Controller synthesis for linear systems and safe linear-time temporal logic

Matthias Rungger[1], Manuel Mazo[2], and Paulo Tabuada[1]

[1]Cyber-Physical Systems Laboratory
Department of Electrical Engineering
University of California at Los Angeles

[2]Deft Center for Systems and Control
TUDelft

Motivation

- Linear discrete-time system:

$$\Sigma : \quad x^{+} = Ax + Bu$$

with state space $\mathbb{R}^n$ and input space $\mathbb{R}^m$.

Motivation

- Linear discrete-time system:

$$\Sigma : \quad x^{+} = Ax + Bu$$

  with state space $\mathbb{R}^n$ and input space $\mathbb{R}^m$.

- Safe linear temporal logic specification (LTL) φ over atomic propositions:

$$\mathcal{P} = \{p_1, \ldots, p_\ell\}$$

  with each $p_i \in \mathcal{P}$ a polytope in $\mathbb{R}^n$.

Motivation

- Linear discrete-time system:

$$\Sigma : \quad x^{+} = Ax + Bu$$

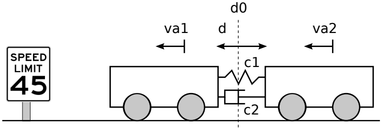
  with state space $\mathbb{R}^n$ and input space $\mathbb{R}^m$.
- Safe linear temporal logic specification (LTL) φ over atomic propositions:

$$\mathcal{P} = \{p_1, \ldots, p_\ell\}$$

  with each $p_i \in \mathcal{P}$ a polytope in $\mathbb{R}^n$.
- Expressive specifications:
  - sequencing of actions, “if then else” requirements, fault recovery, ...
  - guaranteed to only define safety properties;
  - negation is only allowed on atomic propositions, *until* is replaced with *wait*.

- Dynamics:

$$\dot{x} = \begin{bmatrix} 0 & -1 & 1 \\ \frac{k_s}{m} & -\frac{k_d}{m} & \frac{k_d}{m} \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

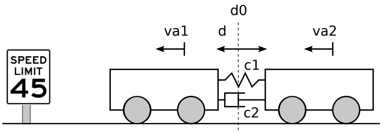


with $x = (d, v_1, v_2) \in \mathbb{R}^3$ and $u \in \mathbb{R}$.

  - d distance between the truck and the trailer
  - v_1 velocity of the truck
  - v_2 velocity of the trailer
  - u acceleration

- Dynamics:

$$\dot{x} = \begin{bmatrix} 0 & -1 & 1 \\ \frac{k_s}{m} & -\frac{k_d}{m} & \frac{k_d}{m} \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$



with $x = (d, v_1, v_2) \in \mathbb{R}^3$ and $u \in \mathbb{R}$.

  - d distance between the truck and the trailer
  - v_1 velocity of the truck
  - v_2 velocity of the trailer
  - u acceleration

- Specification:
  - compliance with speed limits v_a, v_b after at most $T \in \mathbb{N}$ time-steps
  - acceleration constraints $u \in [\underline{u}, \overline{u}]$
  - distance constraints $d \in [\underline{d}, \overline{d}]$

Cruise Control Example

Compliance with Speed Limits

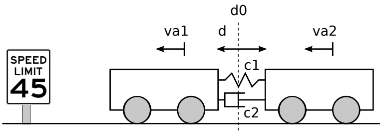
- Dynamics:

$$\dot{x} = \begin{bmatrix} 0 & -1 & 1 \\ \frac{k_s}{m} & -\frac{k_d}{m} & \frac{k_d}{m} \\ 0 & 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

with $x = (d, v_1, v_2) \in \mathbb{R}^3$ and $u \in \mathbb{R}$.

