilon_{10} = 0.001$, obtaining

$$\mathbb{P}^{50}\left\{\mathbb{P}\left\{\boldsymbol{x} \in \mathcal{B} : \rho_{\text{sum}}^{79}(\boldsymbol{x}) > 10^{-3}\right\} \leq 0.088\right\} \geq 0.9.$$



(a) Iteration 1 (b) Iteration 100 (c) Iteration 200 (d) Iteration 300 (e) Iteration 400 (f) Iteration 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.

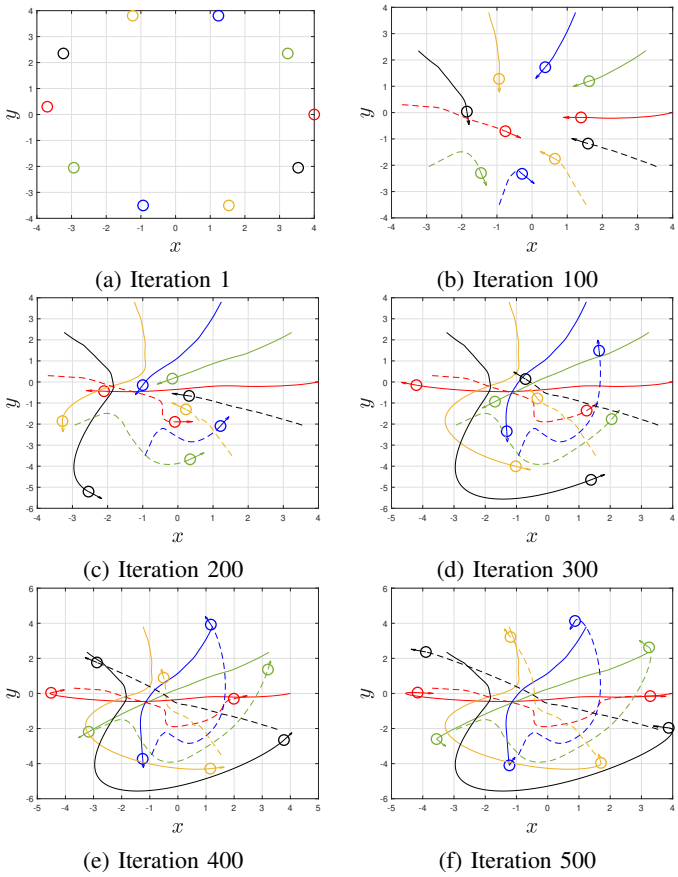
For a practical implementation η could be much smaller, as its theoretical value is calculated by considering the most ill-conditioned matrix (36) for $r = 1, \dots, \bar{N}$. To this end, we numerically investigated the performance of the algorithm with $\eta = 30$; the resulting swapping trajectories are shown in Figure 2. The evolution of the relaxation parameters $\rho_{\text{sum}}^0(\boldsymbol{x})$ and $\rho_{\text{sum}}^\eta(\boldsymbol{x})$ across algorithm iterations, is shown in Figures 3a and 3b.

D. Distributed Safety Verification

Safety verification is performed for a four-robot system, within working space $x^{\max} = 6$, $v^{\max} = 1$. Each robot is using Algorithm 2 to safely move towards the origin. We sample 200 scenarios via Algorithm 3. Theorem 4 yields then that with confidence at least 0.9, $\mathbb{P}\{\boldsymbol{x} \in \mathcal{B} : 0 \notin \mathcal{Z}\} \in [0, 0.146]$. We repeat this procedure 300 times, each time using 300 scenarios, and construct the empirical cumulative distribution function of $\mathbb{P}\{\boldsymbol{x} \in \mathcal{B} : 0 \notin \mathcal{Z}\}$. This is shown in Figure 4; it can be observed that the empirical probability that $\mathbb{P}\{\boldsymbol{x} \in \mathcal{B} : 0 \notin \mathcal{Z}\} \in [0, 0.146] \approx 1$, thus satisfying the theoretical confidence lower bound of 0.9.

