Compatibility checking of multiple control barrier functions for input constrained systems

Xiao Tan, and Dimos V. Dimarogonas

Abstract—State and input constraints are ubiquitous in control system design. One recently developed tool to deal with these constraints is control barrier functions (CBF) which transform state constraints into conditions in the input space. CBF-based controller design thus incorporates both the CBF conditions and input constraints in a quadratic program. However, the CBF-based controller is well-defined only if the CBF conditions are compatible. In the case of perturbed systems, robust compatibility is of relevance. In this work, we propose an algorithmic solution to verify or falsify the (robust) compatibility of given CBFs \textit{a priori}. Leveraging the Lipschitz properties of the CBF conditions, a grid sampling and refinement method with theoretical analysis and guarantees is proposed.

I. INTRODUCTION

Control design for dynamical systems with input and state constraints is an omnipresent problem in engineering and has been extensively investigated over the last few decades. Among all the investigated control methods, such as model predictive control (MPC) [1], reference governor [2], barrier Lyapunov functions (BLF) [3], and prescribed performance control (PPC) [4], the so-called control barrier functions (CBF) has revived recently [5], [6] and gained increasing popularity in the control and robotic community. The latter three methods relate a system state and possibly a time constraint on the system input, which is referred to as the CBF condition. A moderate magnitude control signal is obtained when the system state is close to the boundary of the safety set, making it applicable even in the presence of noise. Compared to MPC schemes, the CBF formulation only requires to solve a small size quadratic program online and thus is more suitable for embedded systems thanks to today’s increasing computational power.

Another nice property of the CBF formulation is its modular design feature. In the case that multiple state constraints are present for the system, i.e., the safe region is the intersection of all these state constraints, we can add multiple CBF conditions to the QP formulation each of which corresponds to one state constraint. Under the core assumption that the CBF-induced QP is always feasible, or equivalently, the CBF conditions are compatible, the satisfaction of all the state constraints is guaranteed for all time.

However, the compatibility of multiple CBFs is in general difficult to check. The problem is even more challenging if input limits are also present. In [7], sufficient and necessary conditions on CBF compatibility are discussed for SISO systems without input constraints. In the recent work [8], the authors propose a mixed-initiative control formulation that satisfies the input bound explicitly and enforces CBF conditions only in a neighborhood of the safety boundary. Under an assumption that the neighborhood around the boundary of safety region specified by each CBF should not overlap, the mixed-initiative formulation is applicable in the presence of multiple CBFs. In [9], a navigation problem is considered and a CBF-based QP for the image of obstacles in the “ball world” is employed. The QP feasibility is guaranteed thanks to the special structure of the problem, yet input constraints are missing.

More relevant to our work is the sum-of-square (SoS) approach in [10] for verifying worst-case and stochastic system safety by constructing a barrier certificate. A recent work [11] extends the SoS technique to verify if a candidate function is indeed a CBF by checking whether a polynomial control input exists and satisfies the CBF condition in the safe region. In [12], a SoS-based compatibility verification scheme for multiple CBFs is proposed. However, these results are only applicable to polynomial control systems. Moreover, the feasibility result is also subject to the order of the polynomials, and a failure to find such polynomial inputs provides no falsification guarantee.

In this paper, we consider the compatibility checking problem when multiple control barrier functions are present for input constrained systems. We aim to give a verification or falsification on the compatibility of multiple CBFs prior to their online implementation. In that respect, a grid sampling and refinement method is proposed leveraging the Lipschitz properties of the CBF conditions. We show that 1) the proposed algorithm will output the exact compatibility result if it terminates, 2) the algorithm is guaranteed to terminate in finite steps if the multiple CBFs are robustly compatible, and 3) the upper bound of the robustness level can be obtained if a lower bound of the lattice size is incorporated.
II. PRELIMINARIES

Notation: The operator $\nabla : C^1(\mathbb{R}^n) \to \mathbb{R}^n$ is defined as the gradient $\frac{\partial}{\partial x}$ of a scalar-valued differentiable function with respect to $x$. The Lie derivatives of a function $h(x)$ for the system $\dot{x} = f(x) + g(x)u$ are denoted by $L_i \dot{h} = \nabla h^i f(x) \in \mathbb{R}$ and $L_i \dot{g} = \nabla h^i g(x) \in \mathbb{R}^{1 \times m}$, respectively. The interior and boundary of a set $A$ are denoted $\text{Int}(A)$ and $\partial A$, respectively. A continuous function $\alpha : [0, a) \to [0, \infty)$ for $a \in \mathbb{R}_{>0}$ is a class $K$ function if it is strictly increasing and $\alpha(0) = 0$ [13]. A continuous function $\alpha : (-b, a) \to (-\infty, \infty)$ for $a, b \in \mathbb{R}_{>0}$ is an extended class $K$ function if it is strictly increasing and $\alpha(0) = 0$. Vector inequalities are to be interpreted element-wise. $0, 1$ refer to vectors of proper dimensions with all entries to be $0$ or $1$, respectively.

