# Compatibility checking of multiple CBFs for input constrained systems

Xiao Tan and Dimos V. Dimarogonas

Division of Decision and Control Systems
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology, Sweden

IEEE-CDC 2022

# Preliminary

- Consider a control-affine system

$$\dot{\boldsymbol{x}} = \mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}, \boldsymbol{u} \in \mathbb{U}.$$

- Multiple state constraints $\Rightarrow \mathcal{C} = \{\boldsymbol{x} : h_i(\boldsymbol{x}) \geq 0, i \in \mathcal{I}\}$.

# Preliminary

- Consider a control-affine system

$$\dot{\boldsymbol{x}} = \mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}, \boldsymbol{u} \in \mathbb{U}.$$

- Multiple state constraints $\Rightarrow \mathcal{C} = \{\boldsymbol{x} : h_i(\boldsymbol{x}) \geq 0, i \in \mathcal{I}\}$.
- Recall the CBF-based controller with one CBF $h_i(\boldsymbol{x})$

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) = \arg\min_{\boldsymbol{v}\in\mathbb{U}} \|\boldsymbol{v} - \boldsymbol{u}_{nom}(\boldsymbol{x})\| \\ \text{s.t. } L_{\mathfrak{f}} h_i(\boldsymbol{x}) + L_{\mathfrak{g}} h_i(\boldsymbol{x})\boldsymbol{v} + \alpha_i(h_i(\boldsymbol{x})) \geq 0 \end{aligned} \quad (1)$$

# Preliminary

- Consider a control-affine system

$$\dot{\boldsymbol{x}} = \mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}, \boldsymbol{u} \in \mathbb{U}.$$

- Multiple state constraints $\Rightarrow \mathcal{C} = \{\boldsymbol{x} : h_i(\boldsymbol{x}) \geq 0, i \in \mathcal{I}\}$.
- Recall the CBF-based controller with one CBF $h_i(\boldsymbol{x})$

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) &= \arg\min_{\boldsymbol{v} \in \mathbb{U}} \|\boldsymbol{v} - \boldsymbol{u}_{nom}(\boldsymbol{x})\| \\ \text{s.t. } & L_{\mathfrak{f}} h_i(\boldsymbol{x}) + L_{\mathfrak{g}} h_i(\boldsymbol{x})\boldsymbol{v} + \alpha_i(h_i(\boldsymbol{x})) \geq 0 \end{aligned} \tag{1}$$

- The CBF-based controller with multiple CBFs

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) &= \arg\min_{\boldsymbol{v} \in \mathbb{U}} \|\boldsymbol{v} - \boldsymbol{u}_{nom}(\boldsymbol{x})\| \\ \text{s.t. } & L_{\mathfrak{f}} h_i(\boldsymbol{x}) + L_{\mathfrak{g}} h_i(\boldsymbol{x})\boldsymbol{v} + \alpha_i(h_i(\boldsymbol{x})) \geq 0, \forall i \in \mathcal{I}. \end{aligned} \tag{2}$$

# Preliminary

- Consider a control-affine system

$$\dot{\boldsymbol{x}} = \mathfrak{f}(\boldsymbol{x}) + \mathfrak{g}(\boldsymbol{x})\boldsymbol{u}, \boldsymbol{u} \in \mathbb{U}.$$

- Multiple state constraints $\Rightarrow \mathcal{C} = \{\boldsymbol{x} : h_i(\boldsymbol{x}) \geq 0, i \in \mathcal{I}\}$.
- Recall the CBF-based controller with one CBF $h_i(\boldsymbol{x})$

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) = \arg\min_{\boldsymbol{v} \in \mathbb{U}} \|\boldsymbol{v} - \boldsymbol{u}_{nom}(\boldsymbol{x})\| \\ \text{s.t. } L_{\mathfrak{f}} h_i(\boldsymbol{x}) + L_{\mathfrak{g}} h_i(\boldsymbol{x})\boldsymbol{v} + \alpha_i(h_i(\boldsymbol{x})) \geq 0 \end{aligned} \tag{1}$$