V. Conclusion

In this paper we presented distributed safe control design and safety verification algorithms for multi-agent systems. The

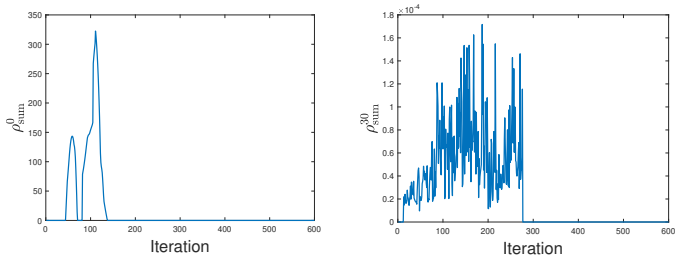


(a) Iteration 1 (b) Iteration 100

(c) Iteration 200 (d) Iteration 300

(e) Iteration 400 (f) Iteration 500

Fig. 2: 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) $\rho^\eta_{\text{sum}}(\boldsymbol{x})$ along the trajectory

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

proposed control algorithms introduce auxiliary and relaxation variables to allow feasibility across iterations. We guaranteed convergence to an optimal solution and establish a linear convergence rate. Furthermore, the relaxation variables are guaranteed to decrease until below a certain level by setting the penalty coefficients locally. This level can be determined by each agent, and can be reached in a finite number of iterations; the latter is determined in a probabilistic way. 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

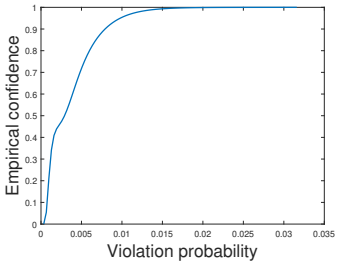


Fig. 4: Cumulative distribution function for safety violation.

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.

APPENDIX

Proof of Lemma 1. Combining the optimisation problems (5) in Step 3 of Algorithm 1 for all agents, we have

$$\begin{aligned} &\min_{\boldsymbol{u},\boldsymbol{\rho}} \; J(\boldsymbol{u},\boldsymbol{\rho}) \\ &\text{subject to } u_i \in \mathcal{U}_i, \forall i = 1,\dots,N, \boldsymbol{\rho} \geq 0 \\ &\qquad \sum_{k\in\mathcal{V}_e} h_{ke}(u_k) \leq \sum_{k\in\mathcal{V}_e} \rho_{ke}, \forall e = 1,\dots,E, \end{aligned} \tag{34}$$

where for each $e = 1,\dots,E$, μ_e is the dual variable associated with the last constraint in (34). The dual function of (34) is given by

$$\begin{aligned} q_R(\boldsymbol{\mu}) &= \inf_{\{u_i\in\mathcal{U}_i\},\boldsymbol{\rho}\geq 0} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e\in\mathcal{C}_i} (\rho_{ie}^2 + M_i\rho_{ie}) \right\} \\ &\quad + \sum_{e=1}^{E} \mu_e \left\{ \sum_{k\in\mathcal{V}_e} h_{ke}(u_k) - \sum_{k\in\mathcal{V}_e} \rho_{ke} \right\} \\ &= \inf_{\{u_i\in\mathcal{U}_i\},\boldsymbol{\rho}\geq 0} \sum_{i=1}^{N} \left\{ J_i(u_i) + \sum_{e\in\mathcal{C}_i} \mu_e h_{ke}(u_k) \right\} \\ &\quad + \sum_{e=1}^{E} \sum_{k\in\mathcal{V}_e} \left\{ \rho_{ke}^2 + (M_k - \mu_e)\rho_{ke} \right\} \\ &= \begin{cases} -\infty, & \text{if } M_k - \mu_e < 0, \forall e = 1,\dots,E, k \in \mathcal{V}_e \\ \inf\limits_{\{u_i\in\mathcal{U}_i\}} \sum\limits_{i=1}^{N} \left\{ J_i(u_i) + \sum\limits_{e\in\mathcal{G}_i} \mu_e h_{ke}(u_k) \right\}, & \text{else.} \end{cases} \end{aligned}$$