  - d distance between the truck and the trailer
  - v_1 velocity of the truck
  - v_2 velocity of the trailer
  - u acceleration

- Specification:
  - compliance with speed limits v_a, v_b after at most $T \in \mathbb{N}$ time-steps
  - acceleration constraints $u \in [\underline{u}, \overline{u}]$
  - distance constraints $d \in [\underline{d}, \overline{d}]$



- Safe LTL formula

$$\Box(D \wedge U \wedge \varphi_a \wedge \varphi_b)$$

with φ_a and φ_b given by

$$\begin{aligned} m_a &\implies \Diamond_{\leq T}(t_a \,\mathrm{W}\, m_b) \\ m_b &\implies \Diamond_{\leq T}(t_b \,\mathrm{W}\, m_a) \end{aligned}$$

  - m_i: v_i is active $i \in \{a, b\}$
  - t_i: $v_1 \leq v_i$

Reducing Controller Synthesis to a Safety Game

[Kupferman and Vardi, 2001]

- Construct the bad-prefix automaton $A_{\neg\varphi}$ from the safe LTL formula φ:

$$A_{\neg\varphi} = (Q, F, \delta, g, 2^{\mathcal{P}});$$

Reducing Controller Synthesis to a Safety Game

[Kupferman and Vardi, 2001]

- Construct the bad-prefix automaton $A_{\neg\varphi}$ from the safe LTL formula φ:

$$A_{\neg\varphi} = (Q, F, \delta, g, 2^{\mathcal{P}});$$

- Compose $A_{\neg\varphi}$ with the control system Σ:

$$S = A_{\neg\varphi} \| \Sigma;$$

Reducing Controller Synthesis to a Safety Game

[Kupferman and Vardi, 2001]

- Construct the bad-prefix automaton $A_{\neg\varphi}$ from the safe LTL formula φ:

$$A_{\neg\varphi} = (Q, F, \delta, g, 2^{\mathcal{P}});$$

- Compose $A_{\neg\varphi}$ with the control system Σ:

$$S = A_{\neg\varphi} \| \Sigma;$$

- Given the safe set $K = (Q \backslash F) \times H \subseteq (Q \backslash F) \times \mathbb{R}^n$ compute its largest controlled invariant subset:

$$\mathcal{K}(K) = \left\{(q, x) \in Q \times \mathbb{R}^n \mid \exists\, u \in \mathbb{R}^m, \text{Post}_u(q, x) \subseteq K\right\}.$$

Reducing Controller Synthesis to a Safety Game

[Kupferman and Vardi, 2001]

- Construct the bad-prefix automaton $A_{\neg\varphi}$ from the safe LTL formula φ:

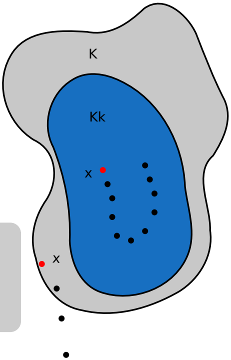
$$A_{\neg\varphi} = (Q, F, \delta, g, 2^{\mathcal{P}});$$

- Compose $A_{\neg\varphi}$ with the control system Σ:

$$S = A_{\neg\varphi} \| \Sigma;$$

- Given the safe set $K = (Q \backslash F) \times H \subseteq (Q \backslash F) \times \mathbb{R}^n$ compute its largest controlled invariant subset:

$$\mathcal{K}(K) = \left\{ (q, x) \in Q \times \mathbb{R}^n \mid \exists\, u \in \mathbb{R}^m, \ \mathrm{Post}_u(q, x) \subseteq K \right\}.$$



Reducing Controller Synthesis to a Safety Game

[Kupferman and Vardi, 2001]

- Construct the bad-prefix automaton $A_{\neg\varphi}$ from the safe LTL formula φ:

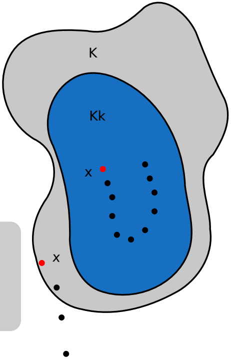
$$A_{\neg\varphi} = (Q, F, \delta, g, 2^{\mathcal{P}});$$

- Compose $A_{\neg\varphi}$ with the control system Σ:

$$S = A_{\neg\varphi} \| \Sigma;$$

- Given the safe set $K = (Q \backslash F) \times H \subseteq (Q \backslash F) \times \mathbb{R}^n$ compute its largest controlled invariant subset:

$$\mathcal{K}(K) = \left\{(q, x) \in Q \times \mathbb{R}^n \mid \exists\, u \in \mathbb{R}^m,\ \mathrm{Post}_u(q, x) \subseteq K\right\}.$$



- We know that:

$(q, x) \in \mathcal{K}(K) \Leftrightarrow$ existence of a control strategy enforcing φ from x.