Consider the nonlinear control affine system

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

(1)

where the state $x \in \mathbb{R}^n$, and the control input $u \in U \subset \mathbb{R}^m$. Assume that the vector fields $f(x)$ and $g(x)$ are locally Lipschitz functions in $x$. A set $A \subset \mathbb{R}^n$ is called forward invariant, if for any initial condition $x_0 \in A$, the system solution $x(t, x_0) \in A$ for all $t$ in the maximal time interval of existence.

Consider the safety set $C$ defined as an intersection of superlevel sets of continuously differentiable functions $h_i : \mathbb{R}^n \to \mathbb{R}$, $i \in I = \{1, 2, ..., N\}$:

$$C = \{x \in \mathbb{R}^n : h_i(x) \geq 0, i \in I\}. \tag{2}$$

**Definition 1 (Compatible CBFs).** The functions $h_i(x), i \in I$ are compatible control barrier functions (CBF) for (1) if there exists an open set $D \supseteq C$ and locally Lipschitz extended class $K$ functions $\alpha_i$ such that, $\forall x \in D$, $\forall i \in I$,

$$\exists u \in U, L_i h_i(x) + L_i g h_i(x) u + \alpha_i(h_i(x)) \geq 0. \tag{3}$$

For given differentiable functions $h_i$ and extended class $K$ functions $\alpha_i$, $i \in I$, define

$$K(x) = \{u \in U : L_i h_i(x) + L_i g h_i(x) u + \alpha_i(h_i(x)) \geq 0, \forall i \in I\}. \tag{4}$$

Then $h_i(x), i \in I$, being compatible CBFs is equivalent to that $K(x) \neq \emptyset, \forall x \in D$.

**Proposition 1.** If $h_i(x), i \in I$, are compatible CBFs, then any locally Lipschitz continuous feedback control law $u(x) \in K(x)$ renders the safety set $C$ forward invariant.

**Proof.** This is evident from the Brezis version of Nagumo’s Theorem at $\partial C$. Please check [14, Theorem 4] for details. □

When the system is subject to disturbances/uncertainty, a relevant concept is that of robustly compatible CBFs.

**Definition 2 (Robustly compatible CBFs).** The functions $h_i(x), i \in I$ are robustly compatible control barrier functions with robustness level $\eta > 0$ for system (1) if there exists an open set $D \supseteq C$ and locally Lipschitz extended class $K$ functions $\alpha_i$ such that, $\forall x \in D, \forall i \in I$,

$$\exists u \in U, L_i h_i(x) + L_i g h_i(x) u + \alpha_i(h_i(x)) \geq \eta. \tag{5}$$

The condition in (5) is stricter compared to the condition in (4). If (5) holds, then the safety set $C$ can be rendered forward invariant for the perturbed system $\dot{x} = f(x) + g(x)u + p(x)\omega$ as long as $|L_{p \omega} h_i(x)| \leq \eta, \forall i \in I, \forall x \in D$. The analysis follows similarly as in [15] [16, Remark 3].

A CBF-based safety controller $u : D \to \mathbb{R}^m$ is given in the following form

$$u(x) = \arg\min_v \|v - u_{nom}(x)\| \quad \text{s.t. } v \in K(x), \tag{6}$$

where $u_{nom}$ is a nominal controller focusing on task completion. For example, $u_{nom}$ can be designed for state stabilization, reference tracking or can be given directly by a human user. One core problem for the CBF-based controller formulation in (6) is the compatibility, i.e., $K(x) \neq \emptyset, \forall x \in D$. When external disturbances are present, robustly compatibility is a desired property for practical implementation.