From the dual function we obtain that if $M_k - \mu_e \geq 0$ for any $e = 1,\dots,E$ and $k \in \mathcal{V}_e$, then the minimization of the relaxed dual function is equivalent to the minimization of the dual function of problem (3) that does not include any relaxation. Besides, the optimisation problems (34) and (3) both have zero duality gap since they are convex QPs. We conclude that solving the relaxed problem (34) yields the same minimizer as the original unrelaxed problem (3), i.e. $\boldsymbol{u}^*_{\text{dis}} = \boldsymbol{u}^*_{\text{cen}}$. The cost function $J(\boldsymbol{u},\boldsymbol{\rho})$ is strongly convex with Lipschitz continuous gradient since it is a strict convex quadratic function over both $\boldsymbol{u}$ and $\boldsymbol{\rho}$. □

Proof of Theorem 2. Following the convergence analysis in [31, Section III], (5) and (6) are doing gradient ascent over the dual problem

$$\begin{aligned} &\max\ q_R(\boldsymbol{\mu}) \\ &\text{subject to } \mu_{ie} = \mu_{ke}, \forall i, k \in \mathcal{V}_e, e = 1, \ldots, E. \end{aligned} \tag{35}$$

Following the example [48, Section 5.2.4, Eq. 5.28], the dual function $q_R(\boldsymbol{\mu})$ is a quadratic function with the quadratic part as

$$\boldsymbol{\mu}^\top \begin{bmatrix} {L_{g_1} b_{11}}^\top \\ \vdots \\ {L_{g_N} b_{NE}}^\top \end{bmatrix} \begin{bmatrix} {L_{g_1} b_{11}}^\top \\ \vdots \\ {L_{g_N} b_{NE}}^\top \end{bmatrix}^\top \boldsymbol{\mu}, \tag{36}$$

Therefore, $q_R(\boldsymbol{\mu})$ is a $2\sigma_{\min}(\boldsymbol{x})$-strongly convex, and $2\sigma_{\max}(\boldsymbol{x})$-smooth function, where $\sigma_{\min}(\boldsymbol{x})$ and $\sigma_{\max}(\boldsymbol{x})$ are the minimal and maximal eigenvalues for the square matrix in (36). By selecting stepsize $0 \leq \gamma \leq 1/(\sigma_{\min}(\boldsymbol{x}) + \sigma_{\max}(\boldsymbol{x}))$, and leveraging dual ascent steps for (35), the $q_R(\boldsymbol{\mu}^k)$ converges to $q_R(\boldsymbol{\mu}^*)$ with linear convergence rate $\theta = 1 - \frac{\sigma_{\min}(\boldsymbol{x})}{\sigma_{\max}(\boldsymbol{x})}$ [49]. From the argument in [30, Lemma 5.1],for any $k > 0$, $q_R(\boldsymbol{\mu}^k) = \left(\sum_{i=1}^N ||u_i^k - u^{\text{des}}||^2 + \rho_{\text{sum}}^k\right)$, we obtain

$$\begin{aligned} &\sum_{i=1}^N ||u_i^{k+1} - u^{\text{des}}||^2 + \rho_{\text{sum}}^{k+1} \\ &\leq \theta \left(\sum_{i=1}^N ||u_i^k - u^{\text{des}}||^2 + \rho_{\text{sum}}^k \right). \end{aligned} \tag{37}$$