- The CBF-based controller with multiple CBFs

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}) = \arg\min_{\boldsymbol{v} \in \mathbb{U}} \|\boldsymbol{v} - \boldsymbol{u}_{nom}(\boldsymbol{x})\| \\ \text{s.t. } L_{\mathfrak{f}} h_i(\boldsymbol{x}) + L_{\mathfrak{g}} h_i(\boldsymbol{x})\boldsymbol{v} + \alpha_i(h_i(\boldsymbol{x})) \geq 0, \forall i \in \mathcal{I}. \end{aligned} \tag{2}$$

- (2) is feasible for all $\boldsymbol{x} \in \mathcal{D} \supseteq \mathcal{C} \Leftrightarrow h_i(\boldsymbol{x}), i \in \mathcal{I}$ are compatible.

# Preliminary

In case of perturbations, we define $h_i(\boldsymbol{x}), i \in \mathcal{I}$ are *robustly compatible* with robustness level $\eta > 0$ if $\forall \boldsymbol{x} \in \mathcal{D}$,

$$\exists \boldsymbol{u} \in \mathbb{U},\ L_{\mathfrak{f}} h_i(\boldsymbol{x}) + L_{\mathfrak{g}} h_i(\boldsymbol{x}) \boldsymbol{u} + \alpha_i(h_i(\boldsymbol{x})) \geq \eta, \forall i \in \mathcal{I}. \tag{3}$$

In this work, we propose an algorithmic solution to verify or falsify the hypothesis that $h_i(\boldsymbol{x}), i \in \mathcal{I}$ are (robustly) compatible. This algorithm will run once and offline before online implementation.

# Problem statement

For notational brevity, given $h_i(\boldsymbol{x}), \alpha_i(\cdot)$ and the system dynamics, we denote

$$A(\boldsymbol{x}) := \begin{pmatrix} L_{\mathfrak{g}} h_1(\boldsymbol{x}) \\ L_{\mathfrak{g}} h_2(\boldsymbol{x}) \\ \dots \\ L_{\mathfrak{g}} h_N(\boldsymbol{x}) \end{pmatrix}, b(\boldsymbol{x}) := \begin{pmatrix} L_{\mathfrak{f}} h_1(\boldsymbol{x}) + \alpha_1(h_1(\boldsymbol{x})) \\ L_{\mathfrak{f}} h_2(\boldsymbol{x}) + \alpha_2(h_2(\boldsymbol{x})) \\ \dots \\ L_{\mathfrak{f}} h_N(\boldsymbol{x}) + \alpha_N(h_N(\boldsymbol{x})) \end{pmatrix}.$$

The problem is thus to verify whether

$$\sup_{\boldsymbol{u} \in \mathbb{U}} A(\boldsymbol{x})\boldsymbol{u} + b(\boldsymbol{x}) \geq \mathbf{0}, \forall \boldsymbol{x} \in \mathcal{D}. \tag{4}$$

for compatibility, and whether

$$\sup_{\boldsymbol{u} \in \mathbb{U}} A(\boldsymbol{x})\boldsymbol{u} + b(\boldsymbol{x}) \geq \eta \mathbf{1}, \forall \boldsymbol{x} \in \mathcal{D}. \tag{5}$$

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

# Problem statement

For notational brevity, given $h_i(\boldsymbol{x}), \alpha_i(\cdot)$ and the system dynamics, we denote

$$A(\boldsymbol{x}) := \begin{pmatrix} L_{\mathfrak{g}} h_1(\boldsymbol{x}) \\ L_{\mathfrak{g}} h_2(\boldsymbol{x}) \\ \dots \\ L_{\mathfrak{g}} h_N(\boldsymbol{x}) \end{pmatrix}, b(\boldsymbol{x}) := \begin{pmatrix} L_{\mathfrak{f}} h_1(\boldsymbol{x}) + \alpha_1(h_1(\boldsymbol{x})) \\ L_{\mathfrak{f}} h_2(\boldsymbol{x}) + \alpha_2(h_2(\boldsymbol{x})) \\ \dots \\ L_{\mathfrak{f}} h_N(\boldsymbol{x}) + \alpha_N(h_N(\boldsymbol{x})) \end{pmatrix}.$$

