POLITECNICO
MILANO 1863

Dipartimento di Elettronica, Informazione e Bioingegneria

Reachability in Cyber-Physical Systems

Maria Prandini
maria.prandini@polimi.it

12th International Conference on Reachability Problems
September 24-26, 2018 – Marseille, France

Cyber-physical systems

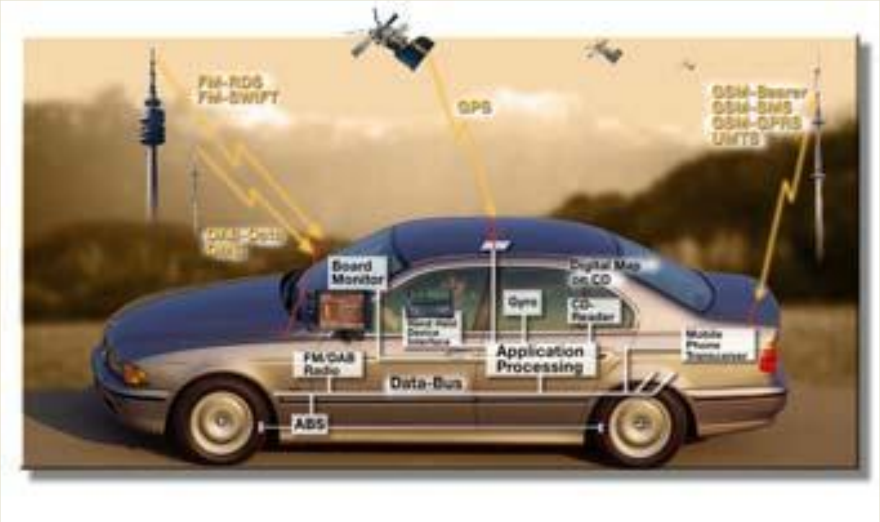
A cyber-physical system is an engineering system where

communication, computation, and control – the cyber part

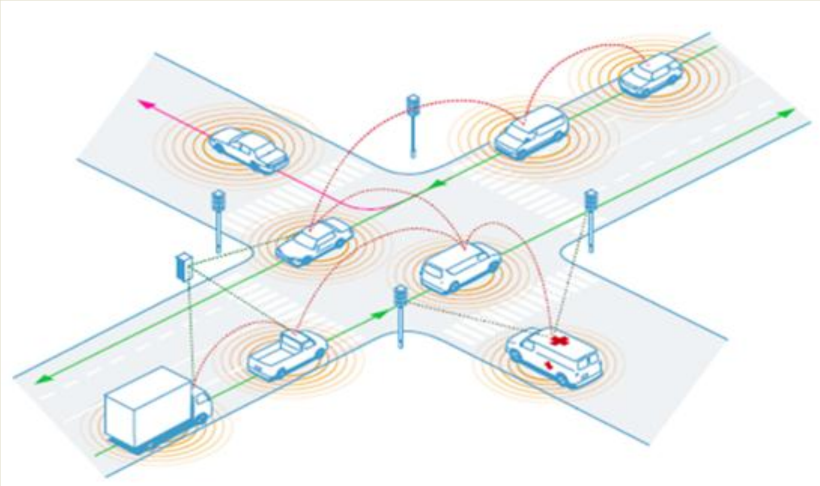
are integrated within

natural and/or human-made systems – the physical part

Cyber-physical systems



Cyber-physical systems of systems



Cyber-physical systems

Cyber – computation, communication, and control

discrete, logical, and switched

Physical – natural and/or human-made systems

continuous variables evolving according to the laws of physics

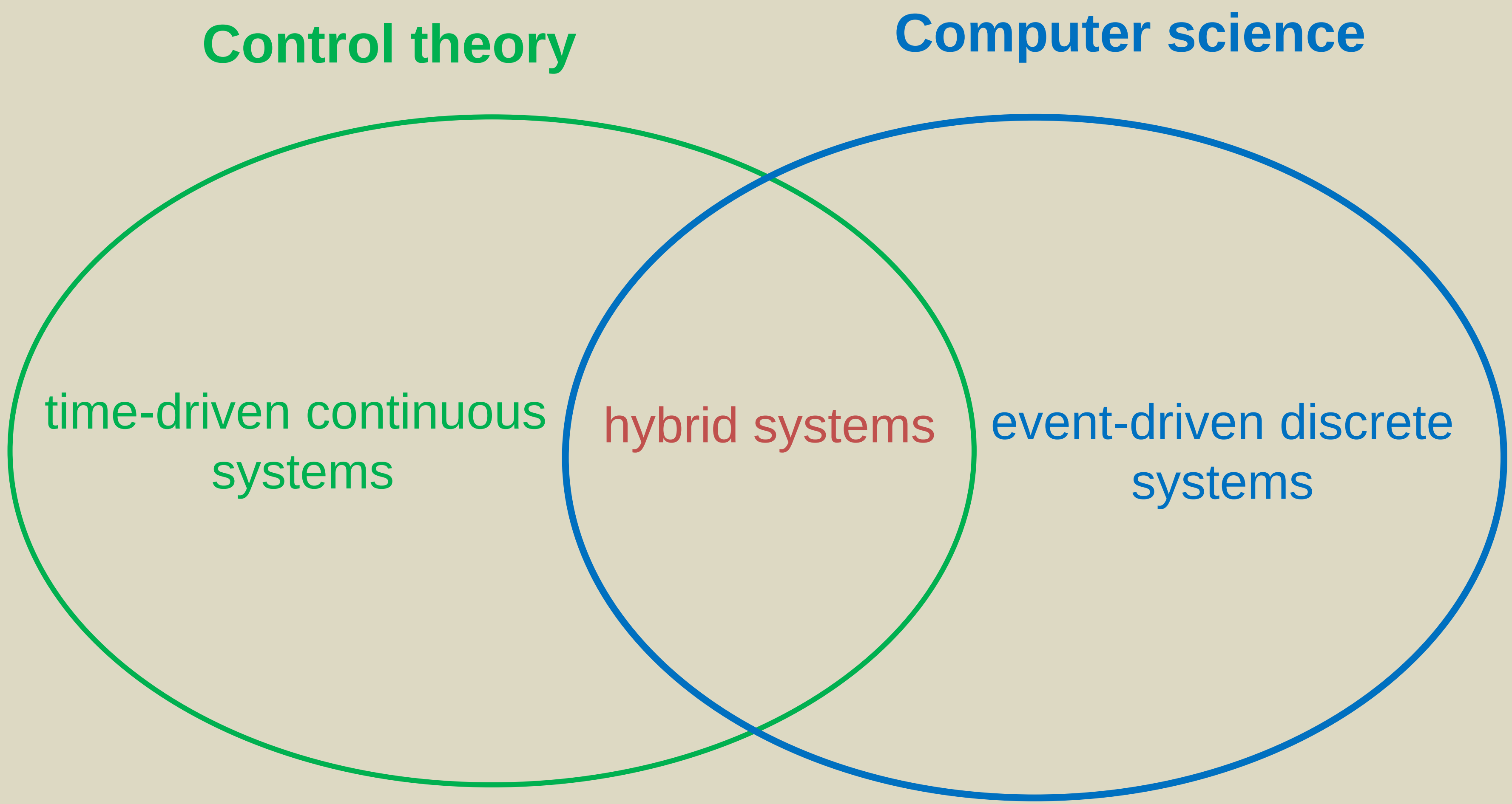
Hybrid systems

characterized by interleaved continuous and discrete dynamics combining

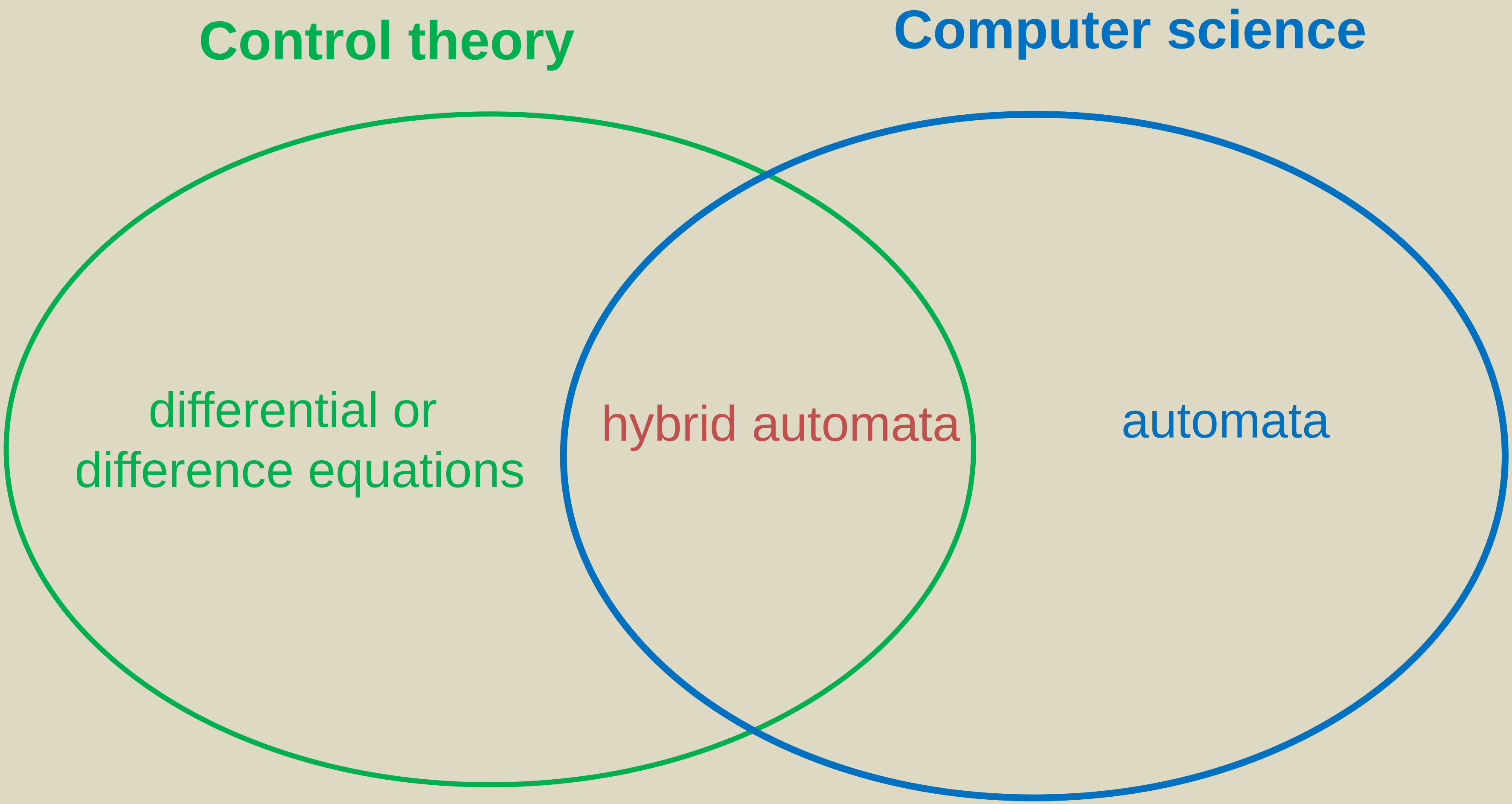
- time-driven continuous systems
  state takes values in a continuous set and changes as time progresses

- event-driven discrete systems
  state takes values in a discrete set and changes due to the occurrence of an event

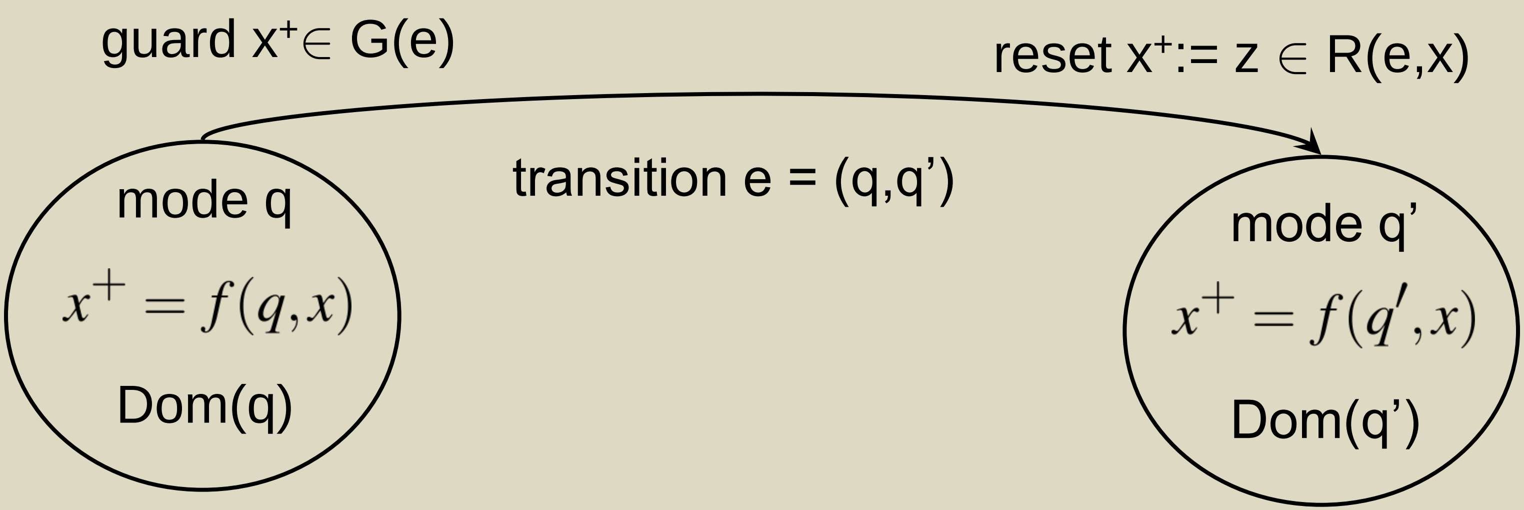
Hybrid systems theory



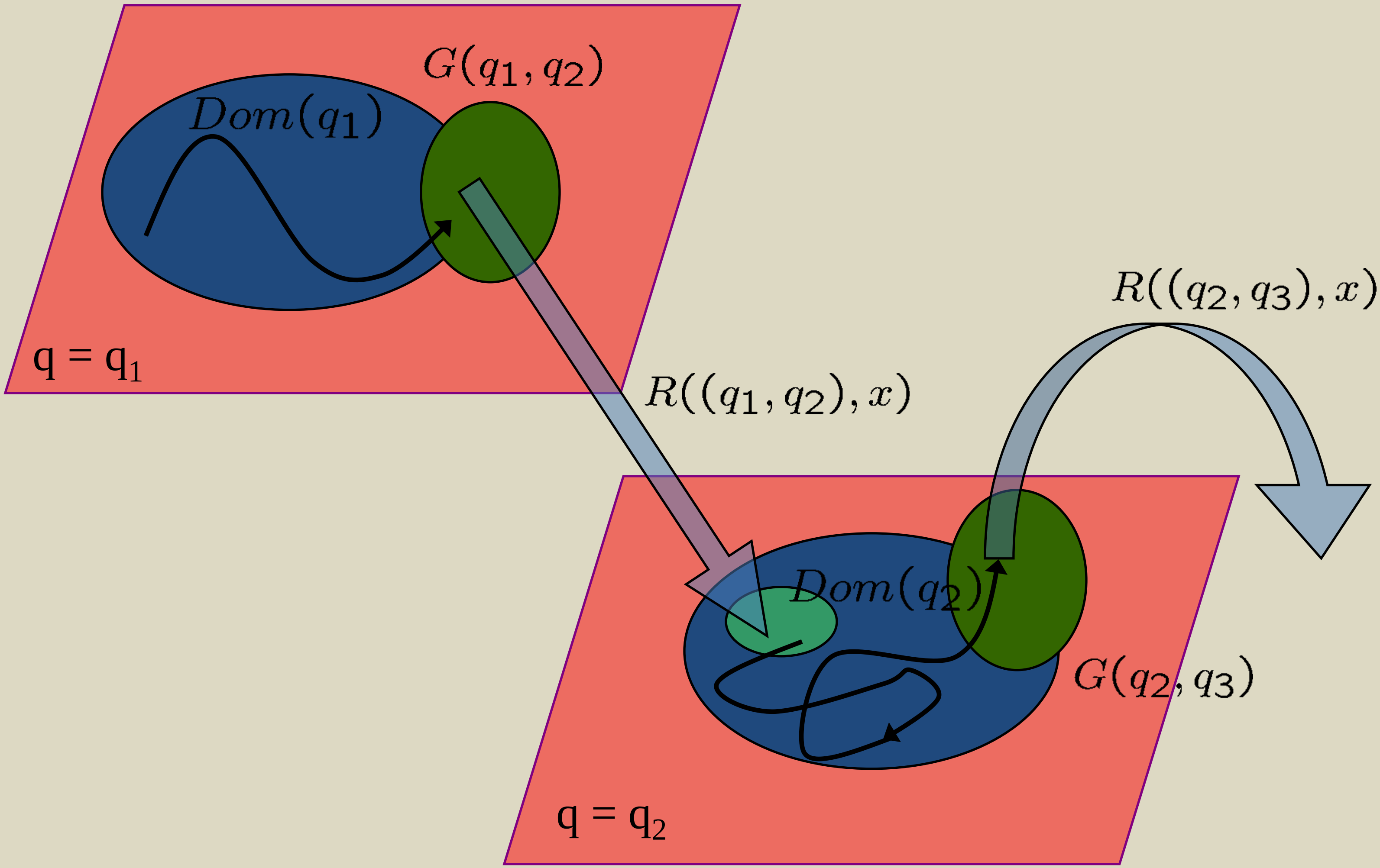
Hybrid systems theory



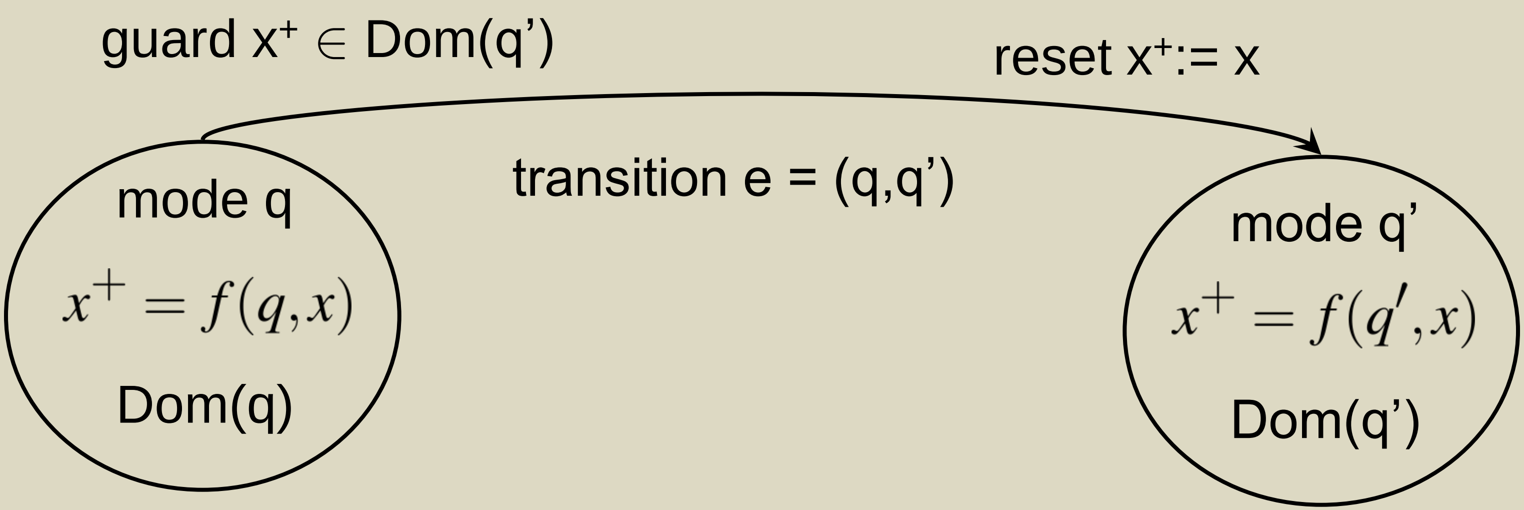
Hybrid automaton



Hybrid automaton



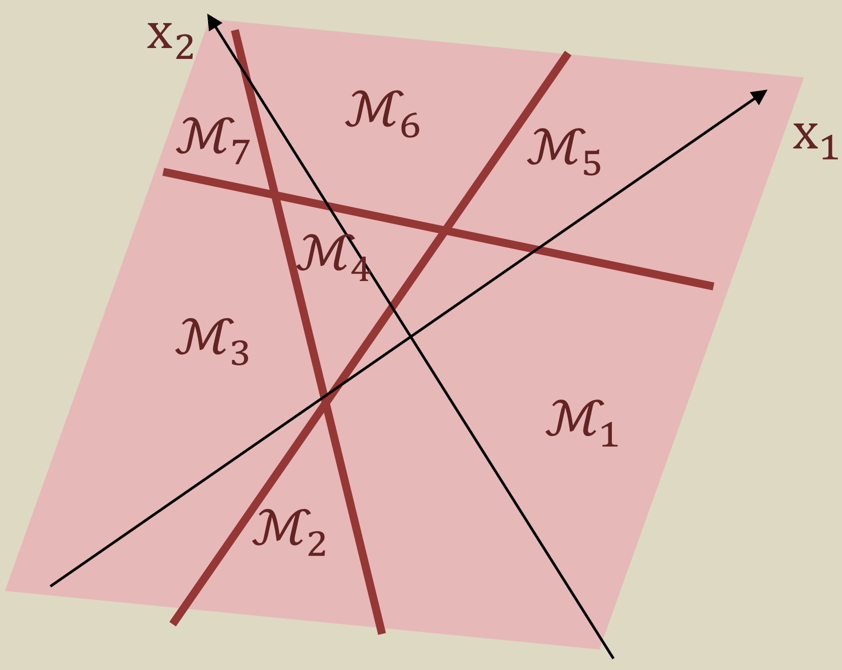
Hybrid automaton: switched system



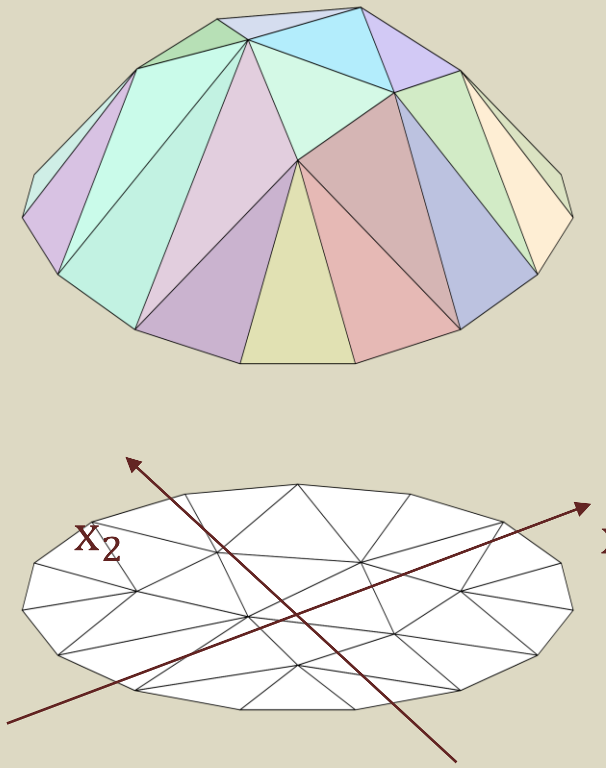
$$\cup_{q \in Q} \mathrm{Dom}(q) = R^n$$

$$\mathrm{Dom}(q) \cap \mathrm{Dom}(q') = \emptyset,\ q, q' \in Q,\ q' \neq q$$

Piecewise affine (PWA) systems



partition in modes



PWA continuous dynamics

Cyber-physical systems

Credit: National Institute of Standards and Technology (NIST)

Safety

In a safety-critical system, some region of the state space is "unsafe".

