Barrier certificate with Interval Analysis

A review of Adel's PhD thesis

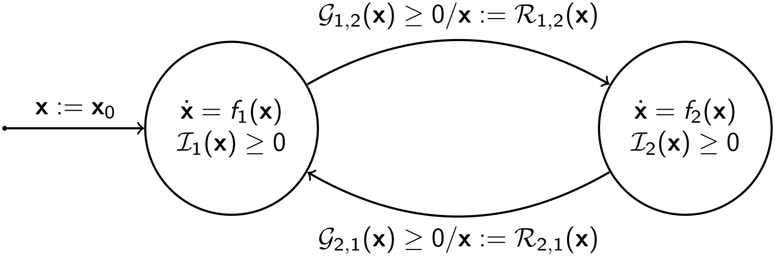
A. Chapoutot

ENSTA ParisTech, U2IS
joint work with
O. Bouissou, A. Djaballah, and M. Kieffer

November 25, 2015

Main goal

Formal verification of safety properties of hybrid systems described by hybrid automata.



Barrier certificate has been considered to achieve this goal and a 2 step process has been considered:

1. for continuous-time dynamical systems
2. then for hybrid automata

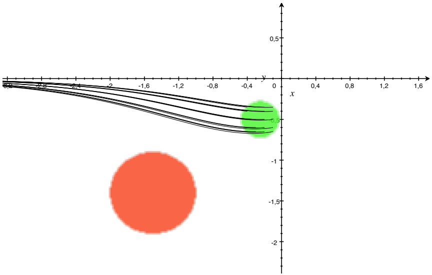
Safety of continuous dynamical systems

Consider a non-linear dynamical system S

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{d})$$

with $\mathbf{d} \in \mathcal{D}$ a constant and bounded disturbance

S is safe iff all trajectories starting from the initial region do not reach the unsafe region.

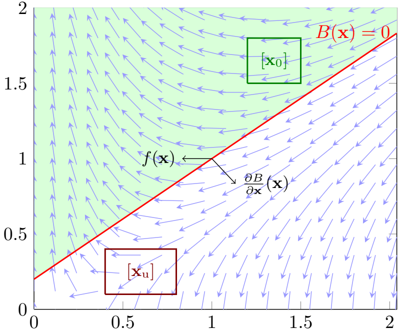


Barrier certificates

Main idea of [Prajna&Jadbabaie HSCC04]

A barrier is a **function** separating

- the unsafe region $\mathcal{X}_u$
- all trajectories starting form the initial region $\mathcal{X}_0$.



does not require computation of reachable set.

Conditions on Barrier Function

To be a valid barrier functions, [Prajna & Jadbabaie, HSCC04] shows that $B(\mathbf{x})$ has to satisfy

$$\begin{cases} B(\mathbf{x}) \le 0 & \forall \mathbf{x} \in \mathcal{X}_0 \\ B(\mathbf{x}) > 0 & \forall \mathbf{x} \in \mathcal{X}_u \\ B(\mathbf{x}) = 0 \Rightarrow \left\langle \frac{\partial B}{\partial x}(\mathbf{x}), f(\mathbf{x}, \mathbf{d}) \right\rangle < 0 & \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

Finding a barrier function is difficult in general

Parametric Barrier Function

In [Prajna & Jadbabaie HSCC04], parametric barrier functions $B(\mathbf{x}, \mathbf{p})$ are considered. They have to satisfy

$$\exists \mathbf{p} \in \mathcal{P} :$$

$$\begin{cases} B(\mathbf{x}, \mathbf{p}) \leq 0 & \forall \mathbf{x} \in \mathcal{X}_0 \\ B(\mathbf{x}, \mathbf{p}) > 0 & \forall \mathbf{x} \in \mathcal{X}_u \\ B(\mathbf{x}, \mathbf{p}) = 0 \Rightarrow \left\langle \frac{\partial B}{\partial x}(\mathbf{x}, \mathbf{p}), f(\mathbf{x}, \mathbf{d}) \right\rangle < 0 & \forall \mathbf{x} \in \mathcal{X} \end{cases}$$