Computation of Controlled Invariant Subsets

Basic Algorithm

- Fixed point computation [Bertsekas, 1972]:

$$K_{j+1} = \mathrm{pre}(K_j) \cap K_j, \quad K_0 = K$$

with $\mathrm{pre}(K_j)$ being the set of states $(q, x) \in Q \times \mathbb{R}^n$ for which there exists an input $u \in \mathbb{R}^m$ forcing a transition to some state in K_j.

- Fixed point computation [Bertsekas, 1972]:

$$K_{j+1} = \mathrm{pre}(K_j) \cap K_j, \quad K_0 = K$$

with $\mathrm{pre}(K_j)$ being the set of states $(q, x) \in Q \times \mathbb{R}^n$ for which there exists an input $u \in \mathbb{R}^m$ forcing a transition to some state in K_j.

- Safe set $K = (Q \backslash F) \times H \subseteq (Q \backslash F) \times \mathbb{R}^n$ given by:

$$H = \bigcup_{i=1}^{p} H_i, \quad \text{each } H_i \text{ is a polytope}$$

$\Rightarrow$ each K_j is computable and the iteration is known to asymptotically converge:

$$\mathcal{K}(K) = \lim_{j \to \infty} K_j.$$

- No termination guarantees;

Computation of Controlled Invariant Subsets

Several Problems

- No termination guarantees;
- Set iterates K_j are not controlled invariant;

Computation of Controlled Invariant Subsets

Several Problems

- No termination guarantees;
- Set iterates K_j are not controlled invariant;
- Solutions for $K = (Q \backslash F) \times H$, if H is convex :
  - [De Santis et al., 2004]: iteration is initialized with a controlled invariant set $K_0 \subset K$;
  - [Blanchini and Miani, 2008]: modified iteration using contractive sets;
  - Several other methods based on approximations of K;

- No termination guarantees;
- Set iterates K_j are not controlled invariant;
- Solutions for $K = (Q \backslash F) \times H$, if H is convex :
  - [De Santis et al., 2004]: iteration is initialized with a controlled invariant set $K_0 \subset K$;
  - [Blanchini and Miani, 2008]: modified iteration using contractive sets;
  - Several other methods based on approximations of K;
- For H given as union of polytopes, the iterative computation introduces combinatorial complexity!

Computation of Controlled Invariant Subsets

Several Problems

- No termination guarantees;
- Set iterates K_j are not controlled invariant;
- Solutions for $K = (Q \backslash F) \times H$, if H is convex :
  - [De Santis et al., 2004]: iteration is initialized with a controlled invariant set $K_0 \subset K$;
  - [Blanchini and Miani, 2008]: modified iteration using contractive sets;
  - Several other methods based on approximations of K;
- For H given as union of polytopes, the iterative computation introduces combinatorial complexity!

In this work we approximate K by sets adapted to the dynamics.
(Finite termination and symbolic implementation)

Adapted Sets

- Any *controllable* linear system can be transformed to the special Brunovsky normal form by an invertible linear change of coordinates and feedback:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix};$$

Adapted Sets

- Any *controllable* linear system can be transformed to the special Brunovsky normal form by an invertible linear change of coordinates and feedback:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix};$$

- We can *under-approximate* the safe set $K = (Q \backslash F) \times H$ by $\check{K} = (Q \backslash F) \times \check{H}$:

$$\check{H} = \bigcup_{i=1}^{p} B_i \subseteq H$$

with each box B_i defined by $B_i = [a_1^i,\ b_1^i] \times \ldots \times [a_n^i, b_n^i]$.

Theorem (Finite termination)

Consider the composition $A_{\neg\varphi}\|\Sigma$ where Σ is in special Brunovsky normal form and $K = (Q\backslash F) \times H$ with H being a finite union of boxes. Then the largest controlled invariant subset of $\check{K}$ can be computed in finitely many steps.

Main Results

Termination

Theorem (Finite termination)

Consider the composition $A_{\neg\varphi} \| \Sigma$ where Σ is in special Brunovsky normal form and $K = (Q \backslash F) \times H$ with H being a finite union of boxes. Then the largest controlled invariant subset of $\check{K}$ can be computed in finitely many steps.

This result was first proved in [Tabuada and Pappas, 2003] and was used in [Tabuada and Pappas, 2006] to show, for the first time, that controllers can be synthesized to enforce LTL properties on control systems.