Safety

In a safety-critical system, some region of the state space is "unsafe".

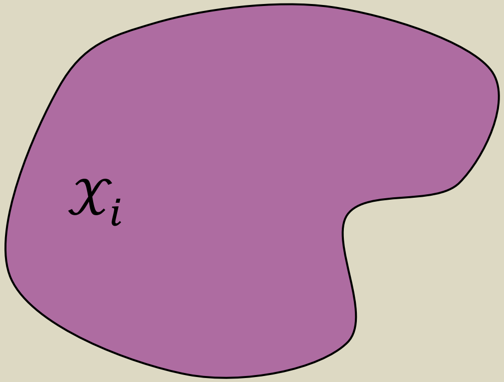
One has to verify that the system operates in safe conditions, i.e., it keeps staying inside the safe set.
If that is not the case the system has to be modified so as to guarantee safety.

Reachability

Given a system and a set of initial conditions $\mathcal{X}_i$
determine the set of states that can be reached by the system starting from $\mathcal{X}_i$

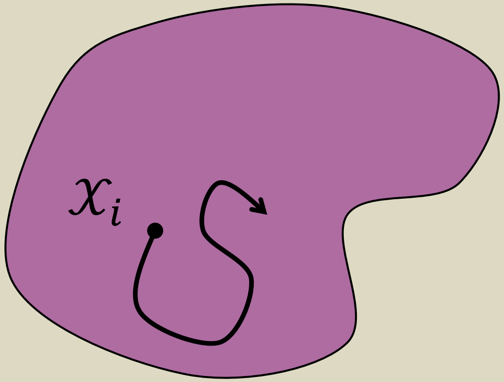
Reachability

Given a system and a set of initial conditions $\mathcal{X}_i$
determine the set of states that can be reached by the system starting from $\mathcal{X}_i$



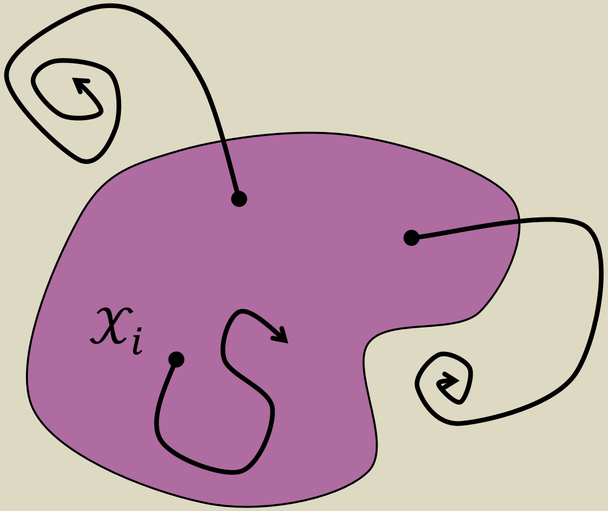
Reachability

Given a system and a set of initial conditions $\mathcal{X}_i$
determine the set of states that can be reached by the system starting from $\mathcal{X}_i$



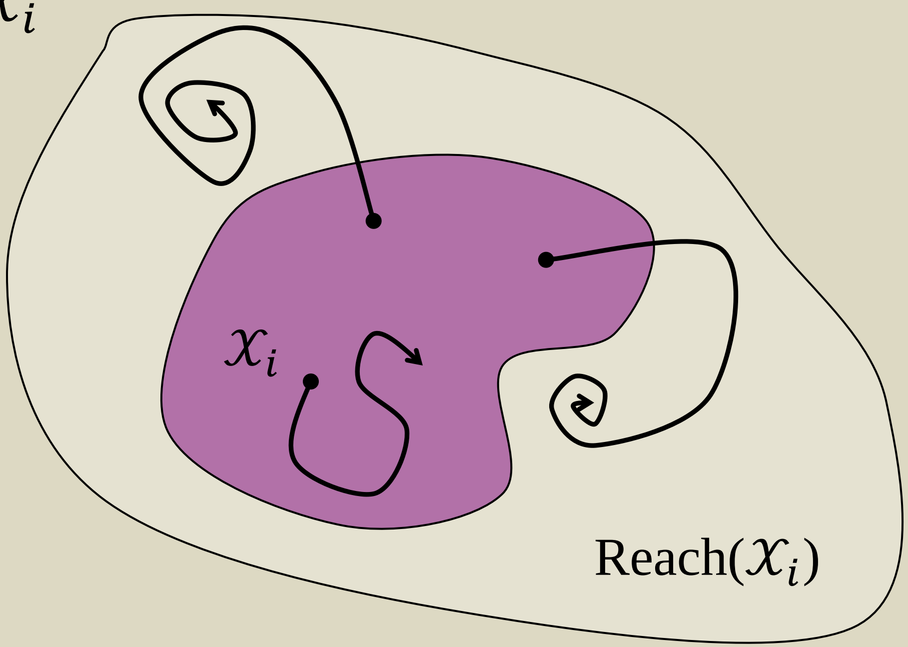
Reachability

Given a system and a set of initial conditions $\mathcal{X}_i$
determine the set of states that can be reached by the system starting from $\mathcal{X}_i$



Reachability

Given a system and a set of initial conditions $\mathcal{X}_i$
determine the set of states that can be reached by the system starting from $\mathcal{X}_i$

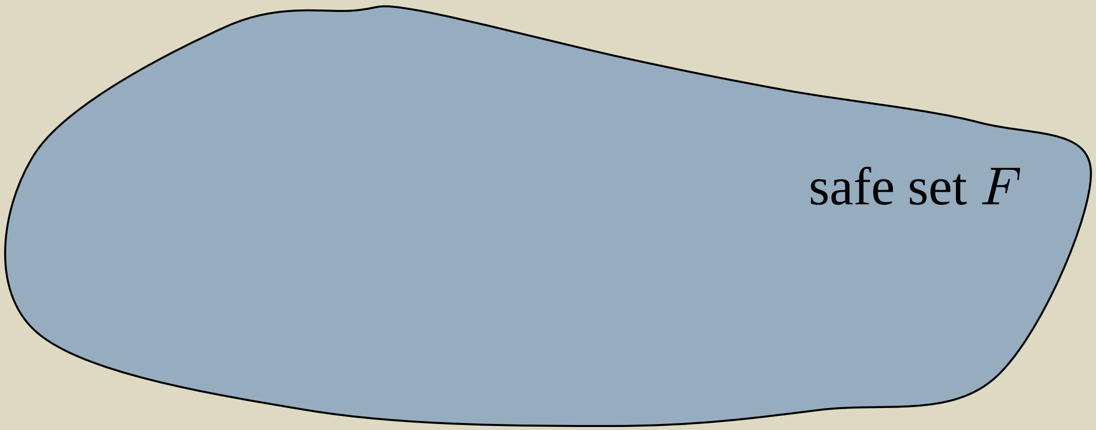


Reachability versus safety

Reachability analysis can be used for safety verification

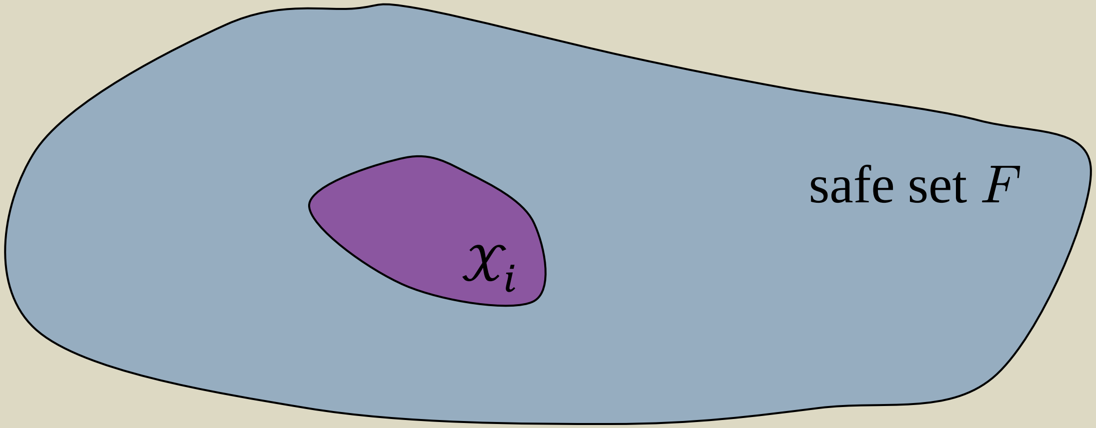
Reachability versus safety

Reachability analysis can be used for safety verification



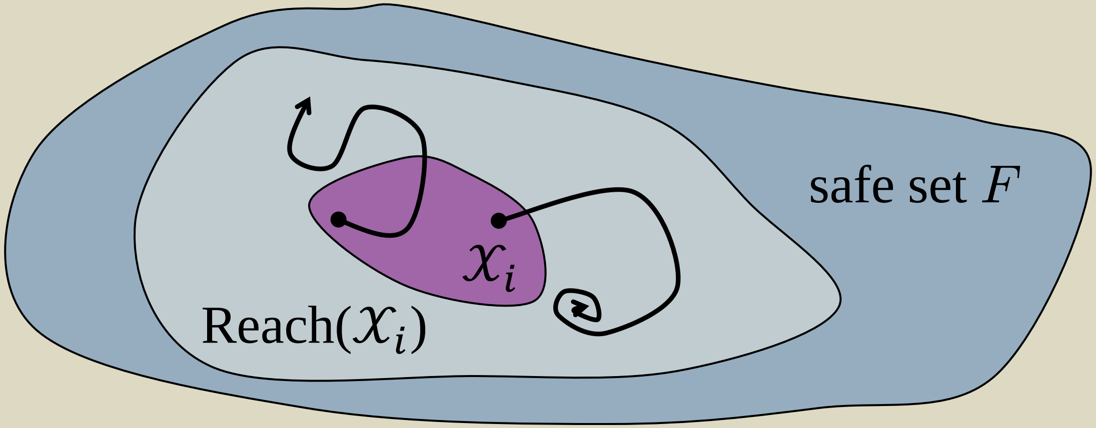
Reachability versus safety

Reachability analysis can be used for safety verification



Reachability versus safety

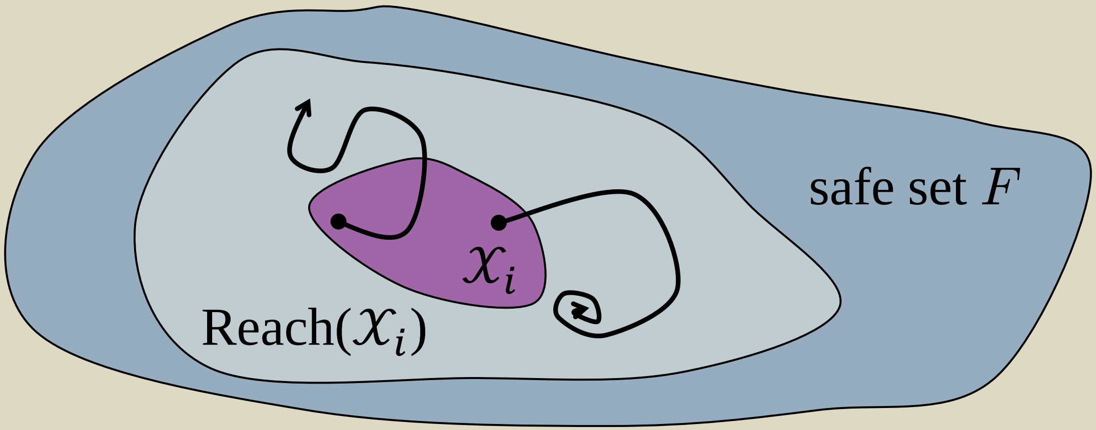
Reachability analysis can be used for safety verification



Reach($\mathcal{X}_i$) $\subseteq F$

Reachability versus safety

Reachability analysis can be used for safety verification

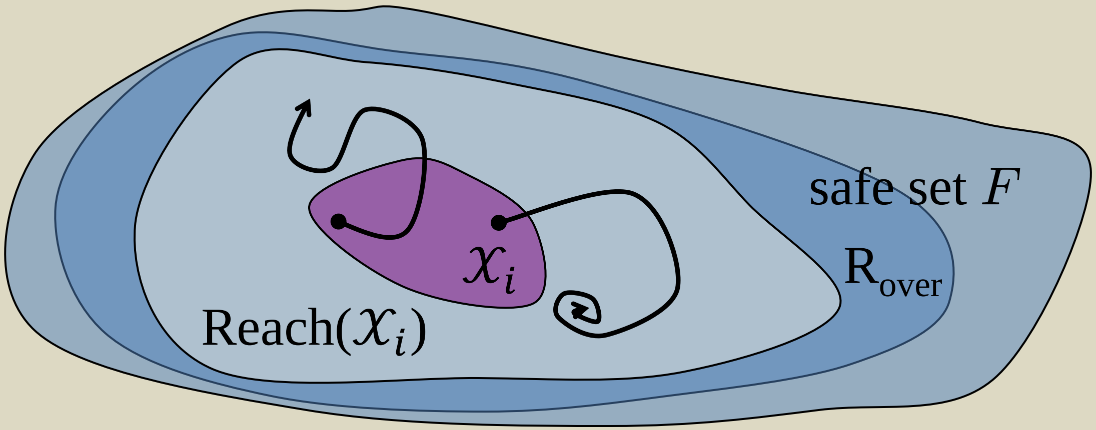


Reach($\mathcal{X}_i$) $\subseteq F$

the system is operating in safe conditions

Reachability versus safety

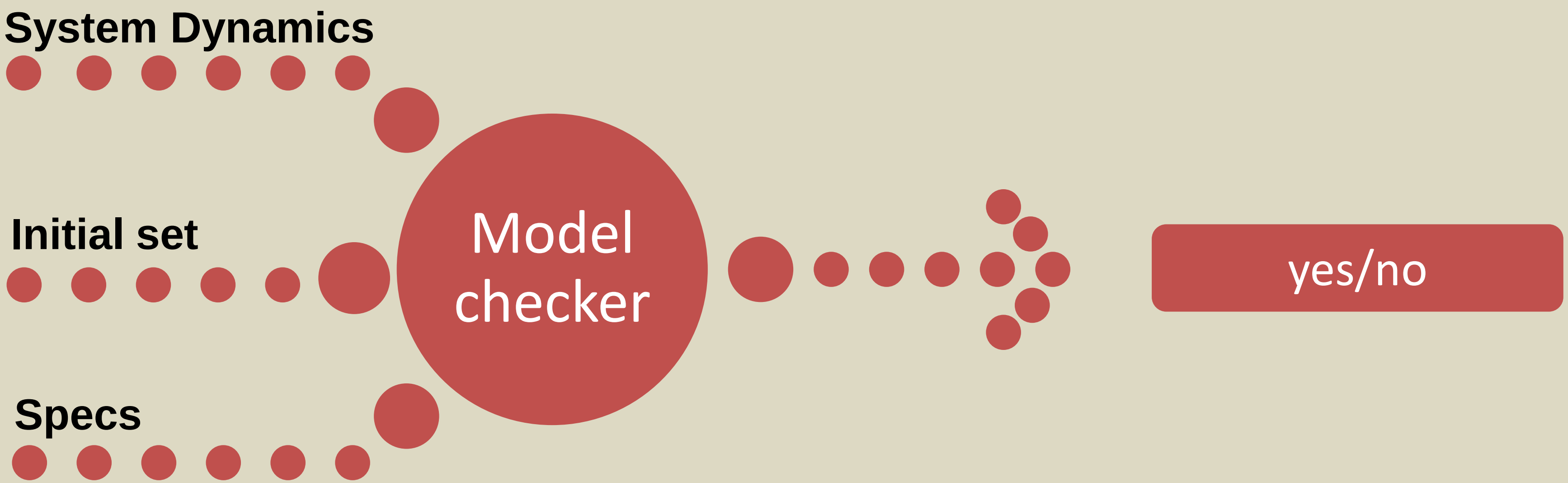
Reachability analysis can be used for safety verification



$$\text{Reach}(\mathcal{X}_i) \subseteq \mathrm{R}_{\text{over}} \subseteq F$$

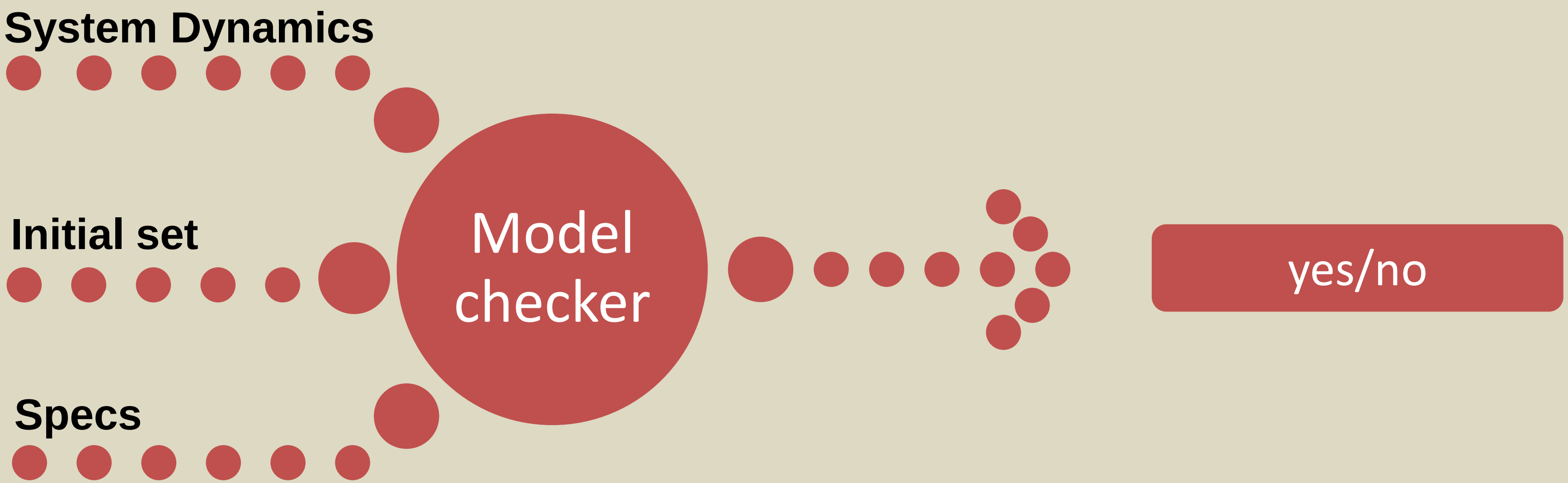
the system is operating in safe conditions

Model checking



Automated verification of safety can be performed via reach set computations based on a model of the system

Model checking



Automated verification of safety can be performed via reach set computations based on a model of the system

this requires to be able to “compute” with sets (represent sets and propagate them through the system dynamics)

Model checking for finite automata

$S = \{s_1, s_2, \ldots\}$ finite set of states
$\mathcal{E} = \{a, b, c, \ldots\}$ finite set of input symbols (events)
$\mathcal{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S}$ transition relation

Model checking for finite automata

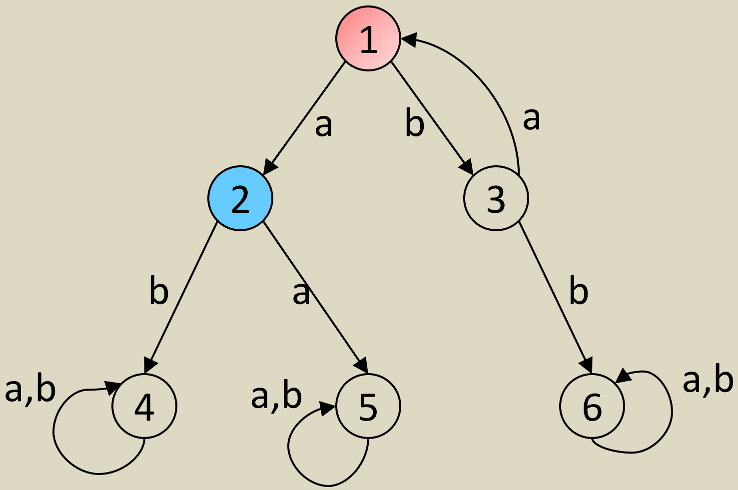
$S = \{s_1, s_2, \ldots\}$ finite set of states
$\mathcal{E} = \{a, b, c, \ldots\}$ finite set of input symbols (events)
$\mathcal{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S}$ transition relation