Example

- $B_1(\mathbf{x}, \mathbf{p}) = p_0 x_0 + p_1 x_1 + p_2$
- $B_2(\mathbf{x}, \mathbf{p}) = p_0 \ln(x_0) + p_1 x_1 + p_2$

In [Prajna & Jadbabaie HSCC04] only polynomial dynamical systems and polynomial barrier functions are considered,
with the 3rd constraint relaxed into

$$\left(\left\langle \frac{\partial B}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{p}), f\left(\mathbf{x}, \mathbf{d}\right)\right) \right\rangle < 0 \right) \qquad \forall \mathbf{x} \in \mathcal{X}$$

Designing barriers via interval analysis

We assume that the sets $\mathcal{X}_0$ and $\mathcal{X}_\mathrm{u}$ are defined by

$$\mathcal{X}_0 = \{\mathbf{x} \in \mathcal{X} \mid g_0(\mathbf{x}) \leqslant 0\}$$
$$\mathcal{X}_u = \{\mathbf{x} \in \mathcal{X} \mid g_u(\mathbf{x}) \leqslant 0\}.$$

with $g_0 : \mathcal{X} \to \mathbb{R}$ and $g_u : \mathcal{X} \to \mathbb{R}$ two known functions

Quantified Constraint Satisfaction Problem approach

Theorem

If $\exists \mathbf{p} \in \mathcal{P}$ *such that* $\forall \mathbf{x} \in \mathcal{X}$, $\forall \mathbf{d} \in \mathcal{D}$

$$
\begin{aligned}
\xi\left(\mathbf{x}, \mathbf{p}, \mathbf{d}\right) = & \left(g_0(\mathbf{x}) > 0 \vee B(\mathbf{x}, \mathbf{p}) \leqslant 0\right) \\
& \wedge \left(g_u(\mathbf{x}) > 0 \vee B(\mathbf{x}, \mathbf{p}) > 0\right) \\
& \wedge \left(B(\mathbf{x}, \mathbf{p}) \neq 0 \vee \left\langle \frac{\partial B}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{p}), f(\mathbf{x}, \mathbf{d}) \right\rangle < 0 \right)
\end{aligned}
$$

then the dynamical system is safe

Note: this offers a convenient way to treat each constraint of the conjunction in the same way.

Starting point

Consider some function $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ and some box $[\mathbf{z}] \in \mathbb{IR}^k$.

CSC-FPS [Jaulin & Walter 1996] is designed to determine whether

$$\exists \mathbf{p} \in [\mathbf{p}]\,,\ \forall \mathbf{x} \in [\mathbf{x}]\,,\ g\left(\mathbf{x}, \mathbf{p}\right) \in [\mathbf{z}]$$

We consider this approach here.

CSC-FPS consists of:

- FPS (Feasible Point Searcher): explores parameter space $\mathcal{P} = [\mathbf{p}]$ to find some satisfying $\mathbf{p}$.
- CSC (Computable Sufficient Condition): checks whether $\mathbf{p}$ satisfies the constraint for all $\mathbf{x} \in \mathcal{X} = [\mathbf{x}]$

Additional contributions

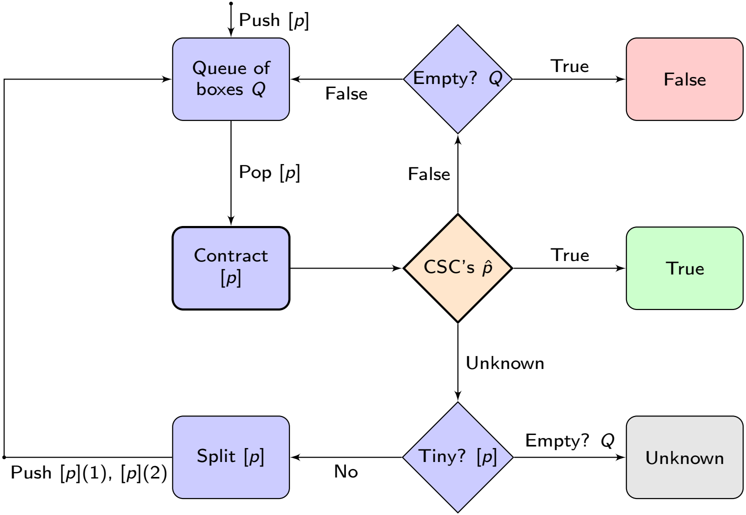
- **Adaptation:** Extending to handle conjunction of constraints, all are of the form

$$\tau\left(\mathbf{x},\mathbf{p},\mathbf{d}\right)=\left(u(\mathbf{x},\mathbf{p})\in\mathcal{A}\right)\vee\left(v(\mathbf{x},\mathbf{p},\mathbf{d})\in\mathcal{B}\right).$$