The problem is thus to verify whether

$$\sup_{\boldsymbol{u} \in \mathbb{U}} A(\boldsymbol{x})\boldsymbol{u} + b(\boldsymbol{x}) \geq \mathbf{0}, \forall \boldsymbol{x} \in \mathcal{D}. \tag{4}$$

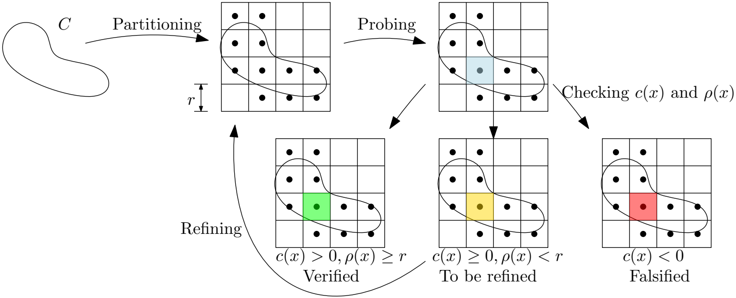
for compatibility, and whether

$$\sup_{\boldsymbol{u} \in \mathbb{U}} A(\boldsymbol{x})\boldsymbol{u} + b(\boldsymbol{x}) \geq \eta \mathbf{1}, \forall \boldsymbol{x} \in \mathcal{D}. \tag{5}$$

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

For simplicity, we assume 1) $\mathcal{C}$ is compact and 2) $\mathbb{U}$ is convex.

# Proposed scheme: Overview




Sampling with $n$-cubes

$$B(\boldsymbol{x}, r) = \{\boldsymbol{y} \in \mathbb{R}^n : \boldsymbol{y} = \boldsymbol{x} + \sum_i k_i r \boldsymbol{e}_i, \forall k_i \in [-1/2, 1/2]\}.$$

# Grid sampling

**Algorithm 1** `GridSampling`

**Require:** Compact set $\mathcal{S} \subset \mathbb{R}^n$, lattice size $r$

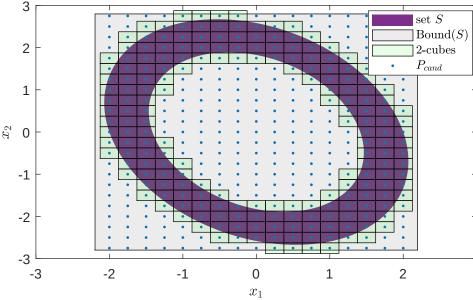
1: Calculate $\rho^{\min}_{\boldsymbol{e}_i} = \min_{\boldsymbol{x}\in\mathcal{S}} \boldsymbol{e}_i^\top \boldsymbol{x}, \rho^{\max}_{\boldsymbol{e}_i} = \max_{\boldsymbol{x}\in\mathcal{S}} \boldsymbol{e}_i^\top \boldsymbol{x}$ for $i \in \{1, 2, ..., n\}$.

2: Construct a regular lattice $P_{\text{lattice}}$ around $(\frac{\rho^{\min}_{\boldsymbol{e}_1}+\rho^{\max}_{\boldsymbol{e}_1}}{2}, \frac{\rho^{\min}_{\boldsymbol{e}_2}+\rho^{\max}_{\boldsymbol{e}_2}}{2}, ..., \frac{\rho^{\min}_{\boldsymbol{e}_n}+\rho^{\max}_{\boldsymbol{e}_n}}{2})$ with size $r$.