One-step successor operator:

$\text{Post}: 2^S \to 2^S$

$\Omega \subseteq 2^S \Rightarrow \text{Post}(\Omega) = \{s' \in S : \exists\, s \in \Omega, e \in \mathcal{E}, (s,e,s') \in \mathcal{T}\}$

Model checking for finite automata

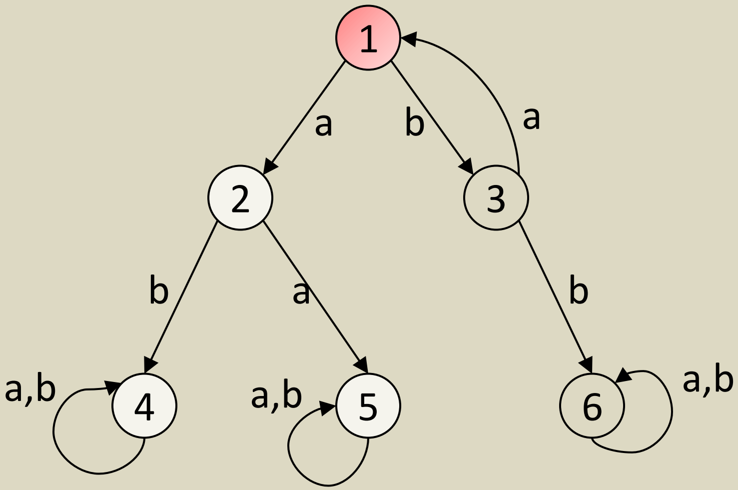


S_0 = {2}

Safe set:

F = {2,3,4,5,6}

Model checking for finite automata



S_0 = {2}

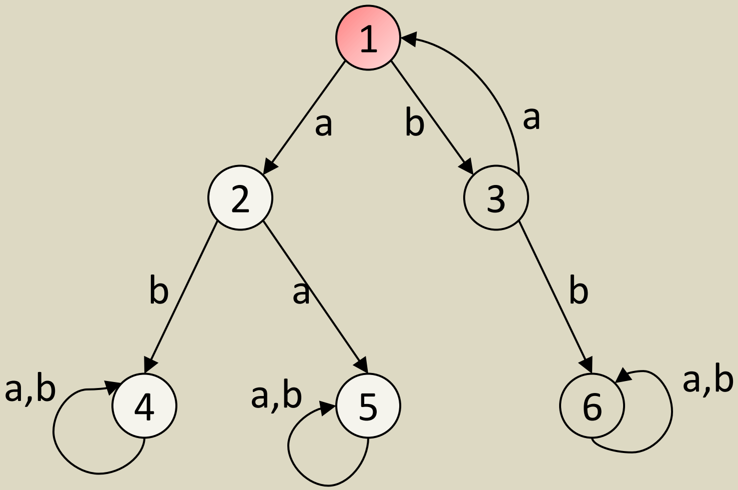
Reach$_0$ = {2}

Reach$_1$ = Post(Reach$_0$) ={4,5}

Safe set:

F = {2,3,4,5,6}

Model checking for finite automata



S_0 = {2}

Safe set:

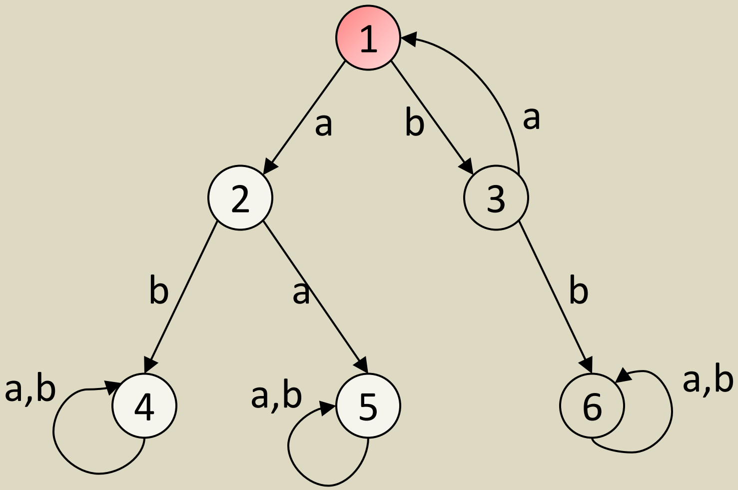
F = {2,3,4,5,6}

Reach$_0$ = {2}

Reach$_1$ = Post(Reach$_0$) ={4,5}

Reach$_{\leq 1}$ = Reach$_0$ $\cup$ Reach$_1$ = {2,4,5}

Model checking for finite automata



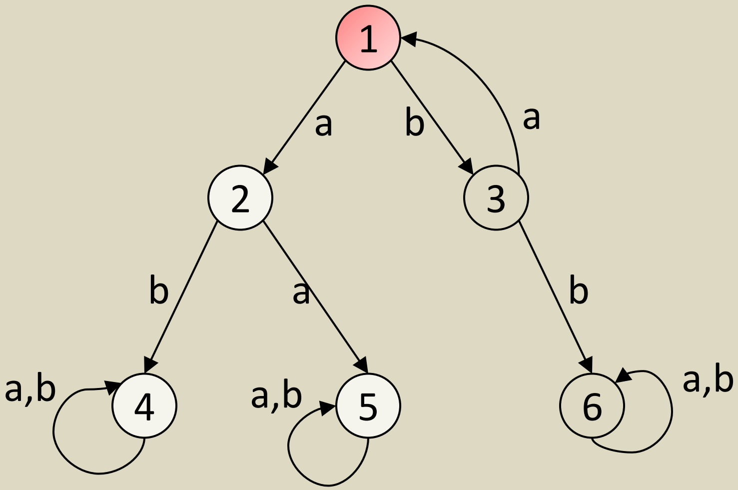
S_0 = {2}

Safe set:

F = {2,3,4,5,6}

$\text{Reach}_0 = \{2\}$
$\text{Reach}_1 = \text{Post}(\text{Reach}_0) = \{4,5\}$
$\text{Reach}_{\leq 1} = \text{Reach}_0 \cup \text{Reach}_1 = \{2,4,5\}$
$\text{Reach}_2 = \text{Post}(\text{Reach}_1) = \{4,5\}$

Model checking for finite automata



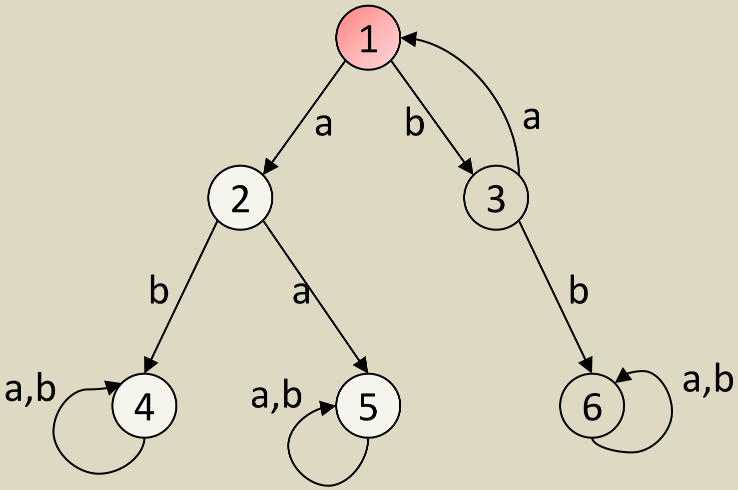
S_0 = {2}

Reach$_0$ = {2}
Reach$_1$ = Post(Reach$_0$) ={4,5}
Reach$_{\leq 1}$ = Reach$_0$ $\cup$ Reach$_1$ = {2,4,5}
Reach$_2$ = Post(Reach$_1$) ={4,5}
Reach$_{\leq 2}$ = Reach$_{\leq 1}$ $\cup$ Reach$_2$ = {2,4,5}

Safe set:

F = {2,3,4,5,6}

Model checking for finite automata



S_0 = {2}

Safe set:

F = {2,3,4,5,6}

Reach_0 = {2}

Reach_1 = Post(Reach_0) ={4,5}

$\text{Reach}_{\leq 1}$ = $\text{Reach}_0 \cup \text{Reach}_1$ = {2,4,5} $\subseteq F$

Reach_2 = Post(Reach_1) ={4,5}

$\text{Reach}_{\leq 2}$ = $\text{Reach}_{\leq 1} \cup \text{Reach}_2$ = {2,4,5}

$\text{Reach}_{\leq 2}$ = $\text{Reach}_{\leq 1} \subseteq F \rightarrow$ safe

Model checking for finite automata

Safety verification algorithm

initialization: Reach $_{\leq -1} = \emptyset$;
Reach $_{\leq 0} = S_0$
$i = 0$

loop: while Reach$_{\leq i} \neq$ Reach$_{\leq i-1}$ and Reach$_{\leq i} \subseteq$ safe set F do
Reach$_{\leq i+1}$ = Reach$_{\leq i} \cup$ Post{Reach$_{\leq i}$}
$i = i + 1$

output: if Reach$_{\leq i}$ = Reach$_{\leq i-1}$ then the system is safe else it is not safe

Model checking for finite automata

Safety verification algorithm

initialization: $\text{Reach}_{\leq -1} = \emptyset$;

$\text{Reach}_{\leq 0} = S_0$

$i = 0$

loop: while $\text{Reach}_{\leq i} \neq \text{Reach}_{\leq i\text{-}1}$ and $\text{Reach}_{\leq i} \subseteq$ safe set F do

$\text{Reach}_{\leq i+1} = \text{Reach}_{\leq i} \cup \text{Post}\{\text{Reach}_{\leq i}\}$

$i = i + 1$

output: if $\text{Reach}_{\leq i} = \text{Reach}_{\leq i\text{-}1}$ then the system is safe else it is not safe

Theorem

For a finite automaton, the safety property is decidable (i.e., there exists a computational procedure that decides in a finite number of steps whether the system is safe or not)

Hybrid automata as transition systems

hybrid automaton $H = (Q, X, f, \mathit{Init}, \mathrm{Dom}, E, G, R)$

transition system

$$\begin{cases} \mathcal{S} = Q \times X & \equiv \text{set of states (infinite)} \\ \mathcal{E} = E \cup \{\tau\} & \equiv \text{alphabet of events:} \\ & \quad e \in E \text{ jump event} \\ & \quad \tau \text{ continuous evolution event} \\ \mathcal{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S} & \equiv \text{transition relation} \end{cases}$$

Hybrid automata as transition systems

hybrid automaton $H = (Q, X, f, \mathit{Init}, \mathrm{Dom}, E, G, R)$

$\Updownarrow$ same reachability properties

transition system
$$\begin{cases} \mathcal{S} = Q \times X & \equiv \text{ set of states (infinite)} \\ \mathcal{E} = E \cup \{\tau\} & \equiv \text{ alphabet of events:} \\ & \quad e \in E \text{ jump event} \\ & \quad \tau \text{ continuous evolution event} \\ \mathcal{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S} & \equiv \text{ transition relation} \end{cases}$$

Hybrid automata as transition systems

hybrid automaton $H = (Q, X, f, \mathit{Init}, \mathrm{Dom}, E, G, R)$

same reachability properties

transition system
$\mathcal{S} = Q \times X$ ≡ set of states (infinite)
$\mathcal{E} = E \cup \{\tau\}$ ≡ alphabet of events:
$e \in E$ jump event
τ continuous evolution event
$\mathcal{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S}$ ≡ transition relation

Same safety algorithm as for finite automata

Hybrid automata as transition systems

hybrid automaton $H = (Q, X, f, \mathit{Init}, \mathrm{Dom}, E, G, R)$

$\Updownarrow$ same reachability properties

transition system

$$\left\{\begin{array}{ll} \mathcal{S} = Q \times \boxed{X} & \equiv \text{set of states (infinite)} \\ \mathcal{E} = E \cup \{\tau\} & \equiv \text{alphabet of events:} \\ & \quad e \in E \text{ jump event} \\ & \quad \boxed{\tau \text{ continuous evolution event}} \\ \mathcal{T} \subset \mathcal{S} \times \mathcal{E} \times \mathcal{S} & \equiv \text{transition relation} \end{array}\right.$$

$\Downarrow$

Same safety algorithm as for finite automata

Model checking for continuous systems

If S is infinite then the safety algorithm is not guaranteed to terminate

Model checking for continuous systems

If <u>S is infinite</u> then the safety algorithm is not guaranteed to terminate

<u>Example</u>

$\mathcal{S} = \mathrm{R}$ $\quad \mathcal{E} = \{e\}$ $\quad \mathcal{T} = \{(s, e, 0.5\, s), s \in \mathrm{R}\}$

$\mathcal{S}_0 = \{1\}$, $F = [-1,2]$

$\mathrm{Reach}_0 = \mathcal{S}_0$

$\mathrm{Reach}_{\leq 1} = \mathrm{Reach}_0 \cup \mathrm{Post}\{\mathrm{Reach}_0\} = \{1, 0.5\} \neq \mathrm{Reach}_0$

$\wedge\ \mathrm{Reach}_{\leq 1} \subseteq F$

$\mathrm{Reach}_{\leq 2} = \mathrm{Reach}_{\leq 1} \cup \mathrm{Post}\{\mathrm{Reach}_{\leq 1}\} = \{1, 0.5, 0.5^2\} \neq \mathrm{Reach}_{\leq 1}$

$\wedge\ \mathrm{Reach}_{\leq 2} \subseteq F$

$\mathrm{Reach}_{\leq 3} = \mathrm{Reach}_{\leq 2} \cup \mathrm{Post}\{\mathrm{Reach}_{\leq 2}\} = \{1, 0.5, 0.5^2, 0.5^3\} \neq \mathrm{Reach}_{\leq 2}$

$\wedge\ \mathrm{Reach}_{\leq 3} \subseteq F$

…..

Model checking for continuous systems

If S is infinite then the safety algorithm is not guaranteed to terminate

Also… set representation and propagation can be an issue

Reach set computation for a linear system

Consider a discrete time continuous system described by

$$x^+ = A\,x$$

The execution of the systems starting from $x = x_0$ can be expressed analytically as

$$x(k) = A^k x_0, k \geq 0$$

Reach set computation for a linear system

Consider a discrete time continuous system described by

$$x^{+} = A\,x$$

The execution of the systems starting from $x = x_0$ can be expressed analytically as

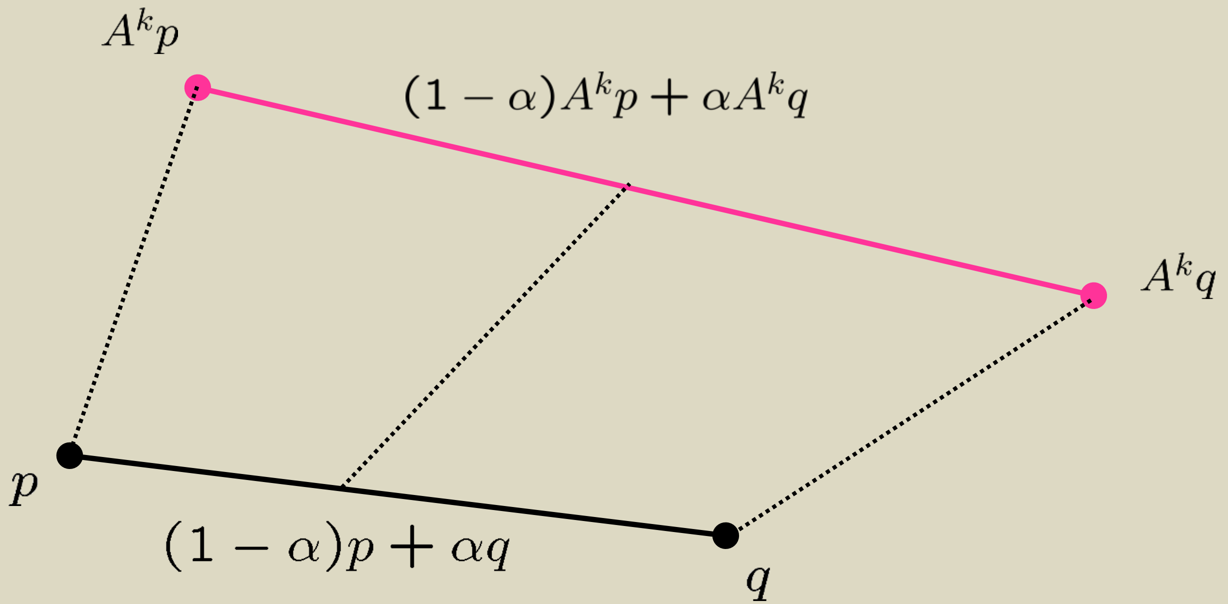
$$x(k) = A^{k} x_0, k \geq 0$$

Since this expression is linear in x_0,

then, initial states on a segment are mapped into a segment

at each time step k> 0

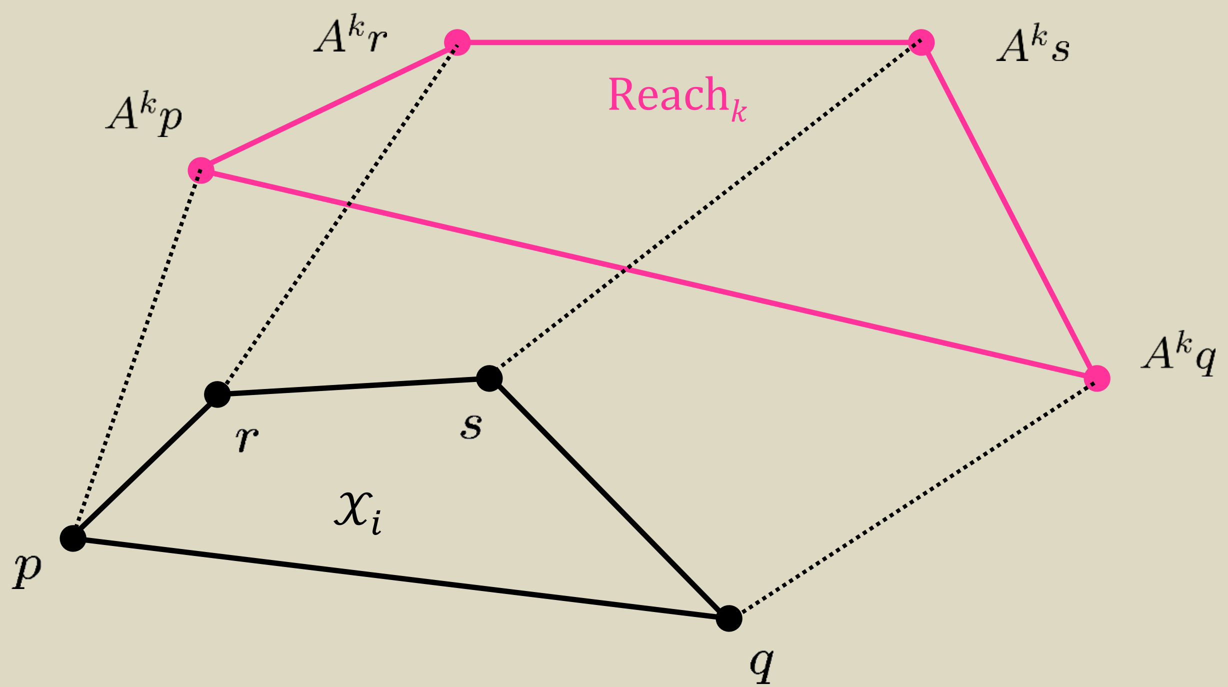
Reach set computation for a linear system



$\mathcal{X}_i = \{(1-\alpha)p + \alpha q, \alpha \in [0,1]\}$

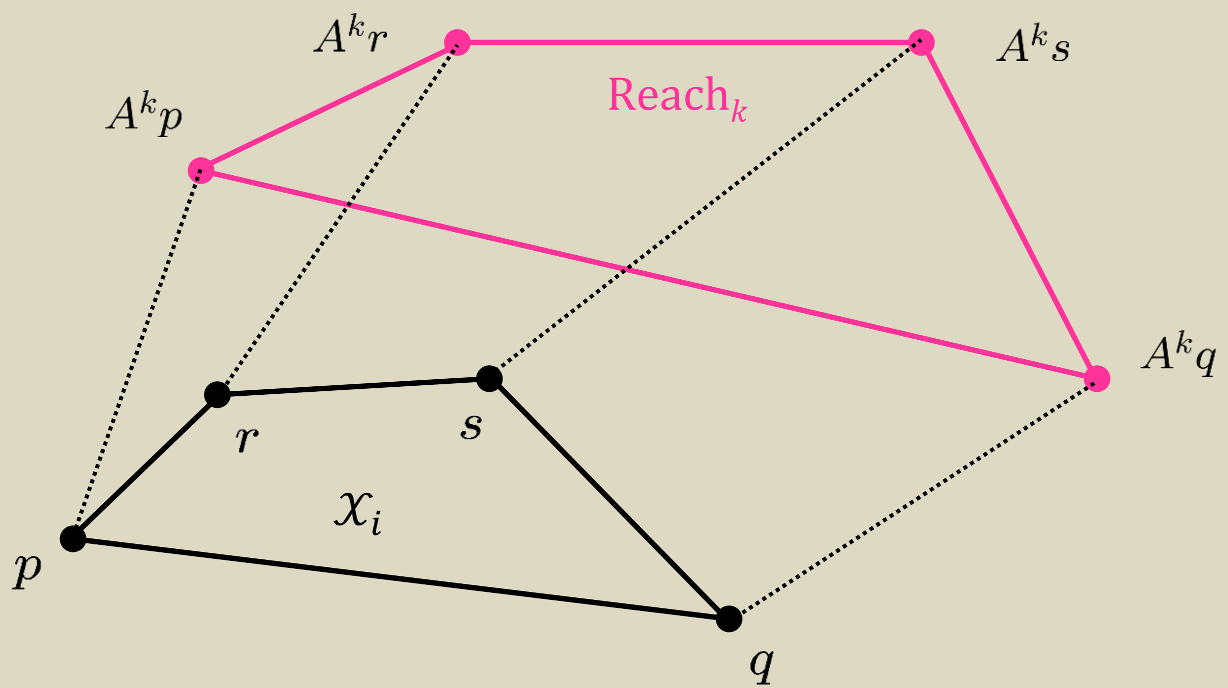
$\text{Reach}_k = \{(1-\alpha)(A^k p) + \alpha(A^k q), \alpha \in [0,1]\}$ reach set at time k