In this work, we propose an algorithmic solution to verify or falsify the hypothesis that $h_i(x), i \in I$ are (robustly) compatible. The compatibility verification algorithm only needs to be executed once and offline, before applying the CBF-based safety controller (6) online. For notational brevity, given $h_i(x), \alpha_i(\cdot)$ and the control system in (1), we denote

$$A(x) \triangleq \begin{pmatrix} L_i h_1(x) \\ L_i h_2(x) \\ \vdots \\ L_i h_N(x) \end{pmatrix}, b(x) \triangleq \begin{pmatrix} L_i h_1(x) + \alpha_1(h_1(x)) \\ L_i h_2(x) + \alpha_2(h_2(x)) \\ \vdots \\ L_i h_N(x) + \alpha_N(h_N(x)) \end{pmatrix}. \tag{7}$$

The problem is thus to verify whether

$$\sup_{u \in U} A(x)u + b(x) \geq 0, \forall x \in D. \tag{8}$$

for compatibility, and whether

$$\sup_{u \in U} A(x)u + b(x) \geq \eta 1, \forall x \in D. \tag{9}$$

for robust compatibility with robustness level $\eta > 0$.

To simplify our analysis though without jeopardizing the generality, we assume the following:

**Assumption 1.** The safety set $C$ is compact.

**Assumption 2.** The input set $U$ is convex.

Under Assumption 2 and in view of (4), we know that $K(x)$ is either empty or convex, for any $x \in D$.

III. PROPOSED SOLUTIONS

A. Grid sampling algorithm using n-cubes

We recall some basic notions for approximating a compact set in $\mathbb{R}^n$ using n-cubes. Let $S \subset \mathbb{R}^n$ be a compact set, and $\{e_i, i = 1, 2, ..., n\}$ the canonical basis of $\mathbb{R}^n$. For $x \in \mathbb{R}^n, r > 0$, define

$$P_{\text{latice}}(x, r) = \{y \in \mathbb{R}^n : y = x + \sum_{i \in \{1, 2, ..., n\}} a_i r e_i, \forall a_i \in \mathbb{N}, i = \{1, 2, ..., n\}\}.$$

(9)
First we calculate the range limit $P$ product of the set $P$, compact, the fact that the hyperrectangle Algorithm 1, are finite for every dimension. This leads to a limit size.

Require: Compact set $P$, cardinality, which implies that $\rho\in\mathbb{R}$, centered at $x$ with size $r$.

Now we propose the following grid sampling algorithm. First we calculate the range limit $P^\text{min}$ and $P^\text{max}$, $i = 1, 2, \ldots, n$ of the set $S$ (Line 1 of Algorithm 1). Since $S$ is compact, $S$ is a subset of the hyperrectangle $[P^\text{min}_1, P^\text{max}_1] \times [P^\text{min}_2, P^\text{max}_2] \times \cdots \times [P^\text{min}_n, P^\text{max}_n]$ (Line 2). Then we construct a regular lattice $P$ around the center point of the hyperrectangle with size $r$. In Line 3, we obtain a set $P_{\text{cand}}$ by intersecting $P$ with the inflated hyperrectangle $[P^\text{min}_1 - r/2, P^\text{max}_1 + r/2] \times [P^\text{min}_2 - r/2, P^\text{max}_2 + r/2] \times \cdots \times [P^\text{min}_n - r/2, P^\text{max}_n + r/2]$. Then we construct $P_{\text{cand}}$ around which the $n$-cubes with size $r$ intersects with the set $S$ (Line 4). The algorithm returns $G$ as a Cartesian product of $P$ and the singleton $r$.

Algorithm 1 GridSampling

Require: Compact set $S \subset \mathbb{R}^n$, lattice size $r$
1. Calculate $P^\text{min} = \min_{x \in S} x^\top e_i, P^\text{max} = \max_{x \in S} x^\top e_i$ for $i = \{1, 2, \ldots, n\}$.
2. Construct a regular lattice $P$ around $(P^\text{min}_1 + P^\text{max}_1)/2, \ldots, (P^\text{min}_n + P^\text{max}_n)/2$ with size $r$.
3. Construct $P_{\text{cand}} = P_{\text{cand}} \cap \left[ P^\text{min}_1 - r/2, P^\text{max}_1 + r/2 \right] \times \left[ P^\text{min}_2 - r/2, P^\text{max}_2 + r/2 \right] \times \cdots \times \left[ P^\text{min}_n - r/2, P^\text{max}_n + r/2 \right]$.
4. $P = \{p \in P_{\text{cand}} : B(p, r) \cap S \neq \emptyset\}, G = P \times \{r\}$.
5. return $G$.