Main Results

Termination

Theorem (Finite termination)

Consider the composition $A_{\neg\varphi} \| \Sigma$ where Σ is in special Brunovsky normal form and $K = (Q \backslash F) \times H$ with H being a finite union of boxes. Then the largest controlled invariant subset of $\check{K}$ can be computed in finitely many steps.

This result was first proved in [Tabuada and Pappas, 2003] and was used in [Tabuada and Pappas, 2006] to show, for the first time, that controllers can be synthesized to enforce LTL properties on control systems.

How can we make use of this result when H is not a union of boxes?

We can under-approximate H by a finite union of boxes $\check{H}$.

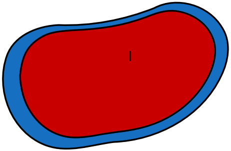
Main Results

Completeness

We can under-approximate H by a finite union of boxes $\check{H}$.

We say that a set $I \subseteq \mathbb{R}^n$ is strictly inside a set $J \subseteq \mathbb{R}^n$ if there exists $\gamma > 0$ for which:

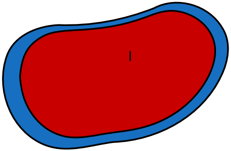
$$I + \gamma\mathcal{B}_\gamma(0) \subseteq J.$$



We can under-approximate H by a finite union of boxes $\check{H}$.

We say that a set $I \subseteq \mathbb{R}^n$ is strictly inside a set $J \subseteq \mathbb{R}^n$ if there exists $\gamma > 0$ for which:

$$I + \gamma \mathcal{B}_\gamma(0) \subseteq J.$$



Theorem (Completeness)

If there exists a controlled invariant set $I \subseteq \mathcal{K}(K)$ for which I_q is strictly inside $\mathcal{K}_q(K)$, then there exists an under-approximation $\check{K} = (Q \backslash F) \times \check{H}$ of K, with $\check{H}$ being a finite union of boxes, such that $I \subseteq \mathcal{K}(\check{K})$.

- Use binary decision diagrams (BDDs) to implement the iteration:

$$K_{j+1} = \mathrm{pre}(K_j) \cap K_j.$$

- Use binary decision diagrams (BDDs) to implement the iteration:

$$K_{j+1} = \mathrm{pre}(K_j) \cap K_j.$$

  - Each set K_j is encoded by a BDD;

- Use binary decision diagrams (BDDs) to implement the iteration:

$$K_{j+1} = \text{pre}(K_j) \cap K_j.$$

  - Each set K_j is encoded by a BDD;
  - The combination of the special Brunovsky normal form with adapted sets results in a simple expression for $\text{pre}(B_i)$ with $B_i = [a_1^i, b_1^i] \times \ldots \times [a_n^i, b_n^i]$:

$$\text{pre}(B_i) = \mathbb{R} \times [a_1^i, b_1^i] \times \ldots \times [a_{n-1}^i, b_{n-1}^i];$$

- Use binary decision diagrams (BDDs) to implement the iteration:

$$K_{j+1} = \mathrm{pre}(K_j) \cap K_j.$$

  - Each set K_j is encoded by a BDD;
  - The combination of the special Brunovsky normal form with adapted sets results in a simple expression for $\mathrm{pre}(B_i)$ with $B_i = [a_1^i, b_1^i] \times \ldots \times [a_n^i, b_n^i]$:

$$\mathrm{pre}(B_i) = \mathbb{R} \times [a_1^i, b_1^i] \times \ldots \times [a_{n-1}^i, b_{n-1}^i];$$

  - Symbolical computation of $\mathrm{pre}(K_j)$ can be done by shifting and variable reordering.

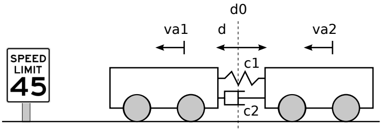
Cruise Control Example

Computational Results

Problem description:

- Σ : 3 states, 1 input;
- Safe LTL formula:

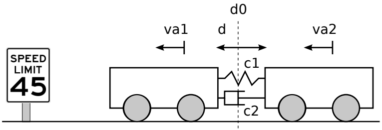
$$\Box(D \wedge U \wedge \varphi_a \wedge \varphi_b \wedge \varphi_c)$$



Problem description:

- Σ : 3 states, 1 input;
- Safe LTL formula:

$$\Box(D \wedge U \wedge \varphi_a \wedge \varphi_b \wedge \varphi_c)$$



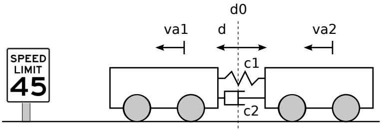
Parameters:

- $T \in \{2, 10\}$ number of time steps after which speed limit is enforced;
- $N \in \{10, \ldots, 13\}$ number of bits (2^N boxes) used in each dimension.