Reach set computation for a linear system



$\mathcal{X}_i$ convex is mapped into Reach$_k$ convex

Reach set computation for a linear system



$\mathcal{X}_i$ convex is mapped into Reach_k convex, but shape varies since distance and orientation of a segment are not preserved

Reach set computation for a linear system

What if the system is affected by some set-valued input?

$$x^+ = A\,x + w, \qquad w \in \mathcal{W}, \qquad x \in \mathcal{X}$$

Reach set computation for a linear system

What if the system is affected by some set-valued input?

$$x^+ = A\,x + w, \qquad w \in \mathcal{W}, \qquad x \in \mathcal{X}$$

$$Post\{\mathcal{X}\} = \{x^+ \colon\; x^+ = Ax + w, x \in \mathcal{X}, w \in \mathcal{W}\}$$

Reach set computation for a linear system

What if the system is affected by some set-valued input?

$$x^+ = A\,x + w, \qquad w \in \mathcal{W}, \qquad x \in \mathcal{X}$$

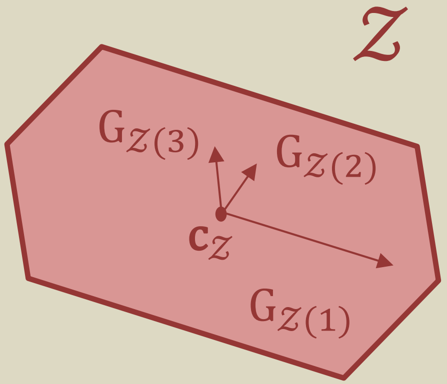
$$Post\{\mathcal{X}\} = \{x^+ \colon\; x^+ = Ax + w, x \in \mathcal{X}, w \in \mathcal{W}\}$$

Desired properties:

- Compact set representation
- Closure with respect to linear transformation and Minkowski sum

Zonotopes

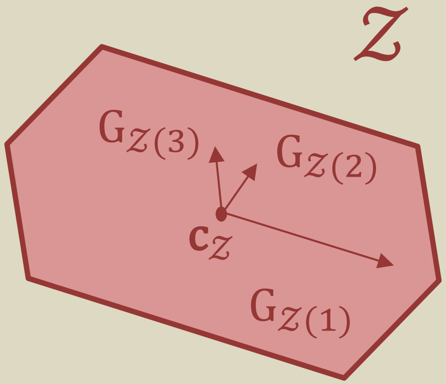
Zonotope = Centrally symmetric polytope



$$x \in \mathcal{Z} \subset \mathcal{R}^n \Leftrightarrow x = \mathrm{c}_{\mathcal{Z}} + \sum_{i=1}^{p} \alpha_i(\mathrm{x})\mathrm{G}_{\mathcal{Z}(i)},\ \alpha_i \in [-1,1] \quad \forall i$$

Zonotopes

Zonotope = Centrally symmetric polytope



$$x \in \mathcal{Z} \subset \mathcal{R}^n \Leftrightarrow x = \mathrm{c}_{\mathcal{Z}} + \sum_{i=1}^{p} \alpha_i(\mathrm{x})\mathrm{G}_{\mathcal{Z}(i)},\ \alpha_i \in [-1,1]\ \ \forall i$$

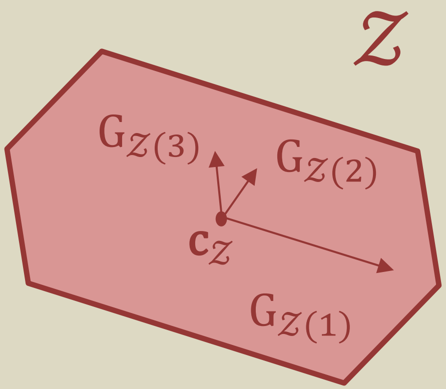
$\mathrm{c}_{\mathcal{Z}}$: center

$\mathrm{G}_{\mathcal{Z}} = [\mathrm{G}_{\mathcal{Z}(1)} \dots \mathrm{G}_{\mathcal{Z}(p)}]$: generators matrix

$\mathcal{Z} = \langle \mathrm{c}_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle$
of order p/n

Zonotopes

Zonotope = Centrally symmetric polytope



$$x \in \mathcal{Z} \subset \mathcal{R}^n \Leftrightarrow x = c_{\mathcal{Z}} + \sum_{i=1}^{p} \alpha_i(x) G_{\mathcal{Z}(i)},\ \alpha_i \in [-1,1] \quad \forall i$$

Parallelotope is a zonotope of order 1 with $G_{\mathcal{Z}}$ invertible

Zonotopes

Closed under affine transformations:

$$\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle \implies A\mathcal{Z} + f = \langle Ac_{\mathcal{Z}} + f, AG_{\mathcal{Z}} \rangle$$

Zonotopes

Closed under affine transformations:

$$\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle \implies A\mathcal{Z} + f = \langle Ac_{\mathcal{Z}} + f, AG_{\mathcal{Z}} \rangle$$

Closed under Minkowski sum:

$$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \{x : x = x_1 + x_2, x_1 \in \mathcal{Z}_1, x_2 \in \mathcal{Z}_2\}$$

$$\mathcal{Z}_1 = \langle c_{\mathcal{Z}_1}, G_{\mathcal{Z}_1} \rangle, \mathcal{Z}_2 = \langle c_{\mathcal{Z}_2}, G_{\mathcal{Z}_2} \rangle \implies \mathcal{Z}_1 \oplus \mathcal{Z}_2 = \langle c_{\mathcal{Z}_1} + c_{\mathcal{Z}_2}, [G_{\mathcal{Z}_1} \, G_{\mathcal{Z}_2}] \rangle$$

Zonotopes

Closed under affine transformations:

$$\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle \implies A\mathcal{Z} + f = \langle Ac_{\mathcal{Z}} + f, AG_{\mathcal{Z}} \rangle$$

Closed under Minkowski sum:

$$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \{x : x = x_1 + x_2, x_1 \in \mathcal{Z}_1, x_2 \in \mathcal{Z}_2\}$$

$$\mathcal{Z}_1 = \left\langle c_{\mathcal{Z}_1}, G_{\mathcal{Z}_1} \right\rangle, \mathcal{Z}_2 = \left\langle c_{\mathcal{Z}_2}, G_{\mathcal{Z}_2} \right\rangle \Rightarrow \mathcal{Z}_1 \oplus \mathcal{Z}_2 = \left\langle c_{\mathcal{Z}_1} + c_{\mathcal{Z}_2}, \left[G_{\mathcal{Z}_1} \, G_{\mathcal{Z}_2} \right] \right\rangle$$

Easy to compute: sum the centers and concatenate the generators

Zonotopes

Closed under affine transformations:

$$\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle \implies A\mathcal{Z} + f = \langle Ac_{\mathcal{Z}} + f, AG_{\mathcal{Z}} \rangle$$

Closed under Minkowski sum:

$$\mathcal{Z}_1 \oplus \mathcal{Z}_2 = \{x : x = x_1 + x_2, x_1 \in \mathcal{Z}_1, x_2 \in \mathcal{Z}_2\}$$

$$\mathcal{Z}_1 = \langle c_{\mathcal{Z}_1}, G_{\mathcal{Z}_1} \rangle, \mathcal{Z}_2 = \langle c_{\mathcal{Z}_2}, G_{\mathcal{Z}_2} \rangle \implies \mathcal{Z}_1 \oplus \mathcal{Z}_2 = \langle c_{\mathcal{Z}_1} + c_{\mathcal{Z}_2}, [G_{\mathcal{Z}_1} G_{\mathcal{Z}_2}] \rangle$$

Easy to compute: sum the centers and concatenate the generators
… but the order keeps growing as we keep propagating

An example

$$x^+ = \begin{bmatrix} 0.2 & 0.4 \\ 0.6 & 0.8 \end{bmatrix} x + w,\ w \in \mathcal{W}$$

$$\mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

$$\mathcal{W} = [-0.2, 0.2]^2$$

An example

$$x^+ = \begin{bmatrix} 0.2 & 0.4 \\ 0.6 & 0.8 \end{bmatrix} x + w,\ w \in \mathcal{W} \qquad \mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

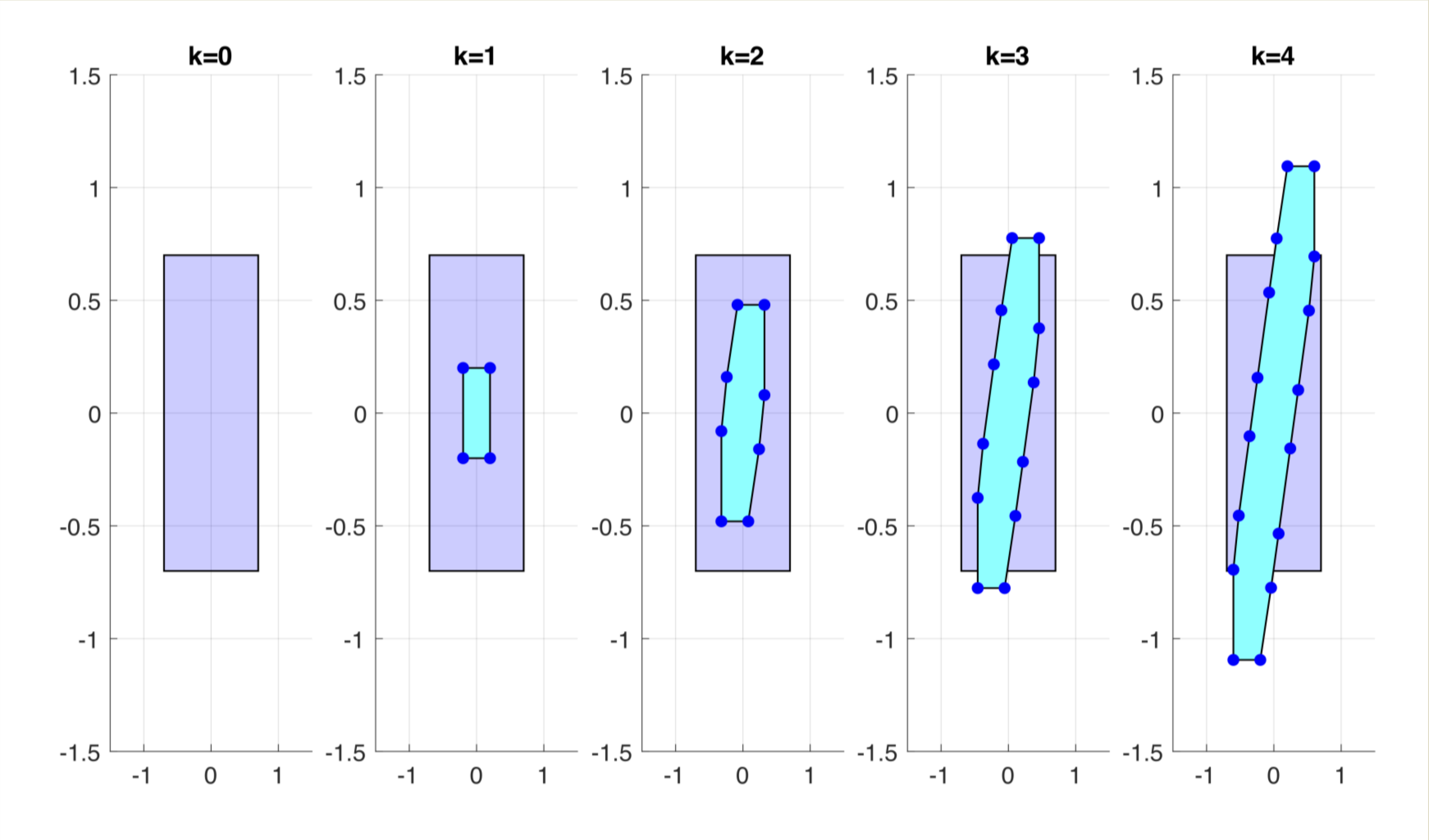
$$\mathcal{W} = [-0.2, 0.2]^2$$

$\mathcal{W}$: Parallelotope (box)

$\mathcal{W} = \langle c_{\mathcal{W}}, G_{\mathcal{W}} \rangle$, with $c_{\mathcal{W}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and $G_{\mathcal{W}} = \begin{bmatrix} 0.2 & 0 \\ 0 & 0.2 \end{bmatrix}$

Number of generators: $p = 2$, order $\frac{p}{n} = 1$

An example



An example

$$x^+ = \begin{bmatrix} 0.2 & 0.4 \\ 0.6 & 0.8 \end{bmatrix} x + w, \; w \in [-0.2, 0.2]^2 \qquad \mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

at every step the order of the reach set increases by $\frac{p}{n} = 1$:

An example

$$x^+ = \begin{bmatrix} 0.2 & 0.4 \\ 0.6 & 0.8 \end{bmatrix} x + w,\ w \in [-0.2, 0.2]^2 \qquad \mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

at every step the order of the reach set increases by $\frac{p}{n} = 1$:

at step k=1: order 1 (2 generators)
at step k=2: order 2 (4 generators)
at step k=3: order 3 (6 generators)
…

An example

$$x^+ = \begin{bmatrix} 0.2 & 0.4 \\ 0.6 & 0.8 \end{bmatrix} x + w,\ w \in [-0.2, 0.2]^2 \qquad \mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

at every step the order of the reach set increases by $\frac{p}{n} = 1$:

at step k=1: order 1 (2 generators)
at step k=2: order 2 (4 generators)
at step k=3: order 3 (6 generators)
…

need for order reduction

Transformation to a parallelotope (order 1)

- linearly transform a zonotope $\mathcal{Z}$ by a matrix such that its shape becomes similar to a box (i.e. an axes-aligned parallelotope)

Transformation to a parallelotope (order 1)

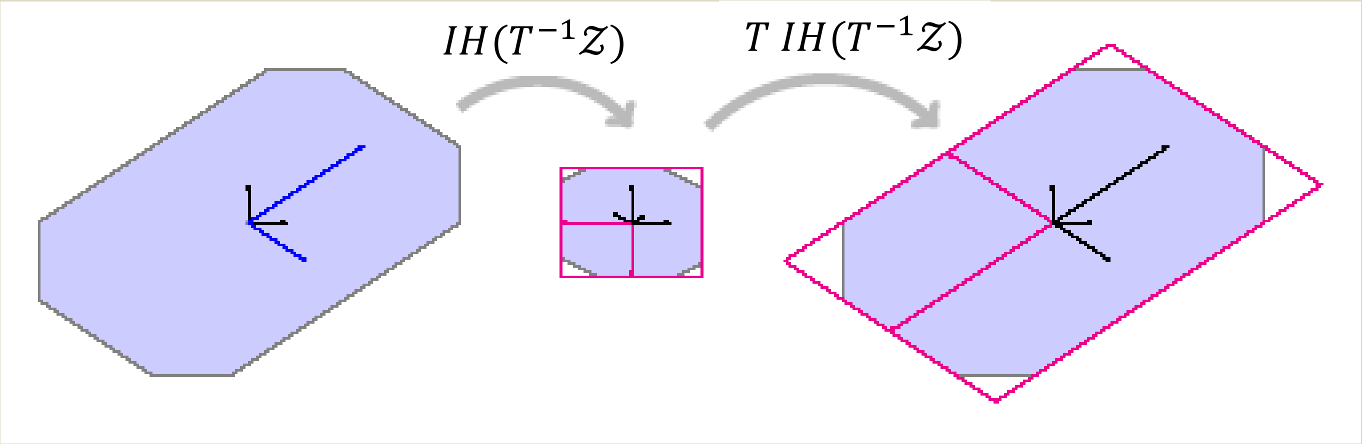
- linearly transform a zonotope $\mathcal{Z}$ by a matrix such that its shape becomes similar to a box (i.e. an axes-aligned parallelotope)
- outer-approximate the transformed zonotope by its box-shaped interval hull

Transformation to a parallelotope (order 1)

- linearly transform a zonotope $\mathcal{Z}$ by a matrix such that its shape becomes similar to a box (i.e. an axes-aligned parallelotope)
- outer-approximate the transformed zonotope by its box-shaped interval hull
- transform back into the original space to obtain an over-approximating parallelotope $\hat{\mathcal{Z}}$

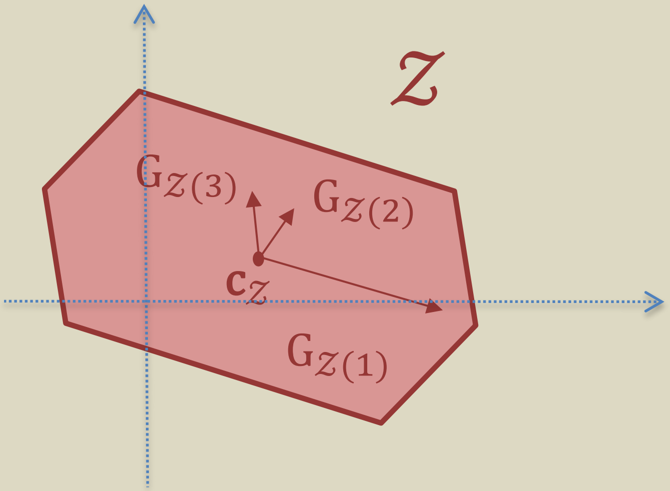
Transformation to a parallelotope (order 1)

- linearly transform a zonotope $\mathcal{Z}$ by a matrix such that its shape becomes similar to a box (i.e. an axes-aligned parallelotope)
- outer-approximate the transformed zonotope by its box-shaped interval hull
- transform back into the original space to obtain an over-approximating parallelotope $\hat{\mathcal{Z}}$



Transformation matrix computed via PCA

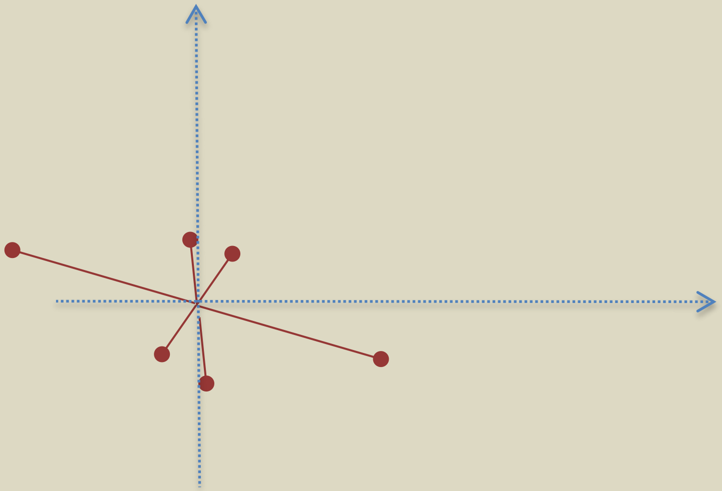
Input: $\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle$



Transformation matrix computed via PCA

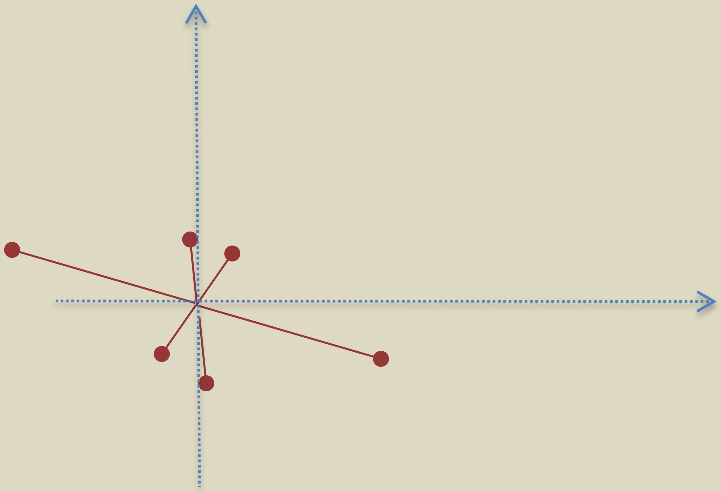
Input: $\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle$

Set $\mathrm{X} = [G_{\mathcal{Z}} \quad - G_{\mathcal{Z}}]$



Transformation matrix computed via PCA

Input: $\mathcal{Z} = \langle c_{\mathcal{Z}}, G_{\mathcal{Z}} \rangle$



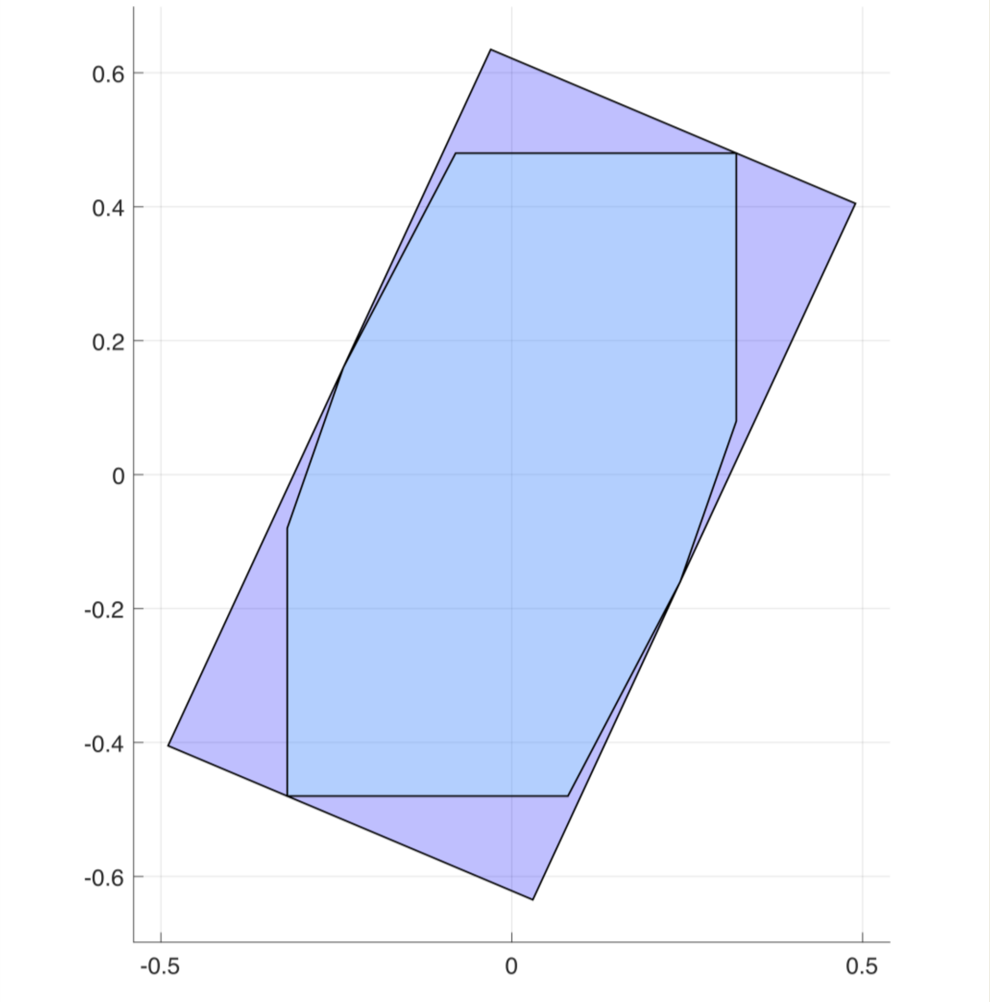
Set $\mathrm{X} = [G_{\mathcal{Z}} \quad -G_{\mathcal{Z}}]$