- **Improvements:** Enhance the algorithm by adding contractors operators

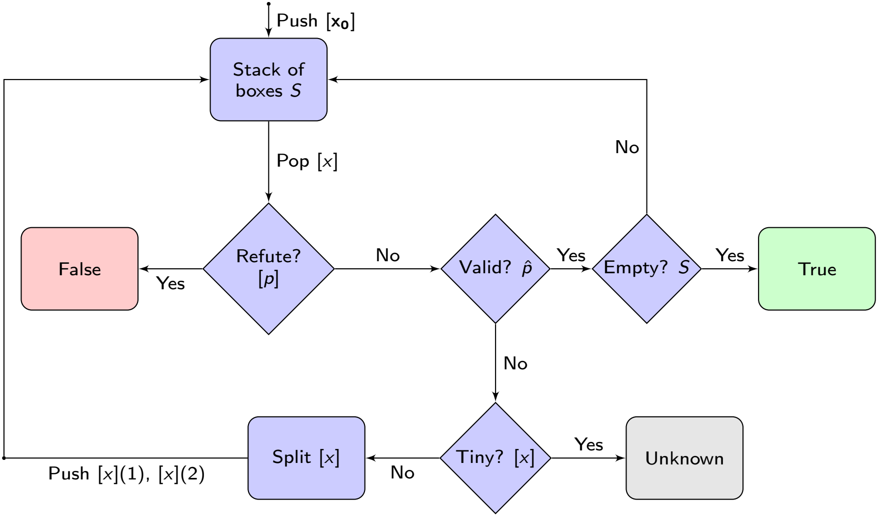
Algorithm: FPS

Input $[p]$ a box of parameter, f and a parametric function B



Algorithm: CSCInit case

Input $[p]$ a box of parameter and $[\mathbf{x}_0]$



Verification of a constraint – validation

For a given $\mathbf{p}$, validation of

$$\exists \mathbf{p} \in [\mathbf{p}], \forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{d} \in [\mathbf{d}], \quad (u(\mathbf{x},\mathbf{p}) \in \mathcal{A}) \vee (v(\mathbf{x},\mathbf{p},\mathbf{d}) \in \mathcal{B}).$$

One can outer-approximate for a given $\mathbf{p} \in [\mathbf{p}]$

$$u([\mathbf{x}], \mathbf{p}) = \{u(\mathbf{x},\mathbf{p}) \mid \mathbf{x} \in [\mathbf{x}]\}$$
$$v([\mathbf{x}], [\mathbf{d}], \mathbf{p}) = \{v(\mathbf{x},\mathbf{d},\mathbf{p}) \mid \mathbf{x} \in [\mathbf{x}], \mathbf{d} \in [\mathbf{d}]\}$$

using **inclusion functions** $[u]([\mathbf{x}], \mathbf{p})$ and $[v]([\mathbf{x}], [\mathbf{d}], \mathbf{p})$

Consequence

If $[u]([\mathbf{x}], \mathbf{p}) \subseteq \mathcal{A}$ or $[v]([\mathbf{x}], [\mathbf{d}], \mathbf{p}) \subseteq \mathcal{B}$ then

$u([\mathbf{x}], \mathbf{p}) \subseteq \mathcal{A}$ or $v([\mathbf{x}], [\mathbf{d}], \mathbf{p}) \subseteq \mathcal{B}$

and

$$\forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{d} \in [\mathbf{d}] \;\; u(\mathbf{x},\mathbf{p}) \in \mathcal{A} \vee u(\mathbf{x},\mathbf{d},\mathbf{p}) \in \mathcal{B}$$

is satisfied.