Proposition 2. Given a compact set $S \subset \mathbb{R}^n$ and a lattice size $r > 0$, then the following hold:
1) $G$, from Algorithm 1 is of finite cardinality, and
2) $S \subset \bigcup_{p \in P} B(p, r)$, where $P$ is given in Algorithm 1.

Proof. Since the set $S$ is compact, the lower and upper range limit $P^\text{min}$ and $P^\text{max}$, $i = 1, 2, \ldots, n$, given in Line 1 of Algorithm 1 are finite for every dimension. This leads to the fact that the hyperrectangle $[P^\text{min}_1 - r/2, P^\text{max}_1 + r/2] \times [P^\text{min}_2 - r/2, P^\text{max}_2 + r/2] \times \cdots \times [P^\text{min}_n - r/2, P^\text{max}_n + r/2]$ is bounded. Recall that by definition 2, $P_{\text{cand}}$ denotes a regular lattice in $\mathbb{R}^n$, and we thus know that $P_{\text{cand}}$ has a finite cardinality, which implies that $G$ also has a finite cardinality. Now we show Property 2) by contradiction. Assume that there exists $x \in S$ and $x \notin \bigcup_{p \in P} B(p, r)$. In view of the definition of $P$, this implies that $x \notin \bigcup_{p \in P_{\text{cand}}} B(p, r)$. This yields a contradiction since $x \in S \subset [P^\text{min}_1, P^\text{max}_1] \times [P^\text{min}_2, P^\text{max}_2] \times \cdots \times [P^\text{min}_n, P^\text{max}_n] \subset \bigcup_{p \in P_{\text{cand}}} B(p, r)$. The former set inclusion is trivial in view of the definition of $P_{\text{cand}}$ and the hyperrectangle.

Proposition 3. Here $\|A\|_{\infty}$, where $A$ is a matrix, refers to the induced matrix norm and can be calculated as the maximum absolute row sum of $A$.

From now on, we denote $Bound(S)$ the bounding box $[P^\text{min}_1 - r/2, P^\text{max}_1 + r/2] \times [P^\text{min}_2 - r/2, P^\text{max}_2 + r/2] \times \cdots \times [P^\text{min}_n - r/2, P^\text{max}_n + r/2]$ of a compact set $S$.

Example 1. Here we show an example of Algorithm 1 with the set $S = \{x \in \mathbb{R}^2 : 1 \leq x^\top Qx \leq 2\}$, where $Q = \begin{pmatrix} 0.5 & 0.1 \\ 0.1 & 0.3 \end{pmatrix}$ and $r = 0.25$. From Fig. 1 we observe that $G$ has a finite cardinality and $S \subset \bigcup_{p \in P} B(p, r)$. It is worth noting that $\bigcup_{p \in P} B(p, r) \subset Bound(S)$, where $P$ is given in Algorithm 1 Line 4, as shown in Fig. 1.

B. Proposed verification algorithm

Now consider the compatibility verification problem in (7). For any $x \in Bound(C)$, define
\[
c(x) = \max_{u \in \mathbb{U}} t s.t. A(x)u + b(x) \geq t1_N,\]
\[
u \in \mathbb{U}.\]

In the case that $U$ is a polytopic set, $c(x)$ is obtained by solving a linear program. In the general case where $U$ is convex, $c(x)$ is obtained from a convex optimization. One interpretation is that $c(x)$ indicates the largest robustness level at $x$ up to which the CBF conditions or the input constraints are to be breached.

Recall that the candidate CBFs $h_i(x)$, $i = 1, 2, \ldots, N$ are continuously differentiable, the vector fields $f(x)$ and $g(x)$ are locally Lipschitz, and thus $A(x), b(x)$ in (7) are locally Lipschitz. Specifically, denote the respective Lipschitz constants in the bounding box $Bound(C)$ with respect to the $l_\infty$ norm as $L_{A, \infty}, L_{b, \infty}$, i.e.,
\[
\|A(x) - A(x')\|_{\infty} \leq L_{A, \infty}\|x - x'\|_{\infty},\]
\[
\|b(x) - b(x')\|_{\infty} \leq L_{b, \infty}\|x - x'\|_{\infty},\]
for all $x, x' \in Bound(C)$

Here $\|A\|_{\infty}$, where $A$ is a matrix, refers to the induced matrix norm and can be calculated as the maximum absolute row sum of $A$.
If $c(x) > 0$ for some $x$, based on the Lipschitz continuity of $A(x)$ and $b(x)$, there must exist a neighborhood around $x$ where the CBFs $h_i(x)$ are compatible. This is formally shown below.