Compute $\mathrm{C} = XX^T$

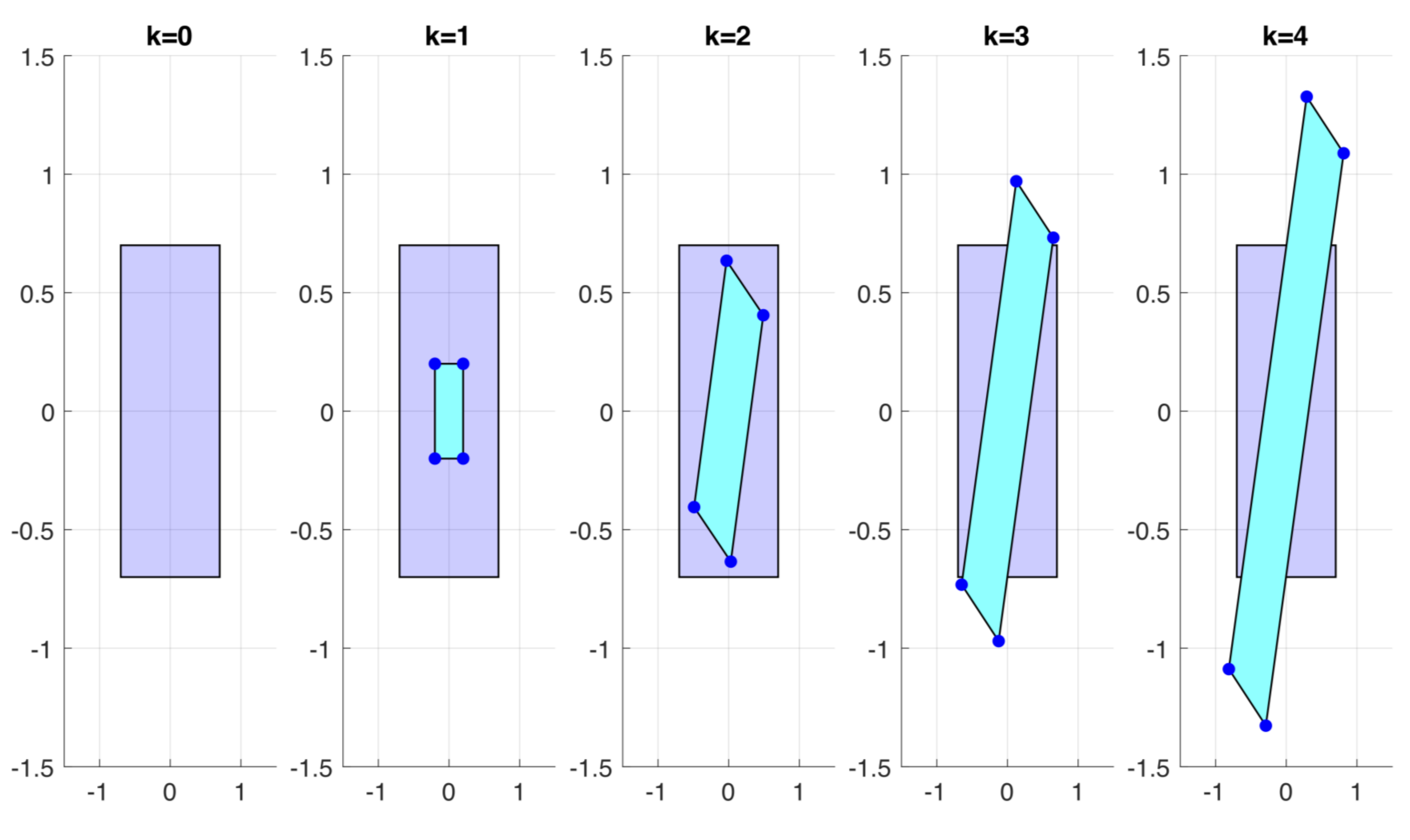
SVD decomposition $\mathrm{C} = U\Sigma V^T$

Output: $\hat{\mathcal{Z}} = T\ IH(T^{-1}\mathcal{Z})$, where $T = U$ and $T^{-1} = U^T$

Reduction to order 1 at step k=2



Reduction to order 1 (2 generators)



Reach computation for continuous systems

Compact set representation and propagation by continuous flow is difficult, in general

- exact methods for classes of systems with simple dynamics

Reach computation for continuous systems

Compact set representation and propagation by continuous flow is difficult, in general

- exact methods for classes of systems with simple dynamics
- approximation methods for more general classes of systems
  - over/under-approximation via polyhedral, ellipsoidal sets, level set of some suitable function
  - asymptotic approximation methods based on gridding (scales badly, adaptive gridding)

Reach computation for continuous systems

Compact set representation and propagation by continuous flow is difficult, in general

- exact methods for classes of systems with simple dynamics
- approximation methods for more general classes of systems
  - over/under-approximation via polyhedral, ellipsoidal sets, level set of some suitable function
  - asymptotic approximation methods based on gridding (scales badly, adaptive gridding)
- statistical model checking

  → results in probability holding with a certain confidence

Reach computation for continuous systems

Compact set representation and propagation by continuous flow is difficult, in general

- exact methods for classes of systems with simple dynamics
- approximation methods for more general classes of systems
  - over/under-approximation via polyhedral, ellipsoidal sets, level set of some suitable function
  - asymptotic approximation methods based on gridding (scales badly, adaptive gridding)
- statistical model checking

  → results in probability holding with a certain confidence

Set-up

- discrete time system with nonlinear dynamics

$$x^+ = f(x)$$

- finite-horizon specifications as a collection of polyhedral safe sets described via intersection of half-spaces

$$Sp_t = \{x\colon H_{A,t}x \le H_{B,t}\}, \qquad t = 1, \dots, m$$

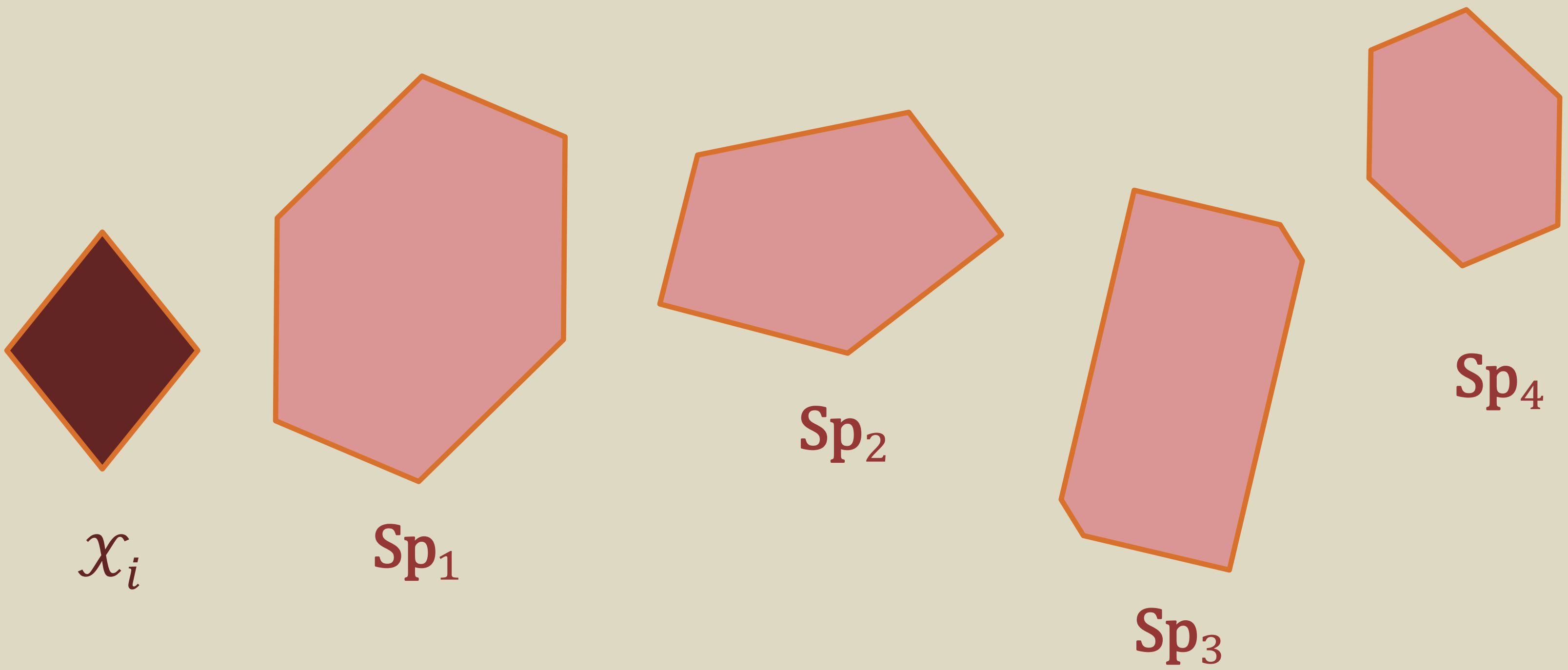
- an initial zonotopic set $\mathcal{X}_i$

Bounded safety problem

Check that the state x evolves within the spec sets, when the system is initialized from $\mathcal{X}_i$ and evolves according to $x^+ = f(x)$

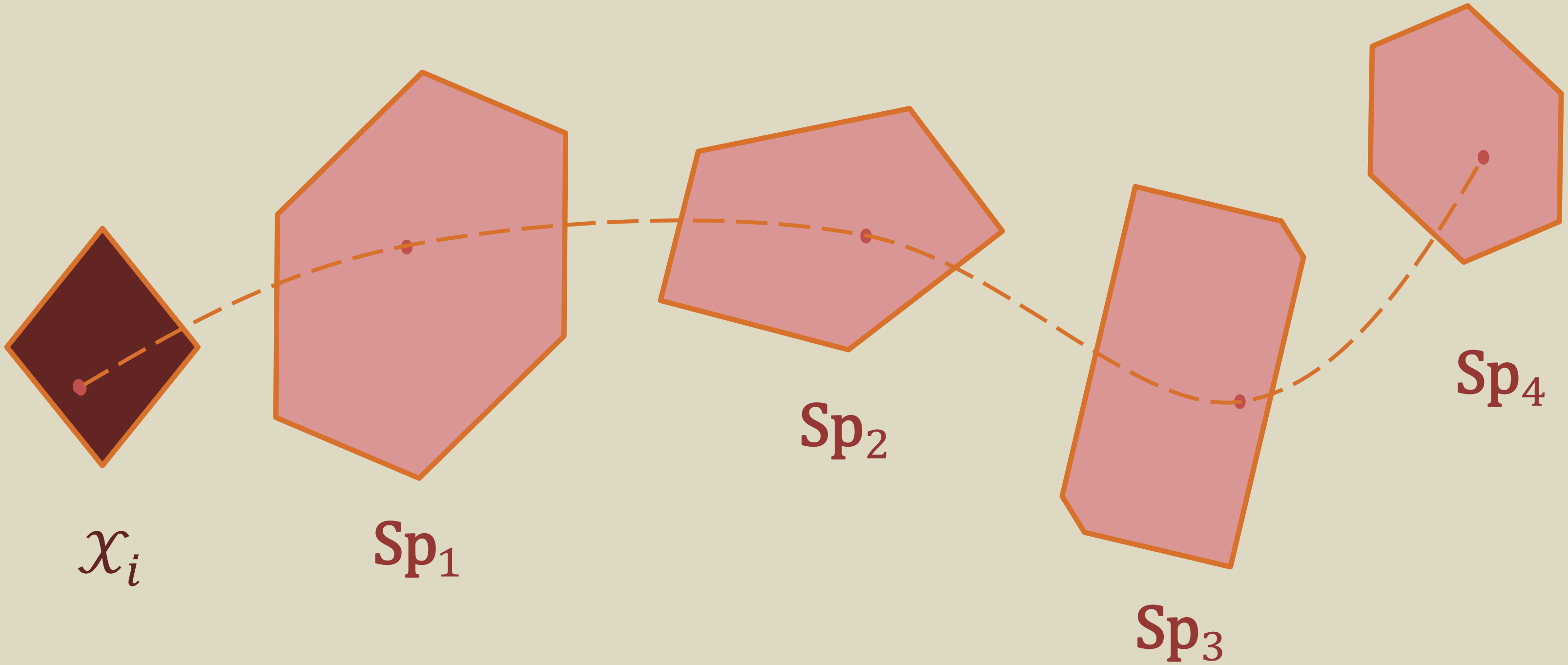
Bounded safety problem

Check that the state x evolves within the spec sets, when the system is initialized from $\mathcal{X}_i$ and evolves according to $x^+ = f(x)$



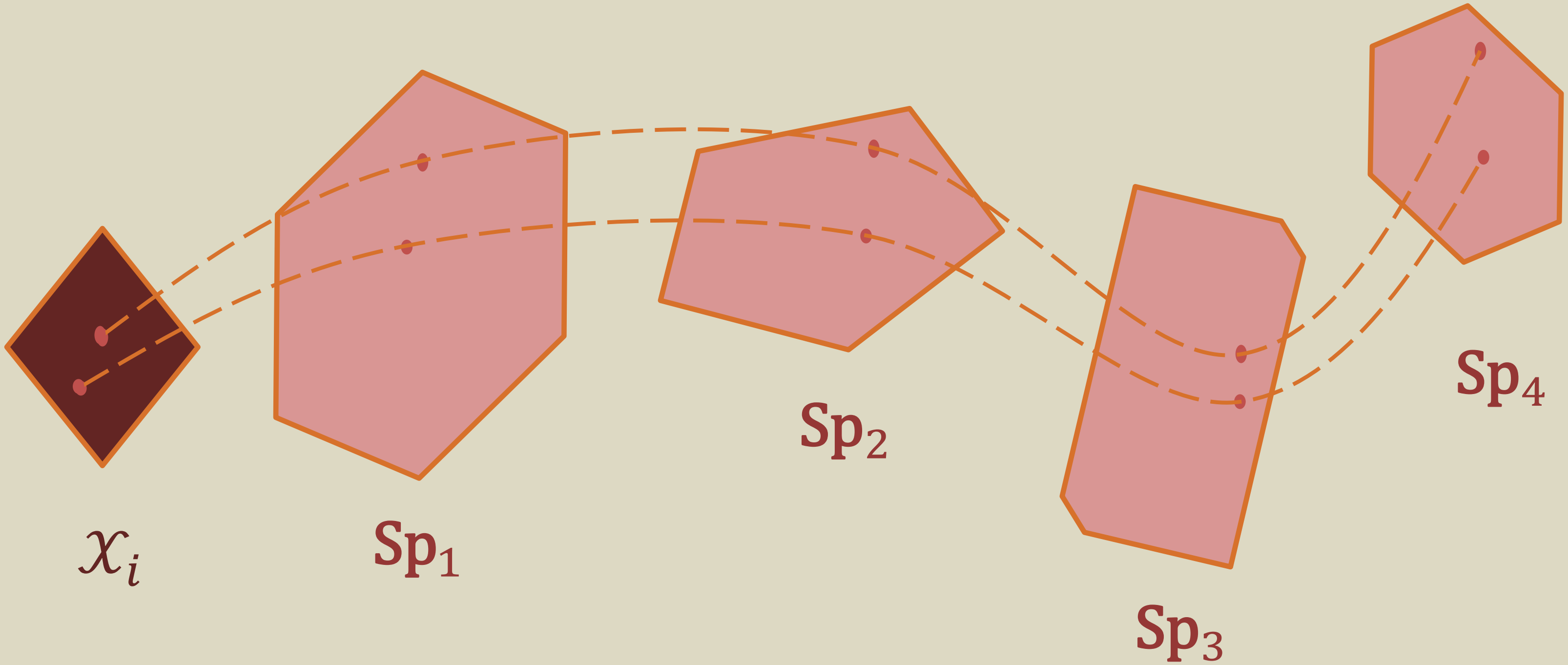
Bounded safety problem

Check that the state x evolves within the spec sets, when the system is initialized from $\mathcal{X}_i$ and evolves according to $x^+ = f(x)$

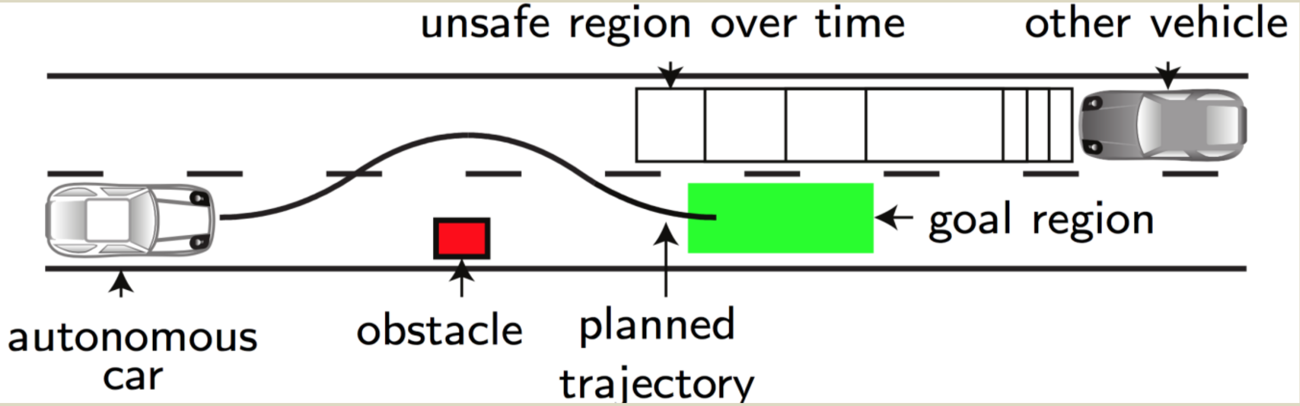


Bounded safety problem

Check that the state x evolves within the spec sets, when the system is initialized from $\mathcal{X}_i$ and evolves according to $x^+ = f(x)$