Using the fact that $0 \leq ||u^{k+1} - u^{\text{des}}||^2 \leq P_i$, hence it can be dropped from (37), and $0 \leq ||u^k - u^{\text{des}}||^2 \leq P_i$, and combining this with (37) we obtain

$$\begin{aligned} &\rho_{\text{sum}}^{k+1} \leq \theta \left(\sum_{i=1}^N P_i + \rho_{\text{sum}}^k \right) \\ \Rightarrow &\rho_{\text{sum}}^k - (\rho_{\text{sum}}^k - \rho_{\text{sum}}^{k+1}) \leq \theta \left(\sum_{i=1}^N P_i + \rho_{\text{sum}}^k \right) \\ \Rightarrow &\rho_{\text{sum}}^k - \rho_{\text{sum}}^{k+1} \geq (1-\theta)\rho_{\text{sum}}^k - \theta \sum_{i=1}^N P_i. \end{aligned} \tag{38}$$

Combining (38) with (9), we have

$$\begin{aligned} &\rho_{\text{sum}}^k - \rho_{\text{sum}}^{k+1} > (1-\theta) \sum_{i=1}^N M_i \epsilon_i - \theta \sum_i^N \frac{1-\theta}{\theta} M_i \epsilon_i \\ \Rightarrow &\rho_{\text{sum}}^k - \rho_{\text{sum}}^{k+1} > 0, \end{aligned} \tag{39}$$

where the inequality is strict due to the strict inequalities in (9). Hence, we conclude the proof. □

Proof of Corollary 2.1. Equation (11) in *Case I*. The proof is conducted by considering two scenarios. For the ease of illustration, let $\epsilon = \sum_{i=1}^N M_i \epsilon_i$.

Firstly, consider the scenario where $\rho_{\text{sum}}^0 > \ldots > \rho_{\text{sum}}^{k-1} > \rho_{\text{sum}}^k > \epsilon$, and use the inequality $\rho_{\text{sum}}^{k+1} \leq \theta \left(\sum_{i=1}^N P_i + \rho_{\text{sum}}^k \right)$ from (38). A proportional sequence $\{a^0, \ldots, a^{k+1}\}$ is defined by $a^0 = \rho_{\text{sum}}^0$, and $a^{k+1} = \theta \left(\sum_{i=1}^N P_i + a^k \right)$. By comparing the sequences $\{a^0, \ldots, a^{k+1}\}$ and $\{\rho_{\text{sum}}^0, \ldots, \rho_{\text{sum}}^{k+1}\}$, we have $\rho_{\text{sum}}^i \leq a^i$, for every $i = 0, \ldots, k+1$. Clearly, we also have $a^{k+1} = \theta^k \rho_{\text{sum}}^0 + \frac{\theta - \theta^{k+1}}{1-\theta} \sum_{i=1}^N P_i$, which shows that $\rho_{\text{sum}}^{k+1} \leq \theta^k \rho_{\text{sum}}^0 + \frac{\theta - \theta^{k+1}}{1-\theta} \sum_{i=1}^N P_i$.

Secondly, we consider the scenario where $\rho_{\text{sum}}^k < \epsilon$. Suppose $\rho_{\text{sum}}^k = \epsilon - \delta$, where $\delta \in (0, \epsilon)$. From (38) we have $\rho_{\text{sum}}^k - \rho_{\text{sum}}^{k+1} \geq (1-\theta)\rho_{\text{sum}}^k - \theta \sum_{i=1}^N P_i$. Then

$$\begin{aligned} \rho_{\text{sum}}^{k+1} &\leq \theta \left(\rho_{\text{sum}}^k + \sum_{i=1}^N P_i \right) = \theta \left(\epsilon - \delta + \sum_{i=1}^N P_i \right) \\ &\leq \theta \left(\epsilon - \delta + \frac{1-\theta}{\theta} \epsilon \right) = \epsilon - \theta\delta < \epsilon. \end{aligned}$$

These two scenarios cover every possibility since if $\rho_{\text{sum}}^k > \epsilon$, from the arguments in the second case we conclude that $\rho_{\text{sum}}^{k-1} > \epsilon$. Directly this corresponds to the first case. Hence, the two cases are complementary, and we conclude the proof for (11). (12) is proved by $\lim_{k\to\infty} \theta^k \rho_{\text{sum}}^0 + \frac{\theta - \theta^{k+1}}{1-\theta} \sum_{i=1}^N P_i = \frac{\theta}{1-\theta} \sum_{i=1}^N P_i < \epsilon$. Equation (14) in *Case II* follows then directly by setting $k = 0$ in the second scenario above.