Verification of a constraint – refutation

- either using **Inclusion functions**, on the negation of the constraint

$$\forall \mathbf{p} \in [\mathbf{p}], \exists \mathbf{x} \in [\mathbf{x}], \exists \mathbf{d} \in [\mathbf{d}], \quad u(\mathbf{x}, \mathbf{p}) \subseteq \overline{\mathcal{A}} \wedge v(\mathbf{x}, \mathbf{p}, \mathbf{d}) \subseteq \overline{\mathcal{B}},$$

  for a given $\mathbf{x}$ and a given $\mathbf{d}$ (Note: we can try several random values)

- either using **Contractors**
  A contractor $\mathcal{C}_{g,[\mathbf{z}]}$ associated to $\{\mathbf{x} \in [\mathbf{x}] : g(\mathbf{x}) \in [\mathbf{z}]\}$ is s.t.
  - Reduction:
  $$\mathcal{C}_{g,[\mathbf{z}]}([\mathbf{x}]) \subseteq [\mathbf{x}]$$
  - Soundness:
  $$[g]([\mathbf{x}]) \cap [\mathbf{z}] = [g]\left(\mathcal{C}_{g,[\mathbf{z}]}([\mathbf{x}])\right) \cap [\mathbf{z}]$$

  They can be composed, with $c_i : \{\mathbf{x} \in [\mathbf{x}] : g_i(\mathbf{x}) \in [\mathbf{z}]_i\}$

$$\mathcal{C}_{c_1 \wedge c_2}([\mathbf{x}]) = \mathcal{C}_{c_1}([\mathbf{x}]) \cap \mathcal{C}_{c_2}([\mathbf{x}])$$
$$\mathcal{C}_{c_1 \wedge c_2}([\mathbf{x}]) = \mathcal{C}_{c_2}(\mathcal{C}_{c_1}([\mathbf{x}]))$$
$$\mathcal{C}_{c_1 \vee c_2}([\mathbf{x}]) = \square\{\mathcal{C}_{c_1}([\mathbf{x}]) \cup \mathcal{C}_{c_2}([\mathbf{x}])\}$$

Note: several contractor algorithms exist, *e.g.*, HC4Revise, 3BCID, etc.

Using contractors – 1

Proposition

Consider a box $[\mathbf{x}]$*, the constraint* $c : \{\mathbf{x} \in [\mathbf{x}] : g(\mathbf{x}) \in [\mathbf{z}]\}$*, and the contracted box* $\mathcal{C}_c\left([\mathbf{x}]\right) \subseteq [\mathbf{x}]$*. Then,*

$$\forall \mathbf{x} \in [\mathbf{x}] \setminus \mathcal{C}_c\left([\mathbf{x}]\right), \text{ one has } g(\mathbf{x}) \notin [\mathbf{z}], \tag{1}$$

where $[\mathbf{x}] \setminus \mathcal{C}_c\left([\mathbf{x}]\right)$ *denotes the box* $[\mathbf{x}]$ *deprived from* $\mathcal{C}_c\left([\mathbf{x}]\right)$*, which is not necessarily a box.*

Using contractors – 2

Consider the constraint

$$\tau : (u(\mathbf{x}, \mathbf{p}) \in \mathcal{A}) \vee (v(\mathbf{x}, \mathbf{p}, \mathbf{d}) \in \mathcal{B})$$

and a contractor $\mathcal{C}_\tau$ for this constraint.

For the boxes $[\mathbf{x}]$, $[\mathbf{p}]$, and $[\mathbf{d}]$, one gets

$$\left([\mathbf{x}]', \quad [\mathbf{p}]', \quad [\mathbf{d}]'\right) = \mathcal{C}_\tau\left([\mathbf{x}], \quad [\mathbf{p}], \quad [\mathbf{d}]\right)$$

We have different cases to consider in function of the values of the contracted boxes $[\mathbf{x}]'$, $[\mathbf{p}]'$, and $[\mathbf{d}]'$.