**Proposition 3.** For any $x \in \text{Bound}(C)$, if $c(x) > 0$, then \( \sup_{x' \in \text{U}} A(x')v + b(x') \geq 0 \) for all $x' \in B(x, \rho(x)) \cap \text{Bound}(C)$ with
\[
\rho(x) = \frac{2c(x)}{L_{\infty} \| u^*(x) \|_{\infty} + L_{b, \infty}},
\]
where $L_{\infty}, L_{b, \infty}$ are the Lipschitz constants of $A(x), b(x)$ with respect to the $l_\infty$ norm as per (12), respectively, and $u^*(x)$ is the optimal solution to (11) at $x$.

**Proof.** For any $x' \in B(x, \rho) \cap \text{Bound}(C)$, we have
\[
A(x')u^*(x) + b(x') = (A(x') - A(x))u^*(x) + (b(x') - b(x)) + A(x)u^*(x) + b(x)
\]
In view of (12), we obtain
\[
\| A(x') - A(x) \|_{\infty} \| u^*(x) \|_{\infty} + \| b(x') - b(x) \|_{\infty} \leq \frac{2c(x)}{L_{\infty} \| u^*(x) \|_{\infty} + L_{b, \infty}} \| x - x' \|_{\infty}
\]
In view of (11), we obtain $\| x - x' \|_{\infty} \leq \rho/2 = \frac{2c(x)}{L_{\infty} \| u^*(x) \|_{\infty} + L_{b, \infty}} \rho(x)$. Thus, $\| A(x') - A(x) \|_{\infty} \| u^*(x) \|_{\infty} + \| b(x') - b(x) \|_{\infty} \leq c(x)$. From (14) and $A(x)u^*(x) + b(x) \geq 0$, we further obtain $A(x')u^*(x) + b(x') \geq 0$, which completes the proof.

**Algorithm 2 CompatibilityChecking**

**Require:** $h_i(x), \alpha_i(\cdot)$, initial size $r_0$, decaying factor $\lambda$
1. **Initialization:**
2. $k = 0$; obtain $C$ from (2), $G_0 \leftarrow \text{GS}(C, r_0)$, $G_1 = \emptyset$.
3. while $G_k \neq \emptyset$ do
4. for each $(x, r) \in G_k$ do
5. $c \leftarrow c(x)$ from (11), $\rho \leftarrow \rho(x)$ from (13).
6. if $c < 0$ then $\triangleright$ Found an incompatible state; return False.
7. if $\rho \geq r$ then $\triangleright$ Compatibility checked; remove $(x, r)$ from $G_k$.
8. else $\triangleright$ Compatibility partially checked; remove $(x, r)$ from $G_k$, $r' \leftarrow \lambda r$.
9. $G_{k+1} \leftarrow G_{k+1} \cup \text{GS}(B(x, r) \setminus B(x, r'), r')$.
10. end if
11. end for
12. $k = k + 1$, $G_{k+2} = \emptyset$.
13. end while
14. return True.

*GS stands for GridSampling given in Algorithm 1*

Built on above analysis, we design a compatibility checking algorithm using grid sampling and refinement. As given in Algorithm 2, the safety set $C$ is firstly over-approximated using GridSampling Algorithm with an initial lattice size $r_0$. This will yield a finite set $G_0$ of $n$-cubes that is to be checked later. Recall that in Problem formulation (7) and (8), we need to check the compatibility over a set $D \supseteq C$. Here we take $D = \bigcup_{(x, r_0) \in G_k} B(x, r_0)$, which is a super set of $C$ from Proposition 2 item 2). Choosing $r_0$ is important and depends on how large buffering zone one allows outside the safety set. For each $n$-cube $B(x, r)$ in $G_k$, represented as a $(x, r)$ pair in Line 4, we calculate the robustness level $c$ and the size $\rho$ of a guaranteed compatible $n$-cube centered at $x$ from (11) and (13), respectively. If $c < 0$, then an incompatible state is found and the algorithm terminates and returns False. If $\rho > r$, then we know that the CBFs are compatible for all the states within the $n$-cube $B(x, r)$ and we remove $(x, r)$ from $G_k$; otherwise, we refine the remaining unchecked region $B(x, r) \setminus B(x, \rho)$ with a discounted lattice size $\lambda r = \lambda r'$ and include the new $n$-cubes in $G_{k+1}$. After checking all the $n$-cubes in $G_k$, we iterate the process again for $G_{k+1}$. Once $G_{k+1} = \emptyset$, the algorithm terminates and returns True.