3: Construct $P_{\text{cand}} = P_{\text{lattice}} \cap [\rho^{\min}_{\boldsymbol{e}_1} - r/2, \rho^{\max}_{\boldsymbol{e}_1} + r/2] \times [\rho^{\min}_{\boldsymbol{e}_2} - r/2, \rho^{\max}_{\boldsymbol{e}_2} + r/2] \times ... \times [\rho^{\min}_{\boldsymbol{e}_n} - r/2, \rho^{\max}_{\boldsymbol{e}_n} + r/2]$.

4: $P = \{\boldsymbol{p} \in P_{\text{cand}} : B(\boldsymbol{p}, r) \cap \mathcal{S} \neq \emptyset\}, G = P \times \{r\}$.

5: **return** $G$.




The following holds:

**1)** $G$, from Algorithm 1, is of finite cardinality, and

**2)** $\mathcal{S} \subseteq \cup_{\boldsymbol{p}\in P} B(\boldsymbol{p}, r)$, where $P$ is the set of sampling points.

# Verification algorithm: Lipschitz properties

- For any $\boldsymbol{x} \in \text{Bound}(\mathcal{C})$, define

$$\begin{aligned} c(\boldsymbol{x}) = & \max_{\boldsymbol{u},t} t \\ \text{s.t. } & A(\boldsymbol{x})\boldsymbol{u} + b(\boldsymbol{x}) \geq t\mathbf{1}_N, \\ & \boldsymbol{u} \in \mathbb{U}. \end{aligned} \tag{6}$$

  - convex optimization; $c(\boldsymbol{x})$: largest robustness level at $\boldsymbol{x}$.

- Recall $A(\boldsymbol{x}), b(\boldsymbol{x})$ are Lipschitz functions, let the respective Lipschitz constants in $\text{Bound}(\mathcal{C})$ w.r.t. the $l_\infty$ norm as $L_{A,\infty}, L_{b,\infty}$.

# Verification algorithm: Lipschitz properties

- For any $\boldsymbol{x} \in \text{Bound}(\mathcal{C})$, define

$$\begin{aligned} c(\boldsymbol{x}) = &\max_{\boldsymbol{u},t} t \\ \text{s.t. } & A(\boldsymbol{x})\boldsymbol{u} + b(\boldsymbol{x}) \geq t\mathbf{1}_N, \\ & \boldsymbol{u} \in \mathbb{U}. \end{aligned} \tag{6}$$

  - convex optimization; $c(\boldsymbol{x})$: largest robustness level at $\boldsymbol{x}$.

- Recall $A(\boldsymbol{x}), b(\boldsymbol{x})$ are Lipschitz functions, let the respective Lipschitz constants in $\text{Bound}(\mathcal{C})$ w.r.t. the $l_\infty$ norm as $L_{A,\infty}, L_{b,\infty}$.

- Given any $\boldsymbol{x} \in \text{Bound}(\mathcal{C})$, $c(\boldsymbol{x}) > 0$, then for all $\boldsymbol{x}' \in B(\boldsymbol{x}, \rho(\boldsymbol{x}))$ $\cap\text{Bound}(\mathcal{C})$, $\sup_{\boldsymbol{v}\in\mathbb{U}} A(\boldsymbol{x}')\boldsymbol{v} + b(\boldsymbol{x}') \geq \mathbf{0}$ holds with

$$\rho(\boldsymbol{x}) = \frac{2c(\boldsymbol{x})}{L_{A,\infty}\|\boldsymbol{u}^\star(\boldsymbol{x})\|_\infty + L_{b,\infty}}, \tag{7}$$

  where $\boldsymbol{u}^\star(\boldsymbol{x})$ is the optimal solution to (6) at $\boldsymbol{x}$.