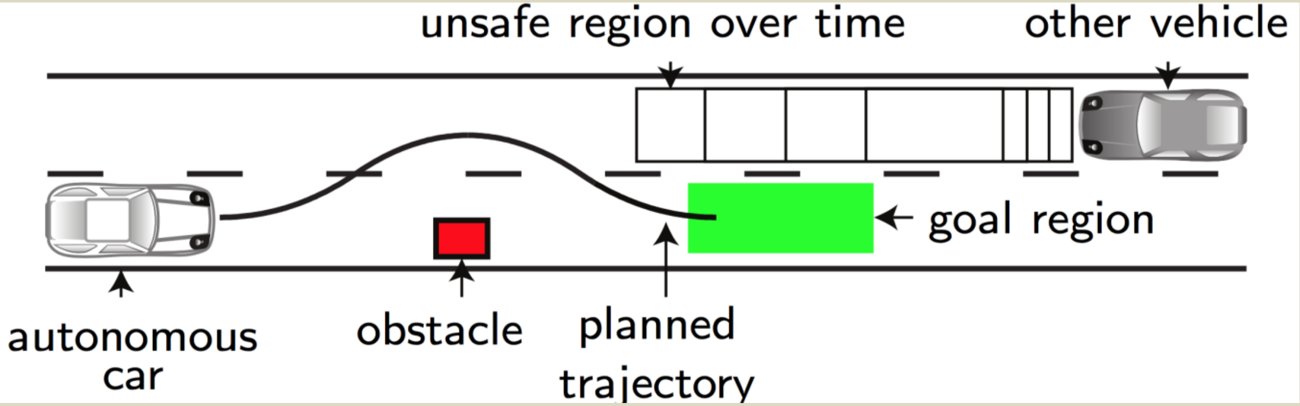
Motivating applications



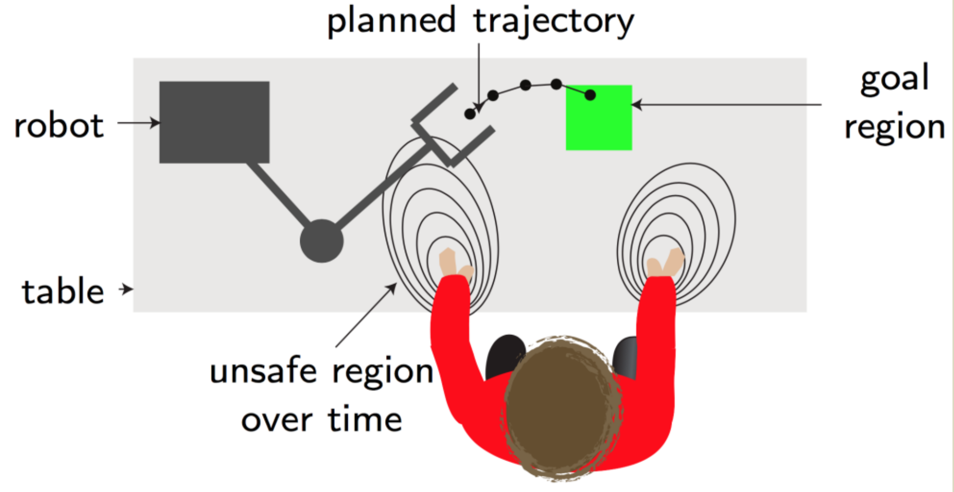
Autonomous driving

Credit: Matthias Althoff, TUM, Germany

Motivating applications



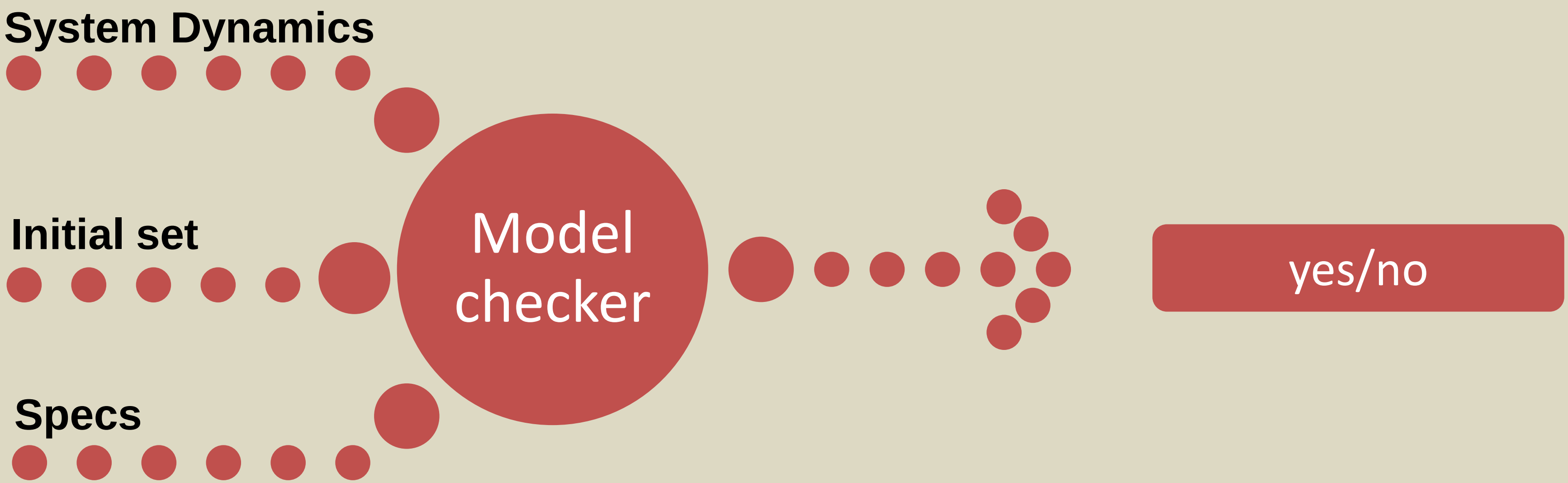
Autonomous driving



Human-robot interaction

Credit: Matthias Althoff, TUM, Germany

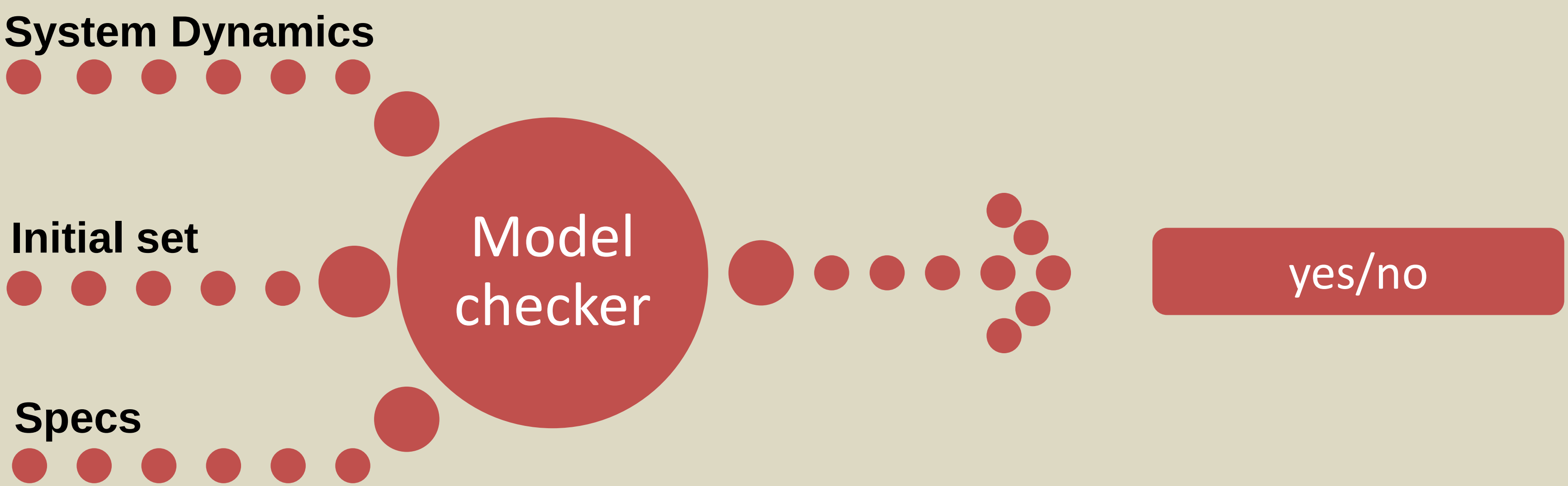
Model checking approach



Challenges:

- Continuous state
- Complex dynamics, Reachability
- Computational cost for online applications

Model checking approach



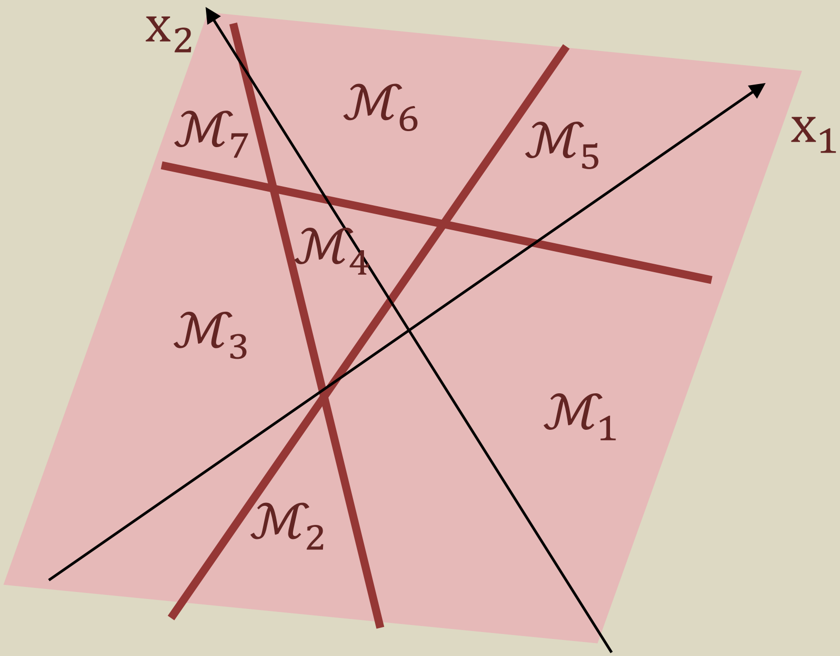
Challenges:

- Continuous state
- Complex dynamics, Reachability
- Computational cost for online applications

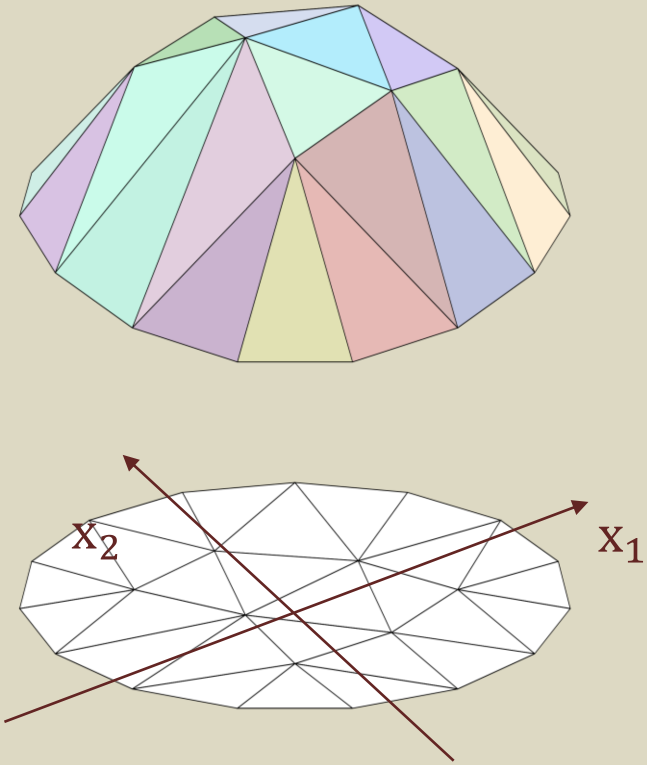
✓ unifying modeling framework
✓ easy to compute with sets

Unifying modeling framework

Piecewise affine (PWA) systems



PWA system modes



Piecewise affine function

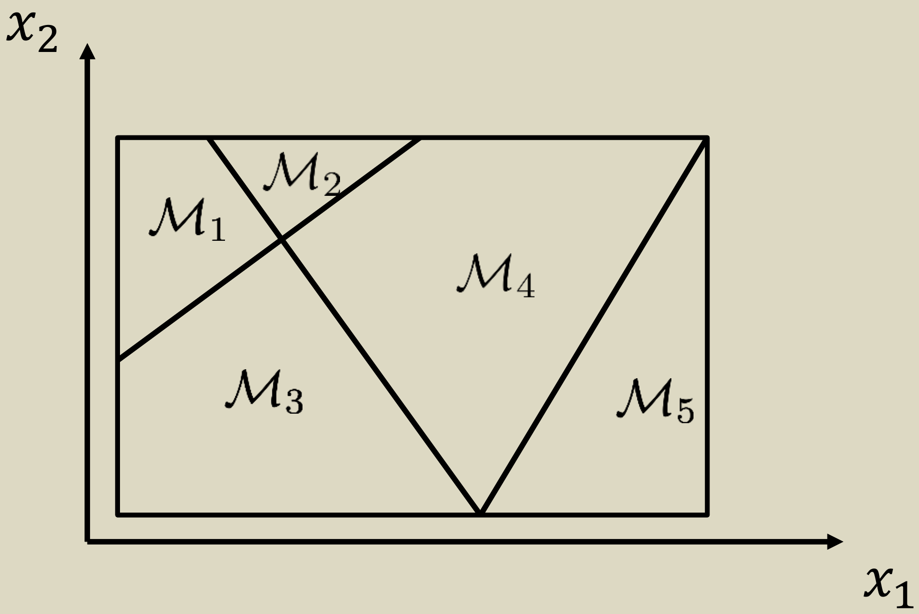
Unifying modeling framework

Why PWA systems?

Continuous dynamics is affine, which makes it easier to compute with sets

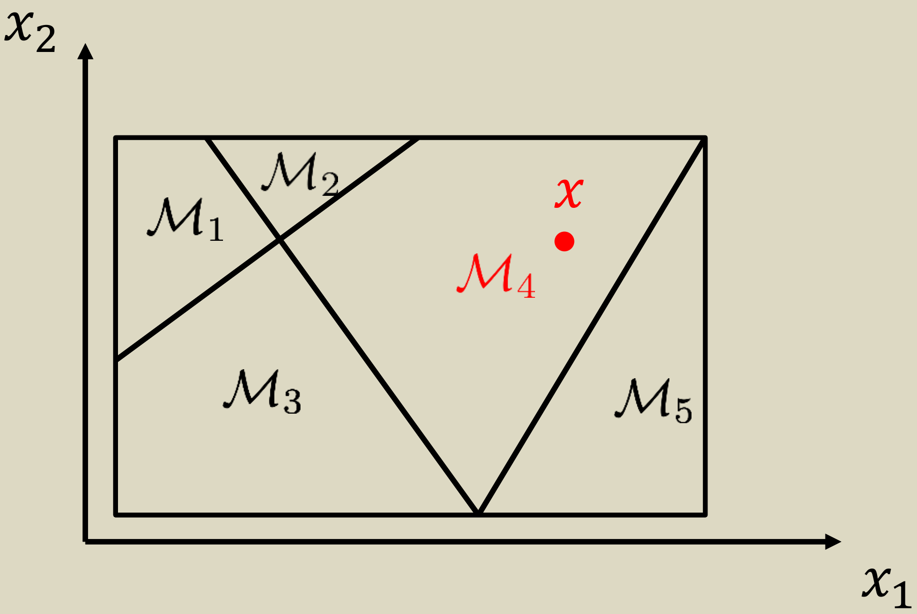
PWA system

$$x^+ = A^{(i)}x + f^{(i)} \text{ if } x \in \mathcal{M}_i \;\; i \in \{1,2,..,s\}$$



PWA system

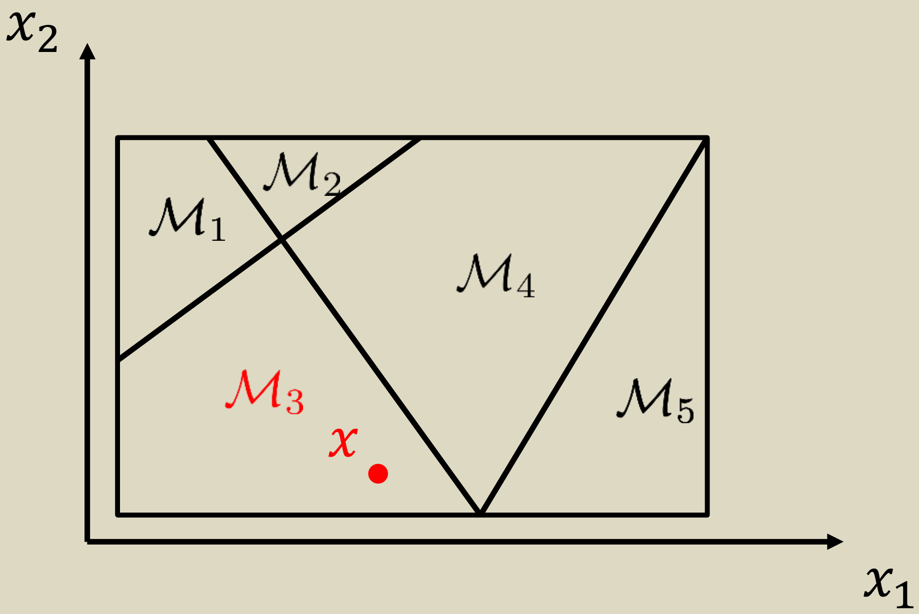
$$x^+ = A^{(4)}x + f^{(4)}$$



$\mathcal{M}_4$

PWA system

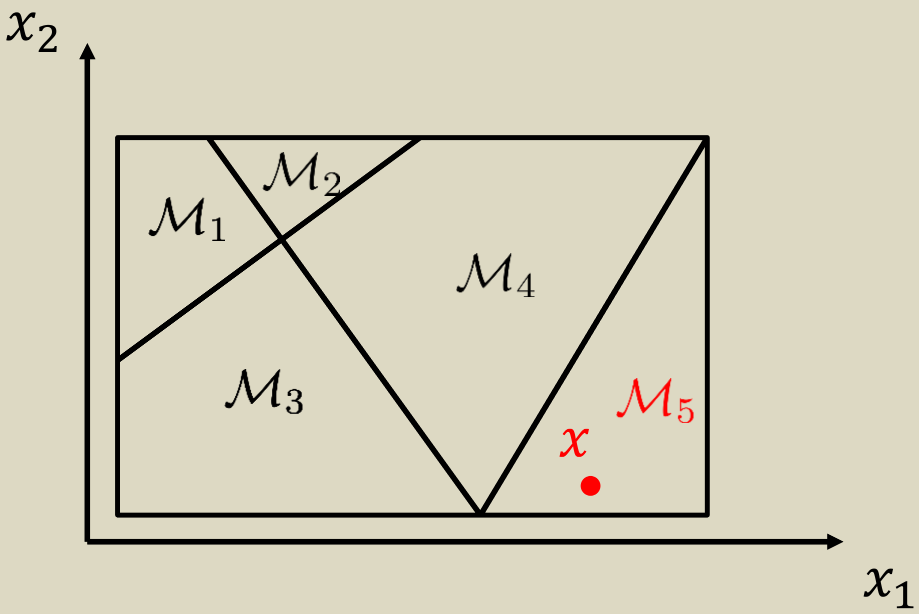
$$x^+ = A^{(3)}x + f^{(3)}$$



$\mathcal{M}_4, \ \mathcal{M}_3$

PWA system

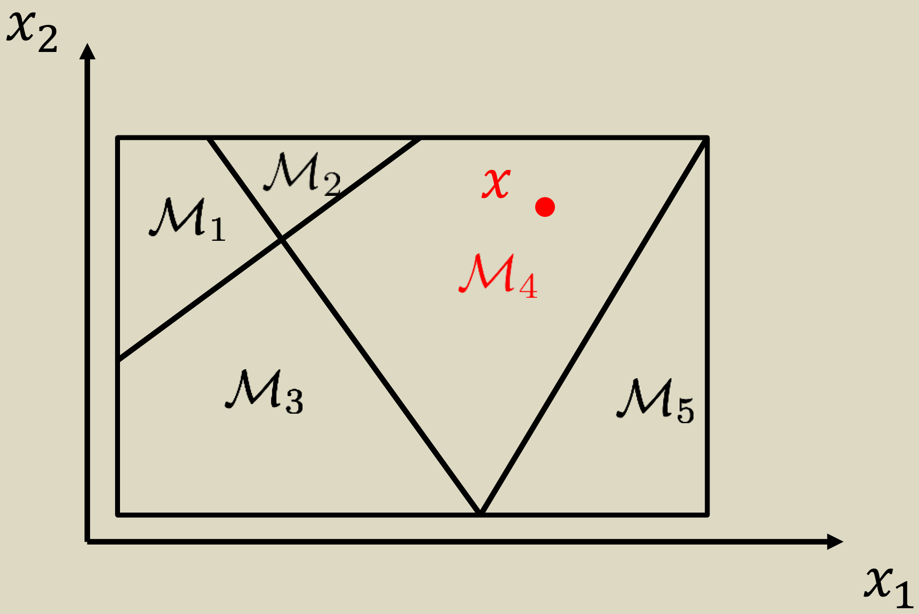
$$x^+ = A^{(5)}x + f^{(5)}$$



$\mathcal{M}_4,\ \mathcal{M}_3,\ \mathcal{M}_5$

PWA system

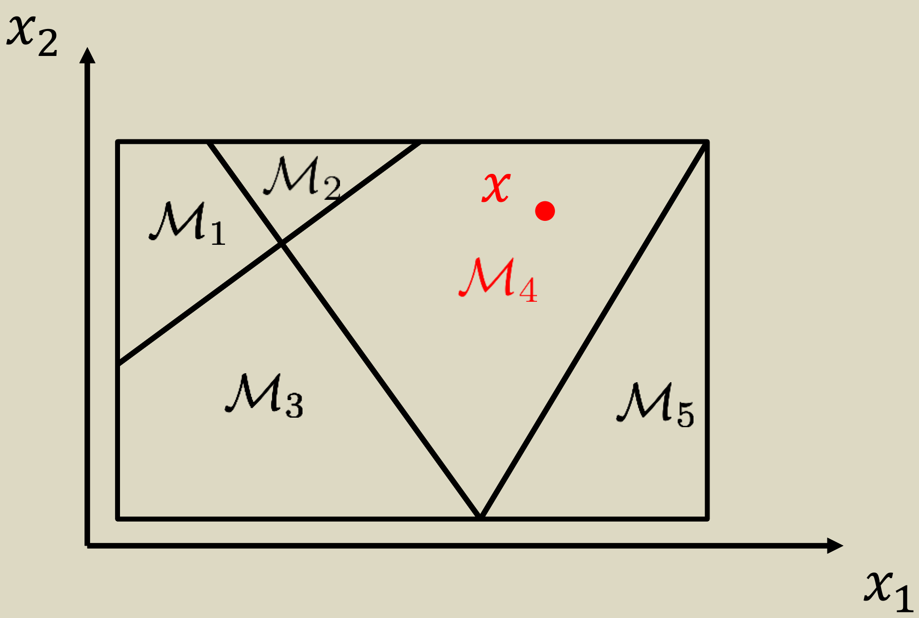
$$x^{+} = A^{(4)}x + f^{(4)}$$



$$\mathcal{M}_4,\ \mathcal{M}_3,\ \mathcal{M}_5,\ \mathcal{M}_4$$

PWA system

$$x^+ = A^{(4)}x + f^{(4)}$$

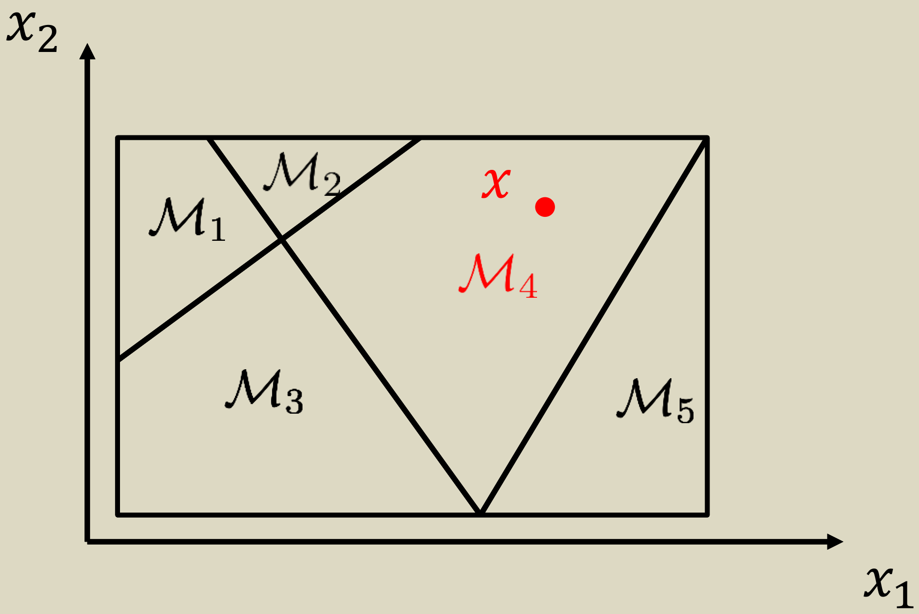


$$\{\mathcal{M}_4,\ \mathcal{M}_3,\ \mathcal{M}_5,\ \mathcal{M}_4, \ldots\}$$

Mode sequence

PWA system

$$x^+ = A^{(4)}x + f^{(4)}$$



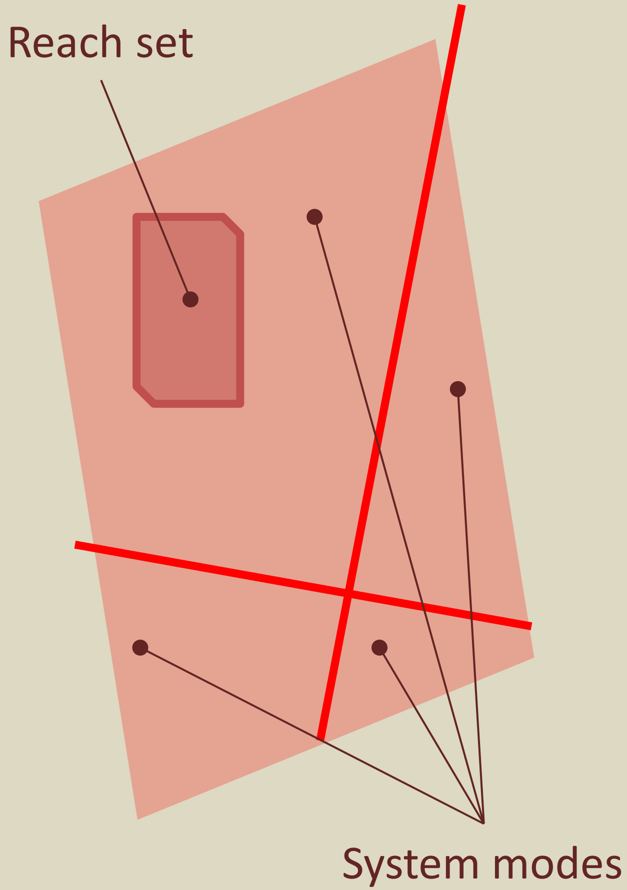
$$\{\mathcal{M}_4,\ \mathcal{M}_3,\ \mathcal{M}_5,\ \mathcal{M}_4, \ldots\}$$