The following properties provide a guarantee on the finite-step termination of the algorithm and the compatibility property certified from its termination.

**Theorem 1.** Given control barrier functions $h_i(x)$, extended class $K$ functions $\alpha_i(\cdot)$ with $i \in I$, an initial lattice size $r_0 > 0$ and a decaying factor $0 < \lambda < 1$, we have:
1. If Algorithm 2 terminates, it gives verification or falsification on the CBF compatibility as per Def. 2).
2. If $I$ is bounded, and the CBFs $h_i(x)$ are robustly compatible with robustness level $\eta > 0$ in Bound(C), then Algorithm 2 terminates in finite steps.
3. If $I$ is bounded, and a lower bound of the lattice size $r$ is incorporated, i.e., Algorithm 2 terminates if $r < \eta$ in Line 4, then Algorithm 2 terminates in finite steps and gives one of the following three results:
   i. $h_i(x), i \in I$ are compatible;
   ii. $h_i(x), i \in I$ are incompatible;
   iii. $h_i(x), i \in I$ are not robustly compatible with robust level greater than
\[
\eta' = \lambda^{-1} (\max_{u \in I} L_{\infty} \| u \|_{\infty} + L_{b, \infty})/2.
\]

Proof to this theorem is omitted here due to page limits. Interested readers are referred to [17] for details as well as a thorough discussion on the proposed algorithm.

**IV. CASE STUDIES**

In this section we show more details on the algorithm implementation, especially the Lipschitz constant calculation, and demonstrate the efficacy of our proposed verification algorithm in several different scenarios. All the simulations are done using Matlab Parallel Computing Toolbox on an Intel i7-8650U CPU laptop.

**Example 2.** Consider a $2 - D$ system with state variable $x = (x_1, x_2)$, input variable $u = (u_1, u_2)$, dynamics
\[
\begin{align*}
\dot{x}_1 &= x_1 + x_2 + 1, \\
\dot{x}_2 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},
\end{align*}
\]
with $f(x) = a(x)$.
The equality holds due to the definition of induced matrix norm. This reveals that $L_{A,\infty} = 4\|Q\|_1 = 2.4$ satisfies (18). Similarly, $L_{b,\infty}$ needs to satisfy

$$\|b(x) - b(x')\|_\infty \leq L_{b,\infty} \|x - x'\|_\infty$$

for any $x, x' \in \text{Bound}(C)$. Let $b_i(x)$ be the $i$th row of $b(x)$. Thus, (21) is equivalent to $|b_i(x) - b_i(x')| \leq L_{b,\infty} \|x - x'\|_\infty, \forall i \in \{1, 2\}$, for any $x, x' \in \text{Bound}(C)$. Recall that

$$b_1(x) = 2x^\top Q_1 f(x) + x^\top Q_2 x - 1 \quad (22)$$

$$b_2(x) = 2x^\top Q_2 f(x) - x^\top Q_2 x + 2 \quad (23)$$

We have $\nabla b_1(x) = 2Qf(x) + 2\frac{\partial f}{\partial x}(x)Qx + 2Qx, \nabla b_2(x) = 2Qf(x) + 2\frac{\partial f}{\partial x}(x)Qx - 2Qx$, where $\frac{\partial f}{\partial x}(x) = \left(-\frac{1}{5}, \frac{1}{5}\right)$. Following Mean Value Theorem, we know

$$|b_i(x) - b_i(x')| \leq \max_{v \in \text{Bound}(C)} \|\nabla b_i(v)\|_\infty \|x - x'\|_\infty, \forall x, x' \in \text{Bound}(C), \forall i \in \{1, 2\}. \text{ Thus we choose}$$

$$L_{b,\infty} \geq \max_{i=1,2} \max_{v \in \text{Bound}(C)} (\|\nabla b_i(v)\|_\infty).$$

This leads to solve two quadratic programs and we obtain $L_{b,\infty} = 13$.