Problem description:

- Σ : 3 states, 1 input;
- Safe LTL formula:

$$\square(D \wedge U \wedge \varphi_a \wedge \varphi_b \wedge \varphi_c)$$



Parameters:

- $T \in \{2, 10\}$ number of time steps after which speed limit is enforced;
- $N \in \{10, \ldots, 13\}$ number of bits (2^N boxes) used in each dimension.

Error bound:

$$\hat{e} = \frac{\operatorname{vol} \mathcal{K}(\hat{K}) - \operatorname{vol} \mathcal{K}(\check{K})}{\operatorname{vol} \mathcal{K}(\check{K})} \geq \frac{\operatorname{vol} \mathcal{K}(K) - \operatorname{vol} \mathcal{K}(\check{K})}{\operatorname{vol} \mathcal{K}(K)}$$

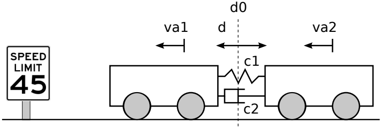
Cruise Control Example

Computational Results

Problem description:

- Σ : 3 states, 1 input;
- Safe LTL formula:

$$\Box(D \wedge U \wedge \varphi_a \wedge \varphi_b \wedge \varphi_c)$$



Parameters:

- $T \in \{2, 10\}$ number of time steps after which speed limit is enforced;
- $N \in \{10, \ldots, 13\}$ number of bits (2^N boxes) used in each dimension.

N\T	2		10	
	t_r	$\hat{e}$	t_r	$\hat{e}$
10	1m39s	2.31	2m40s	2.38
11	4m09s	1.01	4m31s	1.04
12	6m48s	0.58	7m52s	0.62
13	10m38s	0.43	16m01s	0.46

Error bound:

$$\hat{e} = \frac{\mathrm{vol}\,\mathcal{K}(\hat{K}) - \mathrm{vol}\,\mathcal{K}(\check{K})}{\mathrm{vol}\,\mathcal{K}(\check{K})} \geq \frac{\mathrm{vol}\,\mathcal{K}(K) - \mathrm{vol}\,\mathcal{K}(\check{K})}{\mathrm{vol}\,\mathcal{K}(K)}$$

Comparison with the Polyhedral Approach

- Example 5.1 in [Pérez et al., 2011]: 3 states + 2 inputs
- Workspace: $X = [0, 30]^3$ and $U = [0, 2]^2$
- Obstacles in the state space:
  - $O_1 = [-5, 15]^3$
  - $O_2 = [-5, 5]^3$
  - $O_3 = [-15, 10]^3$
- Obstacles in the input space:
  - $V_1 = [-3/2, 1/2]^2$
  - $V_2 = [-1/4, 1/4]^2$
  - $V_3 = [2/5, 1/5]^2$

- Specification with increasing complexity:

$$
\begin{aligned}
\varphi_0 &= \Box(X \times U) \\
\varphi_1 &= \Box((X \wedge \neg O_1) \times U) \\
\varphi_2 &= \Box(X \times (U \wedge \neg V_1)) \\
\varphi_3 &= \Box((X \wedge \neg O_1) \times (U \wedge \neg V_1)) \\
\varphi_4 &= \Box((X \wedge_{i=1}^{2} \neg O_i) \times (U \wedge_{i=1}^{2} \neg V_i)) \\
\varphi_5 &= \Box((X \wedge_{i=1}^{3} \neg O_i) \times (U \wedge_{i=1}^{2} \neg V_i)) \\
\varphi_6 &= \Box((X \wedge_{i=1}^{3} \neg O_i) \times (U \wedge_{i=1}^{3} \neg V_i))
\end{aligned}
$$