□

Proof of Theorem 3. We begin with prove that (15) and (16), together with other prior conditions are sufficient for

$$\rho_{\text{sum}}^\eta(\boldsymbol{x}^{(r)}) \leq \epsilon, \forall r = 1, \ldots, \bar{N}. \tag{40}$$

Substituting η_i with η in (16) yields

$$\begin{aligned} &\theta^\eta M_i \tilde{\epsilon_i} + \frac{\theta - \theta^{\eta+1}}{1-\theta} P_i \leq M_i \epsilon_i, \forall i = 1, \ldots, N \\ \Rightarrow &\theta^\eta \sum_{i=1}^N M_i \tilde{\epsilon_i} + \frac{\theta - \theta^{\eta+1}}{1-\eta} \sum_{i=1}^N P_i \leq \sum_{i=1}^N M_i \epsilon_i = \epsilon. \end{aligned}$$

According to (15), $\sum_{i=1}^N M_i \tilde{\epsilon_i} \geq \rho_{\text{sum}}^0(\boldsymbol{x}^{(r)})$, using this term to substitute the first term on the left hand side, and we obtain

$$\theta^\eta \rho_{\text{sum}}^0(\boldsymbol{x}^{(r)}) + \frac{\theta - \theta^{\eta+1}}{1-\theta} P_i \leq \epsilon, \forall r = 1, \ldots, \bar{N}.$$

Hence, from Corollary 2.1, we have (40) holds.

We next characterize the distribution of $\mathbb{P}\{\boldsymbol{x} \in \mathcal{B} : \rho_{\text{sum}}^\eta(\boldsymbol{x}) > \epsilon\}$. First, consider the random variables

$$\mathbb{P}\left\{\boldsymbol{x} \in \mathcal{B} : \delta_i\right\}, i = 1, \ldots, N,$$

where $\delta_i = \sum_{e \in \mathcal{C}_i} \left(\frac{(\rho_{ie}^\eta(\boldsymbol{x}))^2}{M_i} + \rho_{ie}^\eta(\boldsymbol{x}) \right) > \epsilon_i$ represents an event. Recall that every $\tilde{\epsilon_i}$ is calculated by solving (15), which is a *fully supported* problem [41, Definition 3]. Therefore, δ_i is a Beta variable with parameters $(1, \bar{N})$ [41, Theorem 1], the cumulative distribution is given by

$$\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\left\{\boldsymbol{x} \in \mathcal{B} : \delta_i\right\} \leq 1 - \tilde{\varepsilon_i}\right\} = \beta_i, i = 1, \ldots, N,$$

where $\tilde{\varepsilon_i} = \sqrt[\bar{N}-1]{1-\beta_i}$. Equivalently, we have

$$\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\left\{\boldsymbol{x} \in \mathcal{B} : \bar{\delta_i}\right\} \geq \tilde{\varepsilon_i}\right\} = \beta_i, i = 1, \ldots, N, \tag{41}$$

where $\bar{\delta_i}$ is the event complementary to δ_i. We first show that

$$\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\left\{\boldsymbol{x} \in \mathcal{B} : \bigcup_{i=1}^N \bar{\delta_i}\right\} \leq \sum_{i=1}^N \tilde{\varepsilon_i}\right\} \geq 1 - \sum_{i=1}^N \beta_i. \tag{42}$$