Mode sequence

If the mode sequence were known → affine time-varying system

Reach sets propagation in a PWA system

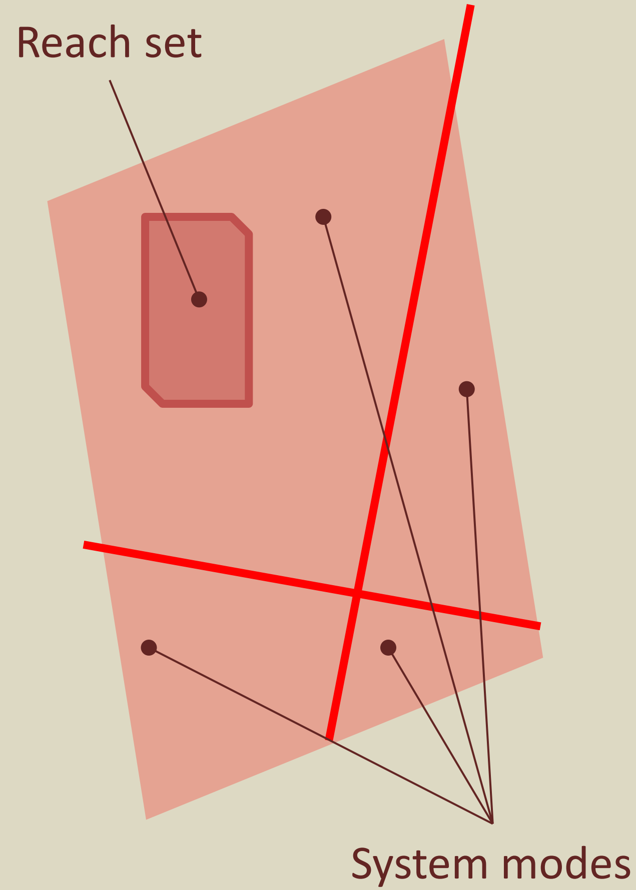
Case a) Reach sets propagate within the modes of the PWA system



Reach sets propagation in a PWA system

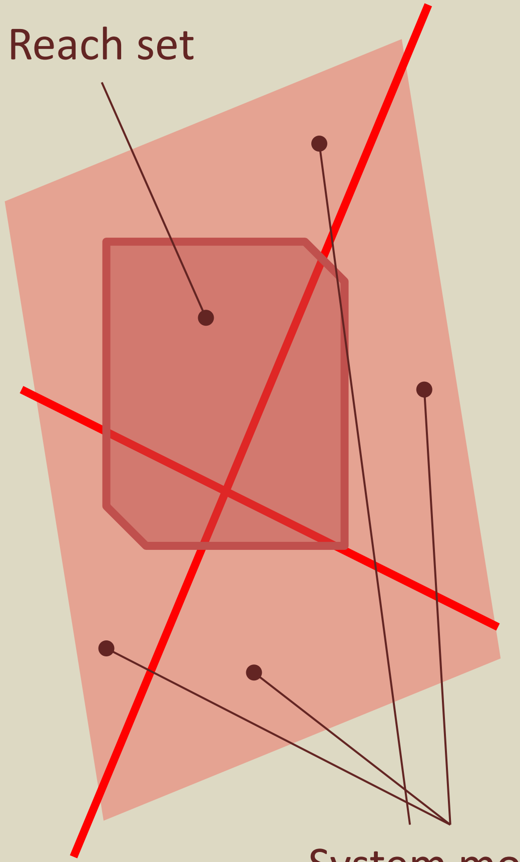
Case a) Reach sets propagate within the modes of the PWA system

- The PWA is reduced to a time-varying affine system
- Reach sets computations are exact if the initial set is a zonotope



Reach sets propagation in a PWA system

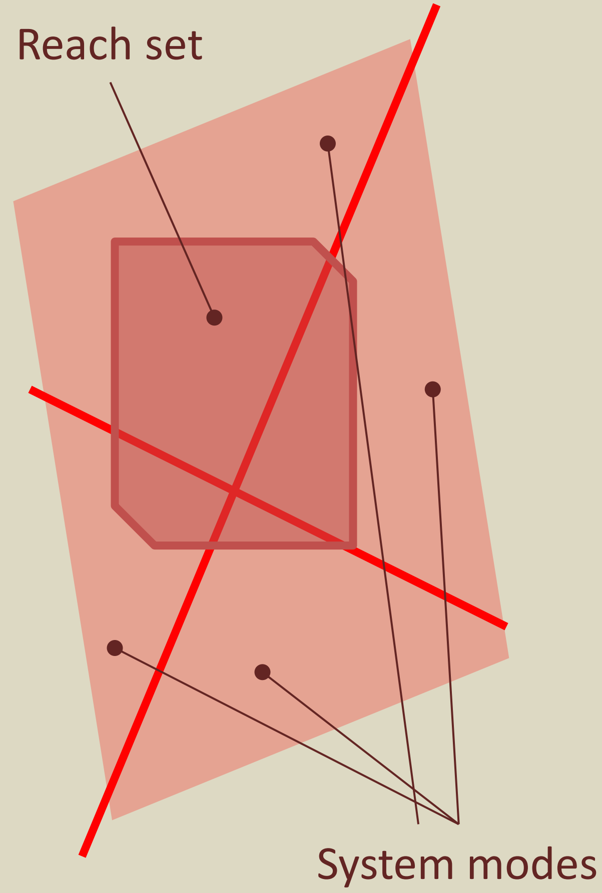
Case b) A reach set splits over multiple modes



Reach sets propagation in a PWA system

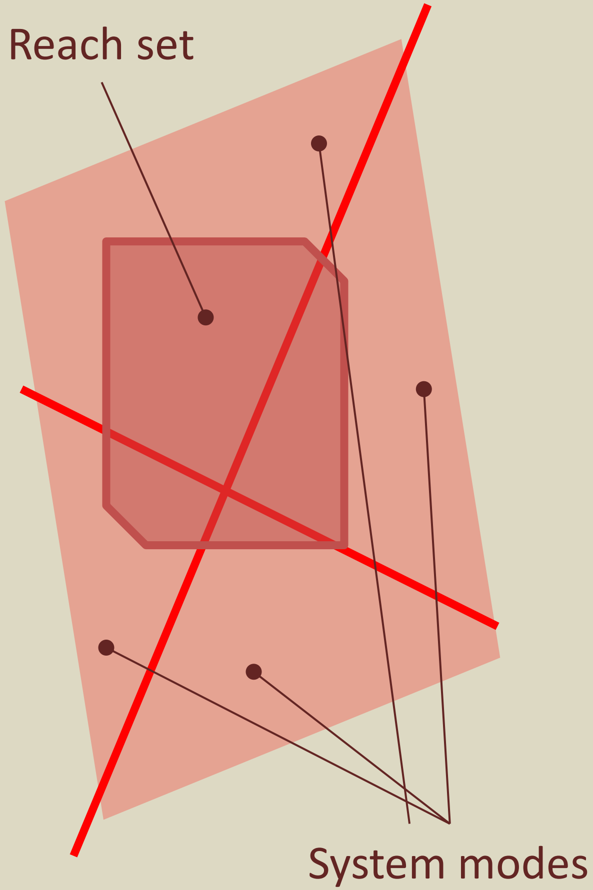
Case b) A reach set splits over multiple modes

- The subsets of the split set are not zonotopes in general
- Different dynamics are associated with different subsets



Reach set splitting among modes

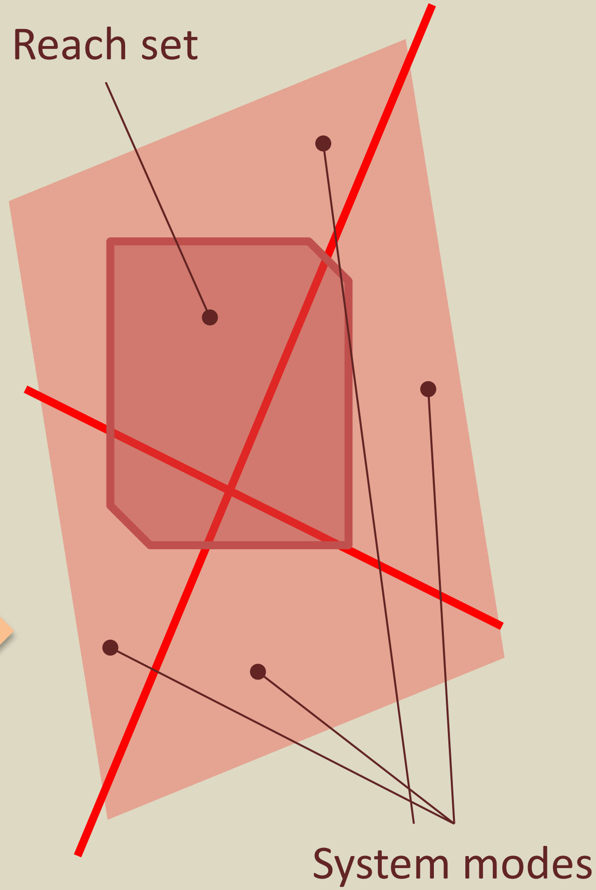
Case b) A reach set splits over multiple modes



Reach set splitting among modes

Case b) A reach set splits over multiple modes

Outer zonotopic approximation of each part of the split reach set

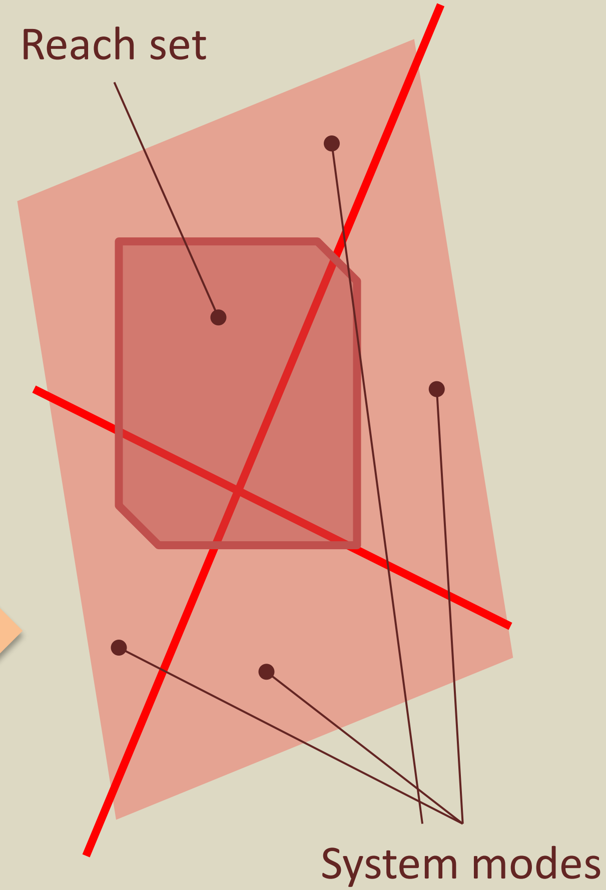


Reach set splitting among modes

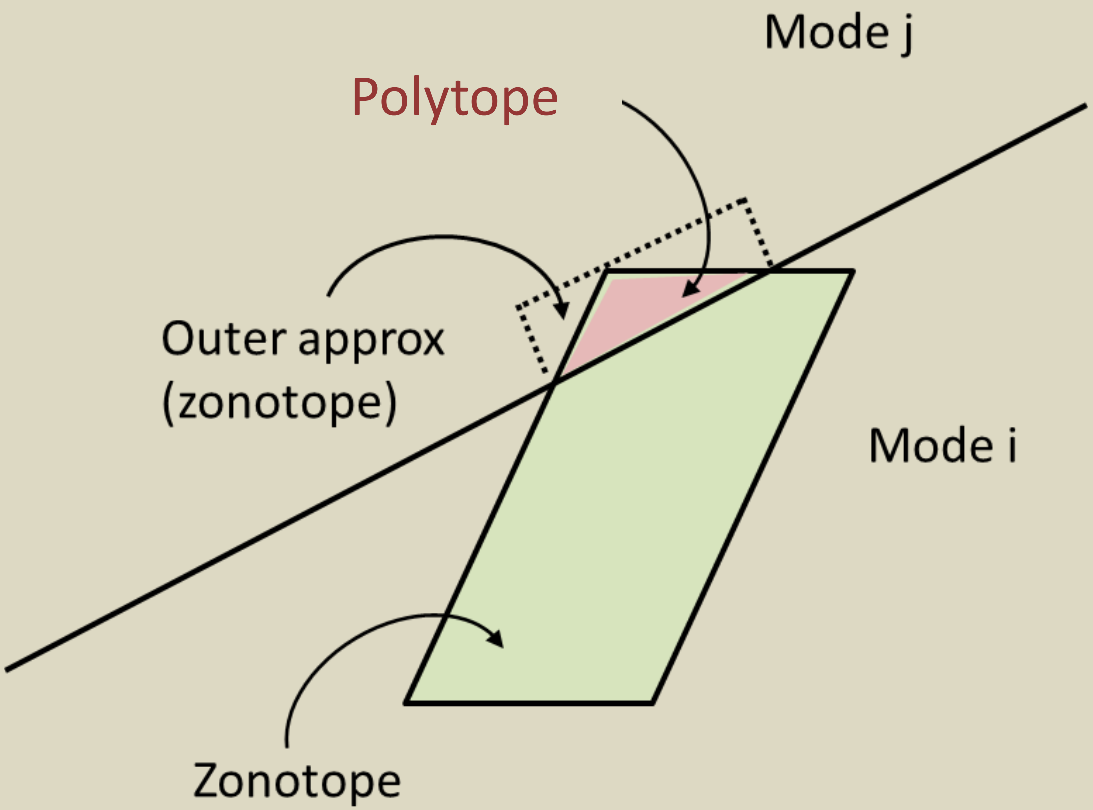
Case b) A reach set splits over multiple modes

Outer zonotopic approximation of each part of the split reach set

Propagation of each zonotopic fragment according to the corresponding mode dynamics

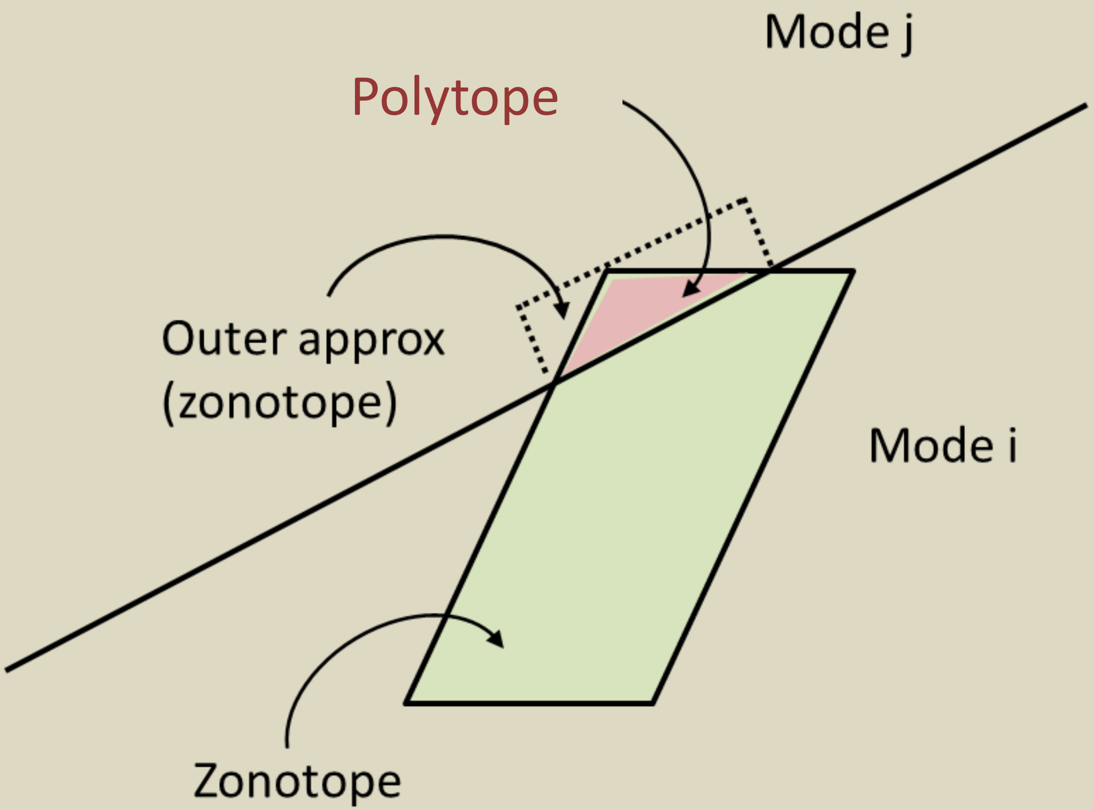


Outer approximation of polytopic fragments



Fragments are polytopes, not zonotopes

Outer approximation of polytopic fragments



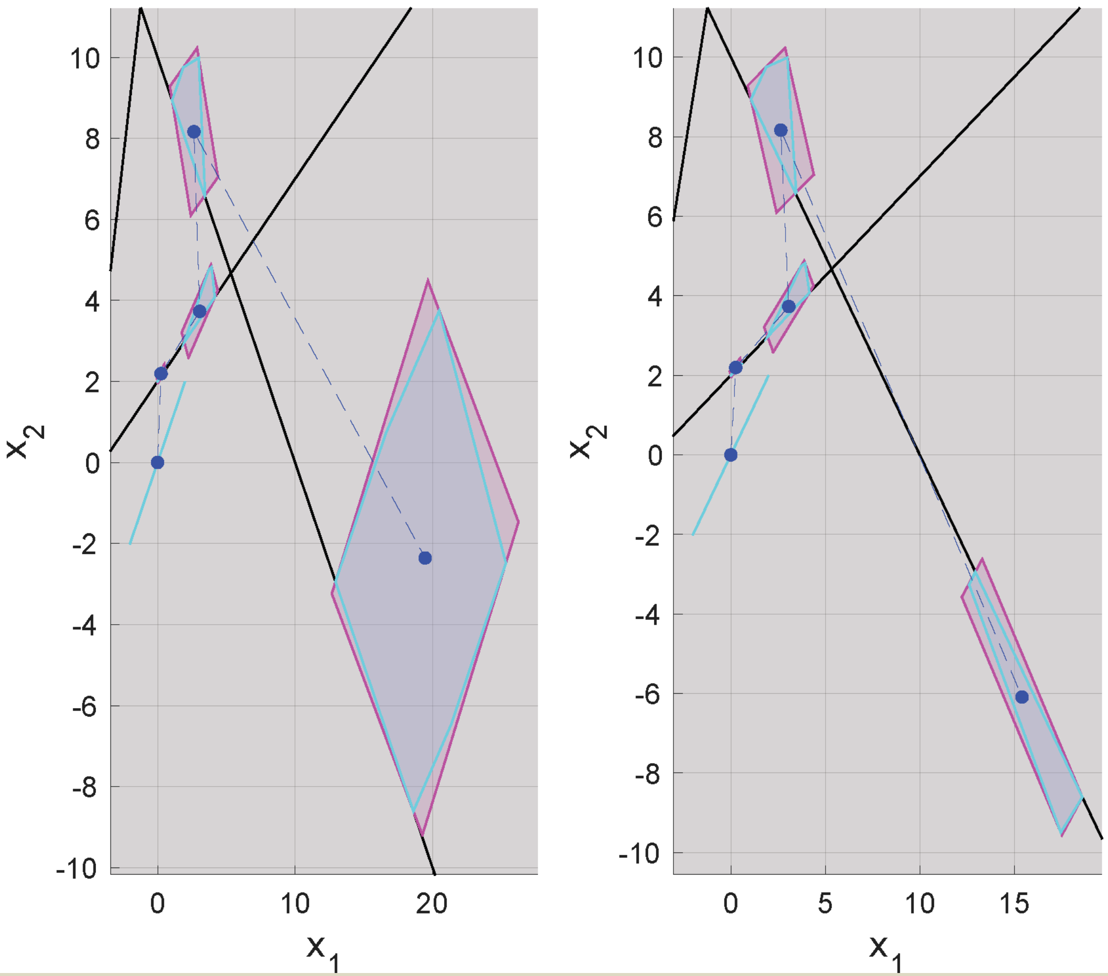
Fragments are polytopes, not zonotopes

Use the transformation method

$$T\ IH(T^{-1}\mathcal{P})$$

with transformation matrix T computed
1) via PCA on the (unbiased) vertices or
2) by fitting the largest ellipsoidal set and using its axes to define T