Now we have all the necessary elements to execute CompatibilityChecking. Choose $\lambda = 0.25$. The algorithm terminates after 3 iterations and verifies the compatibility of the two CBFs. The execution process takes 169s and is shown in Fig. 2.

**Example 3.** Now we consider the same scenario as in Example 2 but with a more stringent input set $U = \{(u_1, u_2) : |u_1| \leq 2, |u_2| \leq 2\}$. This time, CompatibilityChecking gives a falsification on the compatibility. It finds an inconsistent state $x_{in} = (-1.5, -1.25)$, at which point

Here $\|Q\|_1$, where $Q$ is a matrix, refers to the induced matrix norm and can be calculated as the maximum absolute column sum of $Q$. 

---

**Fig. 2:** Execution process of CompatibilityChecking in Example 2. The safety region is between the two ellipsoids. Compatible 2-cubes: 2-cubes within which the CBFs are verified to be compatible (in green); to-be-refined 2-cubes: 2-cubes that need further refinement (in yellow). (a) First iteration with the lattice size $r = 0.25$. All of the total 200 2-cubes are to-be-refined 2-cubes. (b) Second iteration with the lattice size $r = 0.0625$. The refined 2-cubes are checked and 3204 out of 4809 are verified to be compatible 2-cubes, and 1665 2-cubes are refined again. (c) Third iteration with the lattice size $r = 0.0156$. All the 2-cubes are compatible. Algorithm 2 thus gives verification on the compatibility of the CBFs.
This work focuses on the compatibility checking of multiple CBFs. Future directions include how to determine the extended class $K$ functions that mitigate the possible incompatibility and/or increase the robustness level, and how to incorporate the compatibility as a constraint with the online QP to ensure recursive feasibility.

REFERENCES

[1] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, “Constrained model predictive control: Stability and optimality,” Automatica, vol. 36, no. 6, pp. 789–814, 2000.
[2] A. Bemporad, “Reference governor for constrained nonlinear systems,” IEEE Transactions on Automatic Control, vol. 43, no. 3, pp. 415–419, 1998.
[3] K. P. Tee, S. S. Ge, and E. H. Tay, “Barrier lyapunov functions for the control of output-constrained nonlinear systems,” Automatica, vol. 45, no. 4, pp. 918–927, 2009.
[4] C. P. Bechlioulis and G. A. Rovithakis, “Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance,” IEEE Transactions on Automatic Control, vol. 53, no. 9, pp. 2090–2099, 2008.
[5] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” IFAC Proceedings Volumes, vol. 40, no. 12, pp. 462–467, 2007.
[6] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” in Proc. IFAC Conf. Anal. Design Hybrid Syst., vol. 48, 2015, pp. 54–61.
[7] X. Xu, “Constrained control of input-output linearizable systems using control sharing barrier functions,” Automatica, vol. 87, pp. 195–201, 2018.
[8] W. S. Cortez, X. Tan, and D. V. Dimarogonas, “A robust, multiple control barrier function framework for input constrained systems,” IEEE Control Systems Letters, vol. 6, pp. 1742–1747, 2021.
[9] G. Notomista and M. Saveriano, “Safety of dynamical systems with multiple non-convex unsafe sets using control barrier functions,” IEEE Control Systems Letters, vol. 6, pp. 1136–1141, 2021.
[10] 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.
[11] A. Clark, “Verification and synthesis of control barrier functions,” in 2021 60th IEEE Conference on Decision and Control (CDC), 2021, pp. 6105–6112.
[12] I. Axton, G. Masoumeh, G. S. Ricardo, and E. D. Warren, “On the feasibility and continuity of feedback controllers defined by multiple control barrier functions,” in 2022 American Control Conference (ACC). IEEE, 2022.
[13] H. K. Khalil, Nonlinear Systems. Upper Saddle River, N. J.: Prentice Hall, c2002, 2002.
[14] R. Redheffer, “The theorems of Bony and Brezis on flow-invariant sets,” The American Mathematical Monthly, vol. 79, no. 7, pp. 740–747, 1972.
[15] M. Jankovic, “Robust control barrier functions for constrained stabilization of nonlinear systems,” Automatica, vol. 96, pp. 359–367, 2018.
[16] 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.
[17] X. Tan and D. V. Dimarogonas, “Compatibility checking of multiple control barrier functions for input constrained systems,” arXiv preprint arXiv:2209.02284, 2022.