We have that

$$\begin{aligned}
&\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bigcup_{i=1}^{N}\bar{\delta}_i\right\}\leq\sum_{i=1}^{N}\tilde{\varepsilon}_i\right\}\\
=&\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\left\{\bigcup_{i=1}^{N}\left\{\boldsymbol{x}\in\mathcal{B}:\bar{\delta}_i\right\}\right\}\leq\sum_{i=1}^{N}\tilde{\varepsilon}_i\right\}\\
\geq&\mathbb{P}^{\bar{N}}\left\{\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bar{\delta}_i\right\}\leq\sum_{i=1}^{N}\tilde{\varepsilon}_i\right\}\\
\geq&\mathbb{P}^{\bar{N}}\left\{\bigcap_{i=1}^{N}\left\{\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bar{\delta}_i\right\}\leq\tilde{\varepsilon}_i\right\}\right\}\\
\geq&1-\sum_{i=1}^{N}\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bar{\delta}_i\right\}>\tilde{\varepsilon}_i\right\}\geq 1-\sum_{i=1}^{N}\beta_i,
\end{aligned}$$

where the first inequality is due to subbatitivity of $\mathbb{P}$. The second inequality is since an element in the intersection $\bigcap_{i=1}^{N}\left\{\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bar{\delta}_i\right\}\leq\tilde{\varepsilon}_i\right\}$ will also be in the set of samples such that $\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bar{\delta}_i\right\}\leq\tilde{\varepsilon}_i$ for all $i=1,\ldots,N$, and hence also in the set of samples such that $\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bar{\delta}_i\right\}\leq\sum_{i=1}^{N}\tilde{\varepsilon}_i$. The third inequality follows from the second one using the set complement and the subbatitivity of $\mathbb{P}^{\bar{N}}$, while the last inequality is due to (41). Using (42), we have

$$\begin{aligned}
&\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\{\boldsymbol{x}\in\mathcal{B}:\rho^{\eta}_{\mathrm{sum}}(\boldsymbol{x})>\epsilon\}\geq 1-\sum_{i=1}^{N}\tilde{\varepsilon}_i\right\}\\
\geq&\mathbb{P}^{\bar{N}}\left\{\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bigcap_{i=1}^{N}\delta_i\right\}\geq 1-\sum_{i=1}^{N}\tilde{\varepsilon}_i\right\}\\
=&\mathbb{P}^{\bar{N}}\left\{1-\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bigcup_{i=1}^{N}\bar{\delta}_i\right\}\geq 1-\sum_{i=1}^{N}\tilde{\varepsilon}_i\right\}\\
\geq&1-\sum_{i=1}^{N}\beta_i,
\end{aligned}$$

thus concluding the proof. □

Proof of Theorem 4. We have that

$$\begin{aligned}
&\mathbb{P}^{\bar{N}}\left\{\frac{\sum_{i=1}^{N}\underline{\epsilon}_i(s_i^*)}{N}\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}\right\}\leq\sum_{i=1}^{N}\bar{\epsilon}_i(s_i^*)\right\}\\
&=\mathbb{P}^{\bar{N}}\left\{\frac{1}{N}\sum_{i=1}^{N}\underline{\epsilon}_i(s_i^*)\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\right.\right.\\
&\left.\left.\exists i\in\{1,\ldots,N\},\ \boldsymbol{z}^*\notin\mathcal{Z}_i\right\}\leq\sum_{i=1}^{N}\bar{\epsilon}_i(s_i^*)\right\}\\
&=\mathbb{P}^{\bar{N}}\left\{\frac{1}{N}\sum_{i=1}^{N}\underline{\epsilon}_i(s_i^*)\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\bigcup_{i=1}^{N}\{\boldsymbol{z}^*\notin\mathcal{Z}_i\}\right\}\right.\\
&\left.\bigcap\mathbb{P}\left\{\bigcup_{i=1}^{N}\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\}\right\}\leq\sum_{i=1}^{N}\bar{\epsilon}_i(s_i^*)\right\}\qquad(43)
\end{aligned}$$