Note: we can do the same with $\overline{\tau}$

Using contractors – 3

3 cases are considered:

1. If $[\mathbf{p}] \setminus [\mathbf{p}]' \neq \emptyset$, then $\forall \mathbf{p} \in [\mathbf{p}] \setminus [\mathbf{p}]'$, $\forall \mathbf{x} \in [\mathbf{x}]$, $\forall \mathbf{d} \in [\mathbf{d}]$,

$$u(\mathbf{x}, \mathbf{p}) \notin \mathcal{A} \wedge v(\mathbf{x}, \mathbf{p}, \mathbf{d}) \notin \mathcal{B},$$

   $\Rightarrow$ the search space is reduced to $[\mathbf{p}]'$,

2. If $[\mathbf{x}] \setminus [\mathbf{x}]' \neq \emptyset$ then, one has $\forall \mathbf{p} \in [\mathbf{p}]$, $\forall \mathbf{x} \in [\mathbf{x}] \setminus [\mathbf{x}]'$, $\forall \mathbf{d} \in [\mathbf{d}]$,

$$u(\mathbf{x}, \mathbf{p}) \notin \mathcal{A} \wedge v(\mathbf{x}, \mathbf{p}, \mathbf{d}) \notin \mathcal{B}$$

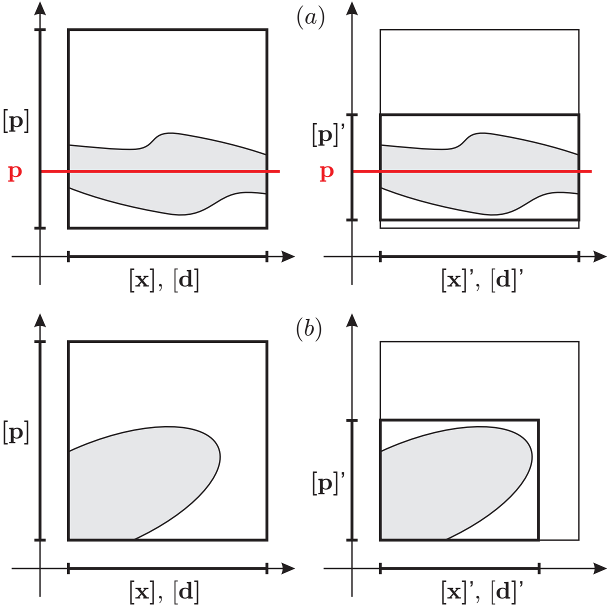
   and there is no $\mathbf{p} \in [\mathbf{p}]$ such that τ holds true for all $\mathbf{x} \in [\mathbf{x}]$,

3. If $[\mathbf{d}] \setminus [\mathbf{d}]' \neq \emptyset$, then $\forall \mathbf{p} \in [\mathbf{p}]$, $\forall \mathbf{x} \in [\mathbf{x}]$, $\forall \mathbf{d} \in [\mathbf{d}] \setminus [\mathbf{d}]'$,

$$u(\mathbf{x}, \mathbf{p}) \notin \mathcal{A} \wedge v(\mathbf{x}, \mathbf{p}, \mathbf{d}) \notin \mathcal{B}$$

   and there is no $\mathbf{p} \in [\mathbf{p}]$ such that τ holds true for all $\mathbf{d} \in [\mathbf{d}]$,

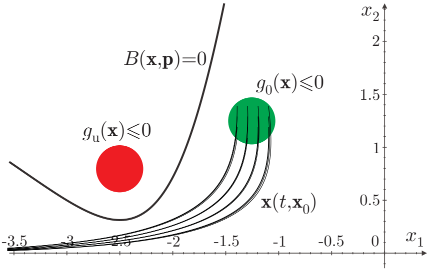
Using contractors – 3



Example: rational barrier function

Example

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_1 + x_2 \\ x_1 x_2 - 0.5 x_2^2 \end{pmatrix}$$

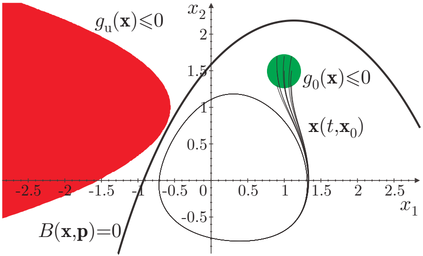


Parametric barrier function: $B(\mathbf{x}, \mathbf{p}) = \frac{p_1 p_2 (x_0 + p_3)}{(x_0 + p_3)^2 + p_2^2} + x_1 + p_4$