# Verification algorithm

**Algorithm 2** `CompatibilityChecking`

**Require:** $h_i(\boldsymbol{x}), \alpha_i(\cdot)$, initial size $r_0$, decaying factor $\lambda$

**Initialization:**<br>
    $k = 0$, obtain $\mathcal{C}$, $G_0 \leftarrow \texttt{GS}(\mathcal{C}, r_0)$, $G_1 = \emptyset$.<br>
**while** $G_k \neq \emptyset$ **do**<br>
    **for** each $(\boldsymbol{x}, r) \in G_k$ **do**<br>
        $c \leftarrow c(\boldsymbol{x})$, $\rho \leftarrow \rho(\boldsymbol{x})$.<br>
        **if** $c < 0$ **then** ▷ Found an incompatible state;<br>
            **return False**.<br>
        **else if** $\rho \geq r$ **then** ▷ Compatibility checked;<br>
            remove $(\boldsymbol{x}, r)$ from $G_k$.<br>
        **else** ▷ Compatibility partially checked;<br>
            remove $(\boldsymbol{x}, r)$ from $G_k$, $r' \leftarrow \lambda r$.<br>
            $G_{k+1} \leftarrow G_{k+1} \cup \texttt{GS}(B(\boldsymbol{x}, r) \setminus B(\boldsymbol{x}, \rho), r')$.<br>
        **end if**<br>
    **end for**<br>
    $k = k + 1, G_{k+2} = \emptyset$.<br>
**end while**<br>
**return True**.

*`GS` stands for `GridSampling` given in Algorithm 1.

# Verification guarantees

**Theorem 1.** *Given $h_i(\boldsymbol{x})$,$\alpha_i(\cdot)$ with $i \in \mathcal{I}$, an initial lattice size $r_0 > 0$ and $0 < \lambda < 1$, we have:*

1) *If Algorithm 2 terminates, it gives verification or falsification on the CBF compatibility;*
2) *if the CBFs $h_i(\boldsymbol{x})$ are robustly compatible with robustness level $\eta > 0$ in $\text{Bound}(\mathcal{C})$, then Algorithm 2 terminates in finite steps.*
3) *If a lower bound of the lattice size $\underline{r}$ is incorporated, i.e., Algorithm 2 terminates if $r \leq \underline{r}$ in Line 4, then Algorithm 2 terminates in finite steps and gives one of the following three results:*
   i. $h_i(\boldsymbol{x}), i \in \mathcal{I}$ *are compatible;*
   ii. $h_i(\boldsymbol{x}), i \in \mathcal{I}$ *are incompatible;*
   iii. $h_i(\boldsymbol{x}), i \in \mathcal{I}$ *are not robustly compatible with robust level greater than*

$$\eta' = \lambda^{-1}\underline{r}(\max_{\boldsymbol{u}\in\mathbb{U}} L_{A,\infty}\|\boldsymbol{u}\|_\infty + L_{b,\infty})/2.$$

# Some discussions

- Computational concerns: exponential growth of $\#$ of $n$-cubes
  - check only around the safety boundary;
  - process $n-$cubes in parallel.
- Generalization to a time-varying setting
  - Let $\tilde{x} := (\boldsymbol{x}, t)$. Then, $\dot{\tilde{\boldsymbol{x}}} = \begin{pmatrix} \mathfrak{f}(\tilde{\boldsymbol{x}}) \\ 1 \end{pmatrix} + \begin{pmatrix} \mathfrak{g}(\tilde{\boldsymbol{x}}) \\ \mathbf{0} \end{pmatrix} \boldsymbol{u}$ and $\tilde{\mathcal{C}} = \{\tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} : h_i(\tilde{\boldsymbol{x}}) \geq 0, \forall i \in \mathcal{I}\}$.
  - Only a bounded time interval can be considered.
- Alternative grid sampling methods:
  - $n$-spheres $B_S(\boldsymbol{x}, r) = \{\boldsymbol{y} \in \mathbb{R}^n : \|\boldsymbol{y} - \boldsymbol{x}\| \leq r\}$
  - However, generating $n$-spheres with a small overlapping ratio is difficult in high-dimensional spaces.
- Other improvements
  - more precise $L_{A,\infty}, L_{b,\infty}$;
  - updated lattice size using $r' \leftarrow \min(\rho, \lambda r)$