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

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

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

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

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 (43) can be then lower-bounded by

$$\begin{aligned}
&\mathbb{P}^{\bar{N}}\left\{N\cdot\frac{1}{N}\sum_{i=1}^{N}\underline{\epsilon}_i(s_i^*)\leq\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}\right.\\
&\left.\bigcap\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}\leq\sum_{i=1}^{N}\bar{\epsilon}_i(s_i^*)\right\}\\
&\geq\mathbb{P}^{\bar{N}}\left\{\bigcap_{i=1}^{N}\left\{\underline{\epsilon}_i(s_i^*)\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}\leq\bar{\epsilon}_i(s_i^*)\right\}\right\}\\
&\geq 1-\sum_{i=1}^{N}\mathbb{P}^{\bar{N}}\left\{\bar{\epsilon}_i(s_i^*)<\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}\right.\\
&\left.\bigcup\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}<\underline{\epsilon}_i(s_i^*)\right\}.\qquad(44)
\end{aligned}$$

By [45, Theorem 1] we have that for any $i=1,\ldots,N$

$$\begin{aligned}
&\mathbb{P}^{\bar{N}}\left\{\boldsymbol{x}\in\mathcal{B}:\underline{\epsilon}_i(s_i^*)\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}\leq\bar{\epsilon}_i(s_i^*)\right\}\\
&\geq 1-\beta_i\\
\Rightarrow&\sum_{i=1}^{N}\mathbb{P}^{\bar{N}}\left\{\bar{\epsilon}_i(s_i^*)<\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}\right.\\
&\left.\bigcup\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}_i\right\}<\underline{\epsilon}_i(s_i^*)\right\}<\sum_{i=1}^{N}\beta_i.\qquad(45)
\end{aligned}$$

Since $\frac{\underline{\epsilon}(s^*)}{N}<\underline{\epsilon}(s^*)<\bar{\epsilon}(s^*)$, substituting (45) into (43) with $i=1,\ldots,N$ we obtain

$$\mathbb{P}^{\bar{N}}\left\{\frac{\underline{\epsilon}(s^*)}{N}\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{B}:\boldsymbol{z}^*\notin\mathcal{Z}\right\}\leq\bar{\epsilon}(s^*)\right\}\geq 1-\beta.\quad(46)$$

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 (46) is equivalent to (29). 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," *submitted to the 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] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th conference on decision and control (CDC)*, pp. 474–479, IEEE, 2019.

[13] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, pp. 322–328, IEEE, 2016.

[14] X. Tan, W. S. Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety, and performance-critical control," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3021–3028, 2021.

[15] W. Xiao, C. Belta, and C. G. Cassandras, "Adaptive control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2267–2281, 2021.

[16] J. Zeng, B. Zhang, Z. Li, and K. Sreenath, "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility," in *2021 American Control Conference (ACC)*, pp. 3856–3863, IEEE, 2021.

[17] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation.," in *Robotics: Science and Systems*, vol. 13, Cambridge, MA, USA, 2017.

[18] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *IEEE Transactions on Automatic Control*, vol. 67, no. 3, pp. 1073–1088, 2021.

[19] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

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

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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.

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

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

[41] 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.

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

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

[44] 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.

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

[46] H. Wang, K. Margellos, and A. Papachristodoulou, "Safety verification and controller synthesis for systems with input constraints," *arXiv preprint arXiv:2204.09386*, 2022.

[47] H. Wang, K. Margellos, and A. Papachristodoulou, "Explicit solutions for safety problems using control barrier functions," *arXiv preprint arXiv:2204.09380*, 2022.

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

[49] D. G. Luenberger, Y. Ye, *et al.*, *Linear and nonlinear programming*, vol. 2. Springer, 1984.