Example: system with limit cycle

Example

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{pmatrix} x_2 + (1 - x_1^2 - x_2^2)x_1 + \ln(x_1^2 + 1) \\ -x_1 + (1 - x_1^2 - x_2^2)x_2 + \ln(x_2^2 + 1) \end{pmatrix}$$



Parametric barrier function: $B(\mathbf{x}, \mathbf{p}) = \left(\frac{x_1 + p_1}{p_2}\right)^2 + \left(\frac{x_2 + p_3}{p_4}\right)^2 - 1$

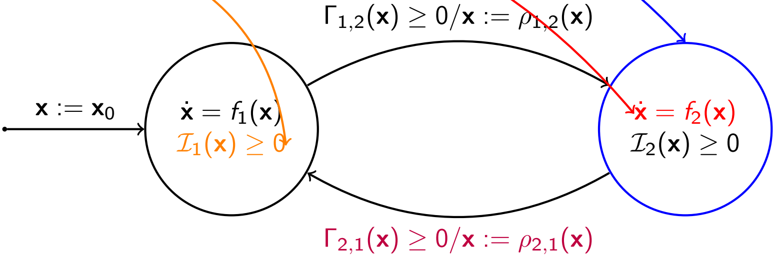
Results

			Without contr.		With contr.	
Example	n	m	time	bisect.	time	bisect.
1	2	4	36s	4520	16s	4553
2	2	3	T.O.	/	1s	159
3	2	6	1133s	20388	1s	6
4	2	6	253s	14733	7s	435
5	2	4	T.O.	/	98s	4072
6	3	4	167s	1753	21s	47
7	6	7	697s	67600	1s	261

- n the dimension of the dynamical system
- m the number of parameters of the template

Extension to hybrid automata

- **Location**
- **Flow**
- **Invariant**



- **Transtion** with **Guard** and **Reset**

Extension to Hybrid Automata

To be a valid barrier functions, [Prajna & Jadbabaie, HSCC04] shows that for an Hybrid Automaton $\mathcal{H} = (\mathcal{X}, \mathcal{L}, \mathcal{X}_0, \mathcal{I}, f, \Gamma, \rho)$

Theorem

Assume that there exist a family of differentiable functions $\beta_\ell(\mathbf{x})$, $\ell \in \mathcal{L}$ such that, for all pairs $(\ell, \ell') \in \mathcal{L}^2$ with $\ell \neq \ell'$, one has

$$
\begin{aligned}
\beta_\ell(\mathbf{x}) &\leqslant 0 \quad \forall \mathbf{x} \in \mathcal{X}_0(\ell) \\
\beta_\ell(\mathbf{x}) &> 0 \quad \forall \mathbf{x} \in \mathcal{X}_u(\ell) \\
\beta_\ell(\mathbf{x}) = 0 \implies \frac{\partial \beta_\ell(\mathbf{x})}{\partial \mathbf{x}} f_\ell(\mathbf{x}, \mathbf{d}) &< 0 \quad \forall \mathbf{x} \in \mathcal{I}(\ell), \forall \mathbf{d} \in \mathcal{D}_\ell \\
\beta_\ell(\mathbf{x}) \leqslant 0 \implies \beta_{\ell'}(\rho_{\ell,\ell'}(\mathbf{x})) &\leqslant 0 \quad \forall \mathbf{x} \in \Gamma(\ell, \ell')
\end{aligned}
$$

then the system $\mathcal{H}$ is safe.