# Case studies: Ex.1

**Example 1:** Consider a $2-D$ system with $\boldsymbol{x} = (x_1, x_2)$, $\boldsymbol{u} = (u_1, u_2)$, dynamics

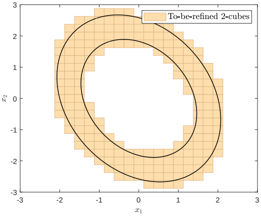
$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \underbrace{\begin{pmatrix} x_1 + x_2 \\ -x_1^2/2 \end{pmatrix}}_{\mathfrak{f}(\boldsymbol{x})} + \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\mathfrak{g}(\boldsymbol{x})} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix},$$

and $\mathbb{U} = \{(u_1, u_2) : |u_1| \leq 3, |u_2| \leq 3\}$. The two CBF candidates are

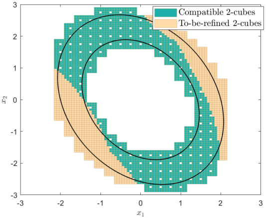
$$h_1(\boldsymbol{x}) = \boldsymbol{x}^\top Q \boldsymbol{x} - 1$$
$$h_2(\boldsymbol{x}) = 2 - \boldsymbol{x}^\top Q \boldsymbol{x}$$

where $Q = \left(\begin{smallmatrix} 0.5 & 0.1 \\ 0.1 & 0.3 \end{smallmatrix}\right)$. $\mathcal{C} = \{\boldsymbol{x} : h_i(\boldsymbol{x}) \geq 0, i = 1, 2\}$. The extended class $\mathcal{K}$ functions are chosen as $\alpha_1(v) = v, \alpha_2(v) = v, v \in \mathbb{R}$.

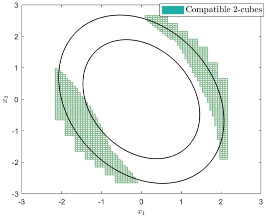
# Case studies: Ex.1




(a) First iteration, $r = 0.25$.




(b) Second iteration, $r = 0.0625$.

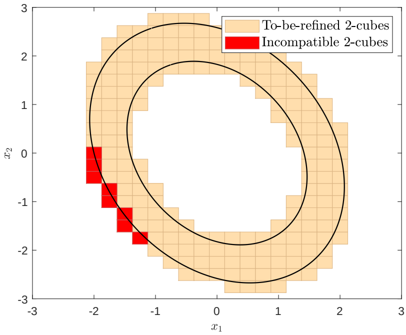



(c) Third iteration, $r = 0.0156$.

All the 2-cubes are compatible. The compatibility of the CBFs is verified.

# Case studies: Ex.2

**Example 2:** Same scenario as before but with $\mathbb{U} = \{(u_1, u_2) : |u_1| \le 2, |u_2| \le 2\}$.




An incompatible state $\boldsymbol{x}_{in} = (-1.5, -1.25)$ is found, at which $c(\boldsymbol{x}_{in}) = -0.36$.

# Case studies: Ex. 3

**Example 3:** Same scenario as Ex. $1$ but a lower bound $\underline{r} = 0.016$ is incorporated in Algorithm 2.

`CompatibilityChecking` terminates after $2$ iterations and gives a result that the multiple CBFs are at most robustly compatible with a robustness level $\eta = 0.6464$. This is validated by, for example, considering that $c(-1.5, -1.25) = 0.5$.

# Future directions

**1** How to determine the extended class $\mathcal{K}$ functions that mitigate the possible incompatibility and/or increase the robustness level;

**2** How to incorporate the compatibility as a constraint with the online QP to ensure recursive feasibility.