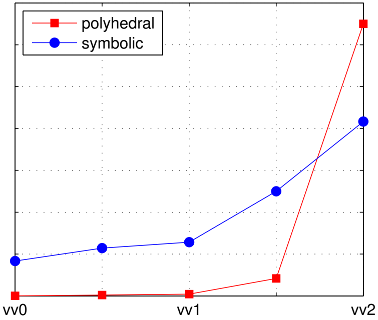
Comparison with the Polyhedral Approach

- Example 5.1 in [Pérez et al., 2011]: 3 states + 2 inputs
- Workspace: $X = [0, 30]^3$ and $U = [0, 2]^2$
- Obstacles in the state space:
  - $O_1 = [-5, 15]^3$
  - $O_2 = [-5, 5]^3$
  - $O_3 = [-15, 10]^3$
- Obstacles in the input space:
  - $V_1 = [-3/2, 1/2]^2$
  - $V_2 = [-1/4, 1/4]^2$
  - $V_3 = [2/5, 1/5]^2$

- Specification with increasing complexity:

$$\begin{aligned}\varphi_0 &= \Box(X \times U)\\ \varphi_1 &= \Box((X \wedge \neg O_1) \times U)\\ \varphi_2 &= \Box(X \times (U \wedge \neg V_1))\end{aligned}$$

- Computation times:



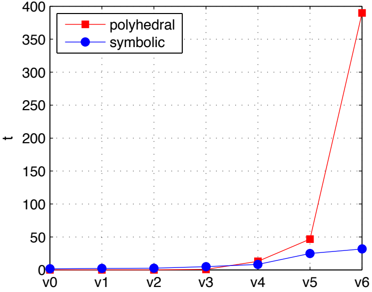
Comparison with the Polyhedral Approach

- Example 5.1 in [Pérez et al., 2011]: 3 states + 2 inputs
- Workspace: $X = [0, 30]^3$ and $U = [0, 2]^2$
- Obstacles in the state space:
  - $O_1 = [-5, 15]^3$
  - $O_2 = [-5, 5]^3$
  - $O_3 = [-15, 10]^3$
- Obstacles in the input space:
  - $V_1 = [-3/2, 1/2]^2$
  - $V_2 = [-1/4, 1/4]^2$
  - $V_3 = [2/5, 1/5]^2$

- Specification with increasing complexity:

$$\begin{aligned} \varphi_0 &= \Box(X \times U) \\ \varphi_1 &= \Box((X \wedge \neg O_1) \times U) \\ \varphi_2 &= \Box(X \times (U \wedge \neg V_1)) \end{aligned}$$

- Computation times:



Summary and Remarks

Algorithm to synthesize controllers enforcing safe LTL specification on controllable linear systems:

- Termination guarantees;
- Best possible completeness guarantees;
- Full symbolic implementation;
- Five continuous variables (state of the art).

Summary and Remarks

Algorithm to synthesize controllers enforcing safe LTL specification on controllable linear systems:

- Termination guarantees;
- Best possible completeness guarantees;
- Full symbolic implementation;
- Five continuous variables (state of the art).

What is next?

- Boxes are not good enough
  (too many required to obtain reasonable approximations, CoD);
- Find more general polyhedra for which termination is guaranteed.

Literature

Bertsekas, D. (1972).
Infinite time reachability of state-space regions by using feedback control.
IEEE TAC, 17:604–613.

Blanchini, F. and Miani, S. (2008).
Set-Theoretic Methods in Control.
Systems & Control: Foundations & Applications. Birkhäuser.

De Santis, E., Di Benedetto, M. D., and Berardi, L. (2004).
Computation of maximal safe sets for switching systems.
IEEE TAC, 49:184–195.

Kupferman, O. and Vardi, M. Y. (2001).
Model checking of safety properties.
Formal Methods in System Design, 19:291–314.

Pérez, E., Ariño, C., Blasco, F. X., and Martínez, M. A. (2011).
Maximal closed loop admissible set for linear systems with non-convex polyhedral constraints.
Journal of Process Control, pages 529 – 537.

Tabuada, P. and Pappas, G. J. (2003).
Model checking LTL over controllable linear systems is decidable.
In *HSCC*, LNCS, pages 498–513. Springer.

Tabuada, P. and Pappas, G. J. (2006).
Linear time logic control of discrete-time linear systems.
IEEE TAC, 51:1862–1877.