Challenge: increasing number of constraints associated to the number of transitions

Interval analysis approach

Assume that there exists for each location $\ell \in \mathcal{L}$ some functions

- $g_0 : \mathcal{L} \times \mathcal{X} \to \mathbb{R}$,
- $g_u : \mathcal{L} \times \mathcal{X} \to \mathbb{R}$,
- $g_\Gamma : \mathcal{L} \times \mathcal{L} \times \mathcal{X} \to \mathbb{R}$
- $g_\mathcal{I} : \mathcal{L} \times \mathcal{X} \to \mathbb{R}$

such that

- $\mathcal{X}_0(\ell) = \{\mathbf{x} \in \mathcal{X} \mid g_0(\ell, \mathbf{x}) \leqslant 0\}$,
- $\mathcal{X}_u(\ell) = \{\mathbf{x} \in \mathcal{X} \mid g_u(\ell, \mathbf{x}) \leqslant 0\}$,
- $\Gamma(\ell, \ell') = \{\mathbf{x} \in \mathcal{X} \mid g_\Gamma(\ell, \ell', \mathbf{x}) \leqslant 0\}$
- $\mathcal{I}(\ell) = \{\mathbf{x} \in \mathcal{X} \mid g_\mathcal{I}(\ell, \mathbf{x}) \leqslant 0\}$.

Quanfitied Constraint Satisfaction Problem

Proposition

Consider a hybrid system described by $\mathcal{H} = (\mathcal{X}, \mathcal{L}, \mathcal{X}_0, \mathcal{I}, f, \Gamma, \rho)$*. Assume there exists a differentiable function* $\beta_\ell(\mathbf{x}, \mathbf{p})$ *which satisfies* $\forall \ell \in \mathcal{L},\ \exists \mathbf{p}_\ell \in [\mathbf{p}]_\ell, \forall \mathbf{x} \in [\mathbf{x}], \forall \mathbf{d} \in [\mathbf{d}]_\ell$

$$g_0(\ell, \mathbf{x}) > 0 \vee \beta_\ell(\mathbf{x}, \mathbf{p}_\ell) \leqslant 0, \tag{2}$$

$$g_u(\ell, \mathbf{x}) > 0 \vee \beta_\ell(\mathbf{x}, \mathbf{p}_\ell) > 0, \tag{3}$$

$$g_{\mathcal{I}}(\ell, \mathbf{x}) > 0 \vee \beta_\ell(\mathbf{x}, \mathbf{p}_\ell) \neq 0 \vee \frac{\partial \beta_\ell(\mathbf{x}, \mathbf{p}_\ell)}{\partial x} f_\ell(\mathbf{x}, \mathbf{d}) < 0, \tag{4}$$

and $\forall \ell' \in \mathcal{L}$*, with* $\ell' \neq \ell$*,*

$$g_\Gamma(\ell, \ell', \mathbf{x}) > 0 \vee \beta_\ell(\mathbf{x}, \mathbf{p}_\ell) > 0 \vee \beta_{\ell'}(\rho_{\ell,\ell'}(\mathbf{x}), \mathbf{p}_{\ell'}) \leqslant 0, \tag{5}$$

then the system $\mathcal{H}$ *is safe.*

Contribution generalization of the formalism used for continuous-time dynamical systems.

Incremental solution of QCSP

Many ways to solve this QCSP

Our approach, incremental algorithm. Main ideas:

- We add an order on the location value ranging from 1 to $|\mathcal{L}|$
- We associate a parametric barrier function to each location
- We consider all the location from 1 to $|\mathcal{L}|$
  - For location $\ell = 1$ we try to find $\mathbf{p}$ such that we have a barrier function
  - For location $\ell > 1$, we try to find $\mathbf{p}$ taking into account all the constraints associated to the transition involving ℓ
  - In case we cannot find $\mathbf{p}$ we go back to location $\ell - 1$ and retry

Results

Example	Dimension	#locations	computation time	#bisections
2-TANKS	2	2	1.7s	9488
ECO	2	2	0.082s	499
prajna	3	2	0.2s	111
CAR	6	3	0.021s	3
Collision	3	6	0.334s	1574

Conclusion and Future Work

Conclusion

- A new method to compute barrier certificate based on interval analysis.
- Can handle non-linear dynamic and non-linear barrier functions.
- **Extension of the state-of-the art** which is limited to:
  - polynomial dynamic and polynomial barrier function

Future work

- Automatically find template, see paper Goubault et al. ACC'14
- Adapt other work of Prajna et al. on reachability

Publications/Submissions

- a paper accepted at CDC'14
- a paper under review in Automatica (round 2)
- a paper in prepration