Branching in reach set propagation



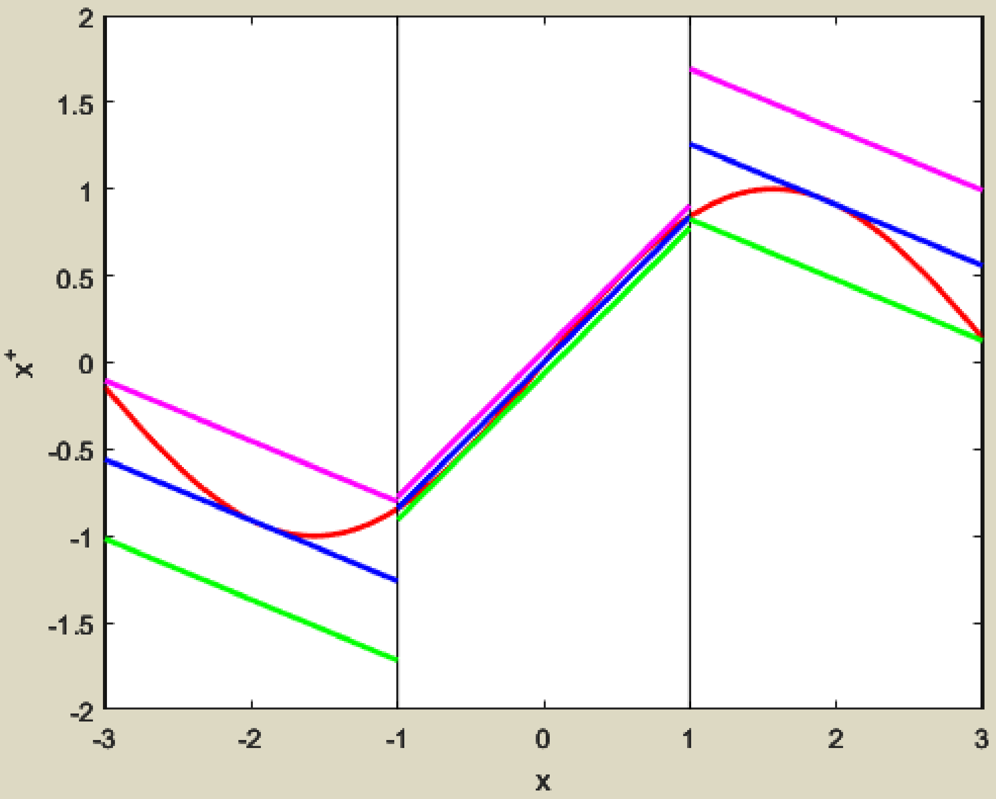
Unifying modeling framework

Can approximate nonlinear (smooth) dynamics with arbitrary accuracy

Why PWA systems?

Continuous dynamics is affine, which makes it easier to compute with sets

Hybridization: method



Given a nonlinear function $x^+ = f(x)$

- Divide function domain into a rectangular grid
- Affinely approximate the function on each grid element
- Compute error bound

Hybridization: method

For each element $X^{(i)}$ of the grid, we introduce

$$x^+ = g^{(i)}(x) + w$$

where $g^{(i)}(x) = A^{(i)}x + f^{(i)}$

and determine $W^{(i)}$ so as to satisfy the following property

$$\forall x \in X^{(i)} \; \exists w_x \in W^{(i)} \text{ such that } f(x) = g^{(i)}(x) + w_x$$

Hybridization: method

For each element $X^{(i)}$ of the grid, we introduce

$$x^+ = g^{(i)}(x) + w$$

where $g^{(i)}(x) = A^{(i)}x + f^{(i)}$

and determine $W^{(i)}$ so as to satisfy the following property

$$\forall x \in X^{(i)} \; \exists w_x \in W^{(i)} \;\text{ such that }\; f(x) = g^{(i)}(x) + w_x$$

Trace conformance is satisfied

[every admissible state trajectory of the original system is also an admissible trajectory for its hybridization]

Hybridization: method

For each element $X^{(i)}$ of the grid, we introduce

$$x^+ = g^{(i)}(x) + w$$

where $g^{(i)}(x) = A^{(i)}x + f^{(i)}$

and determine $W^{(i)}$ so as to satisfy the following property

$$\forall x \in X^{(i)} \ \exists w_x \in W^{(i)} \ \text{ such that } f(x) = g^{(i)}(x) + w_x$$

Reach set conformance is satisfied

[the reachability sets of the hybridization contain those of the original system]

Hybridization: method

For each element $X^{(i)}$ of the grid, we introduce

$$x^+ = g^{(i)}(x) + w$$

where $g^{(i)}(x) = A^{(i)}x + f^{(i)}$

and determine $W^{(i)}$ so as to satisfy the following property

$$\forall x \in X^{(i)} \ \exists w_x \in W^{(i)} \ \text{ such that } f(x) = g^{(i)}(x) + w_x$$

Reach set conformance is satisfied

$$\mathcal{R}(f_{PWA}) \supseteq \mathcal{R}(f)$$

where $f_{PWA}(x) = A^{(i)}x + f^{(i)} + w, \ w \in W^{(i)}$

Hybridization: method

For each element $X^{(i)}$ of the grid, we introduce

$$x^+ = g^{(i)}(x) + w$$

where $g^{(i)}(x) = A^{(i)}x + f^{(i)}$

and determine $W^{(i)}$ so as to satisfy the following property

$$\forall x \in X^{(i)} \ \exists w_x \in W^{(i)} \ \text{ such that } f(x) = g^{(i)}(x) + w_x$$

If the PWA hybridization satisfies the safety property, robustly with respect to the additive disturbance (robust safety), then the nonlinear system is safe

Hybridization: method

For each element $X^{(i)}$ of the grid, we introduce

$$x^+ = g^{(i)}(x) + w$$

where $g^{(i)}(x) = A^{(i)}x + f^{(i)}$

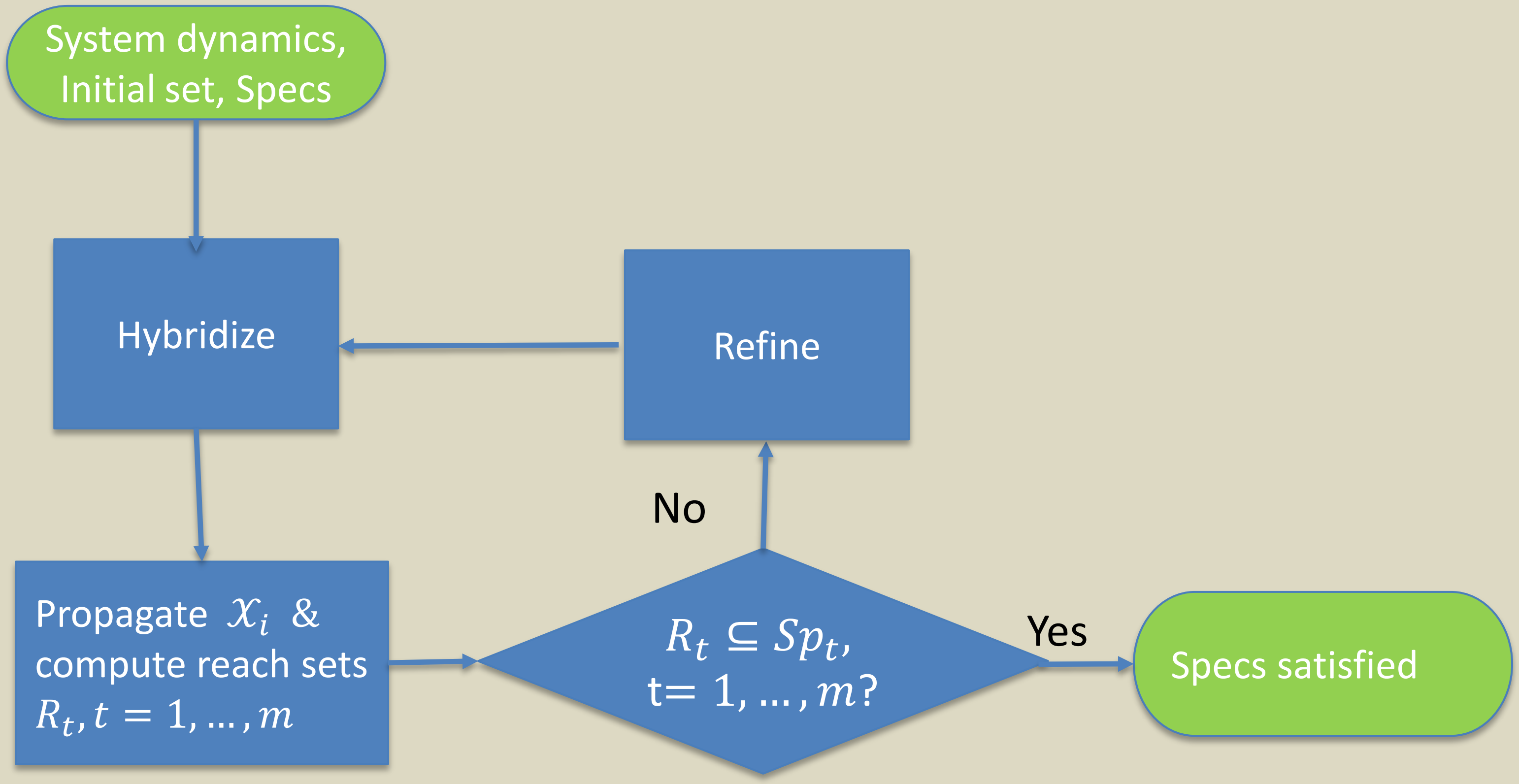
and determine $W^{(i)}$ so as to satisfy the following property

$$\forall x \in X^{(i)} \; \exists w_x \in W^{(i)} \text{ such that } f(x) = g^{(i)}(x) + w_x$$

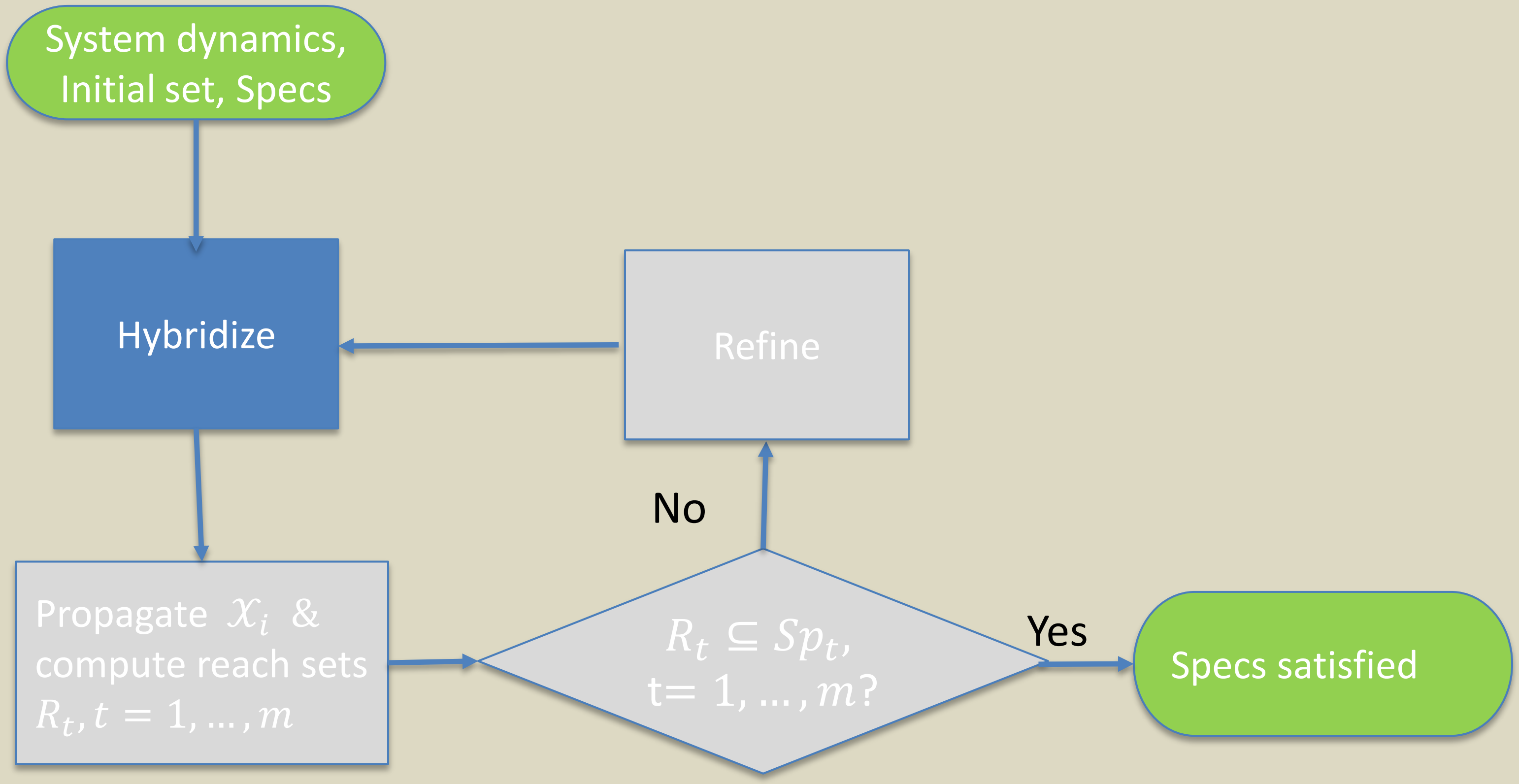
If the PWA hybridization satisfies the safety property, robustly with respect to the additive disturbance (robust safety), then the nonlinear system is safe

if not, then refine and try again

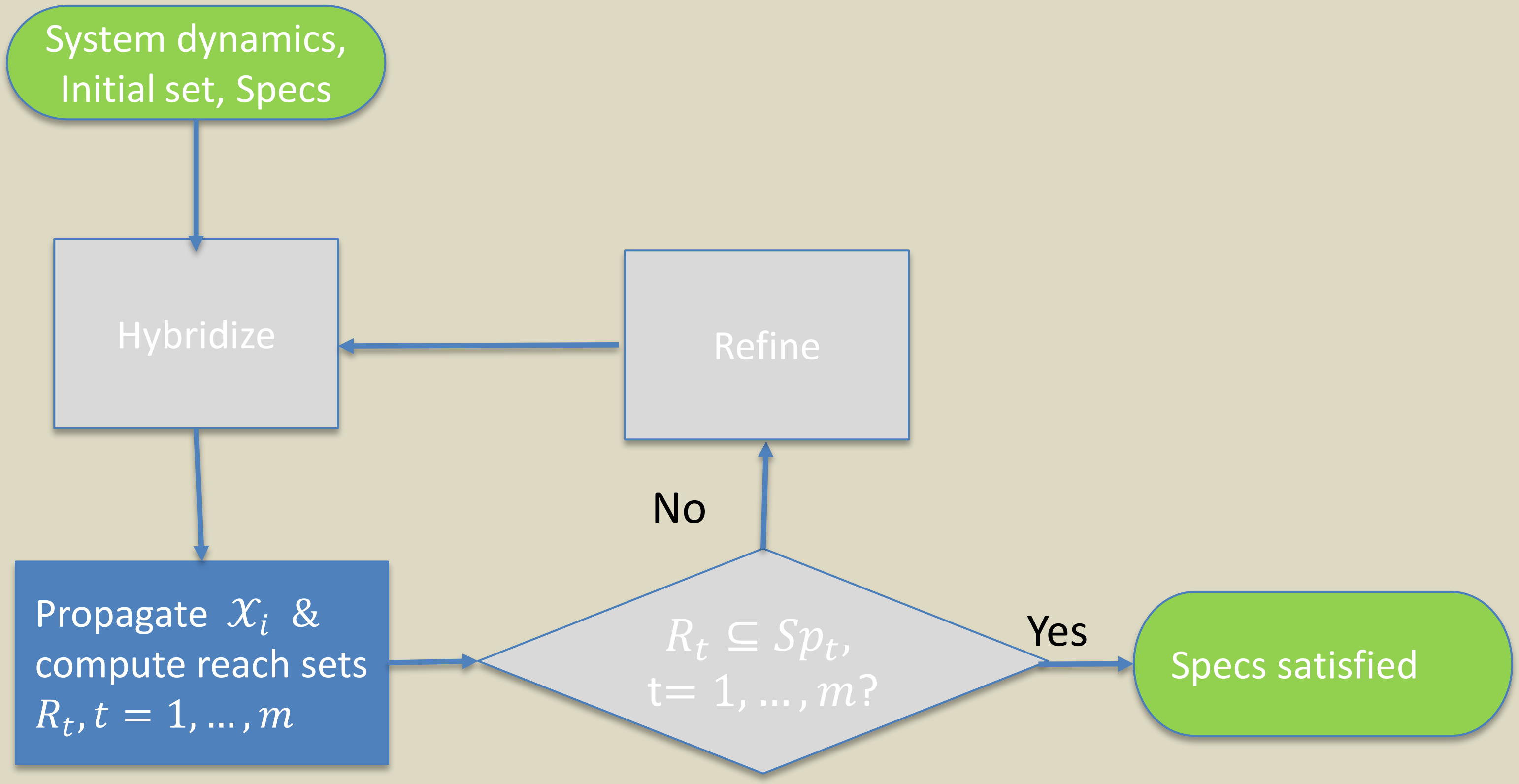
Model checker



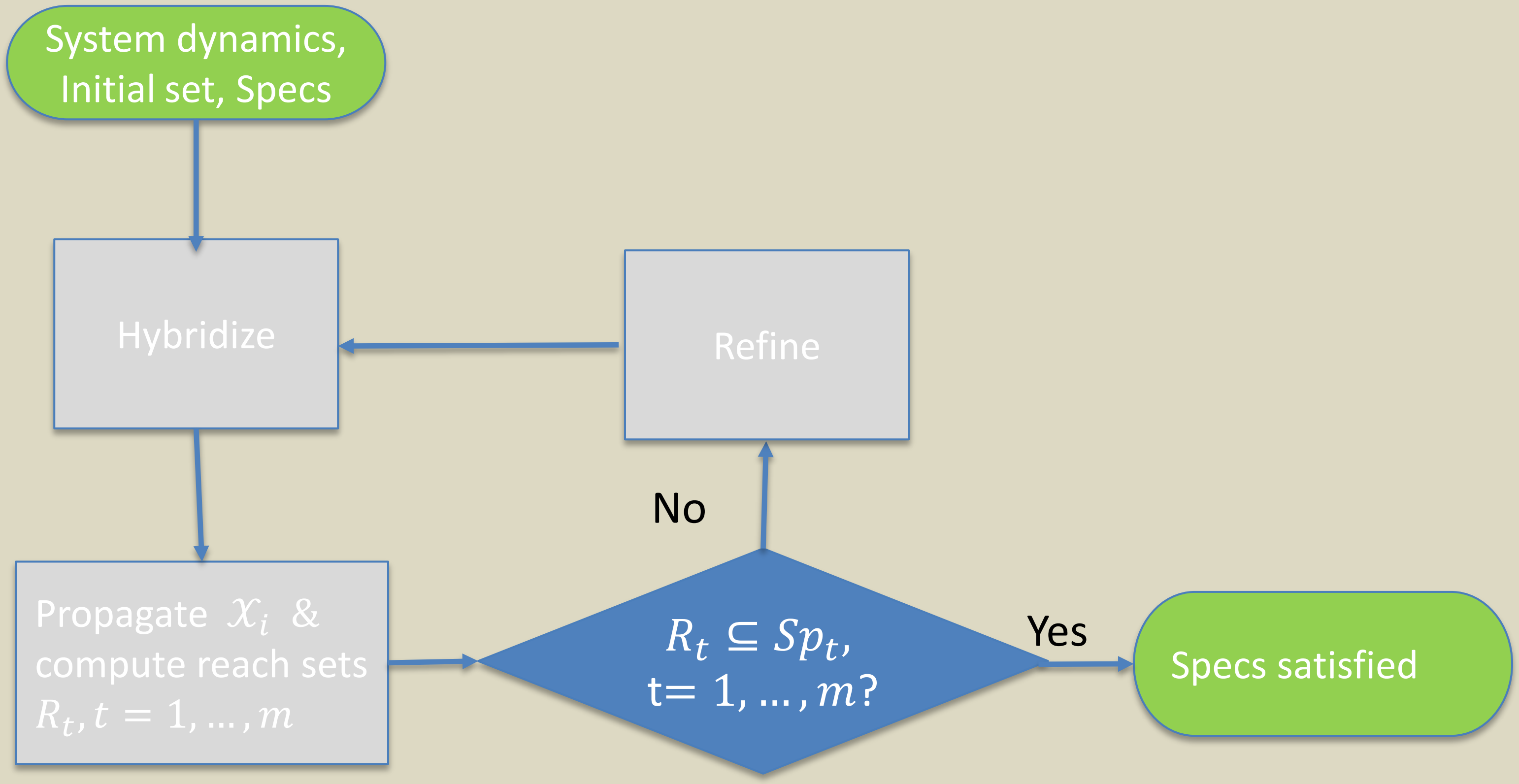
Model checker



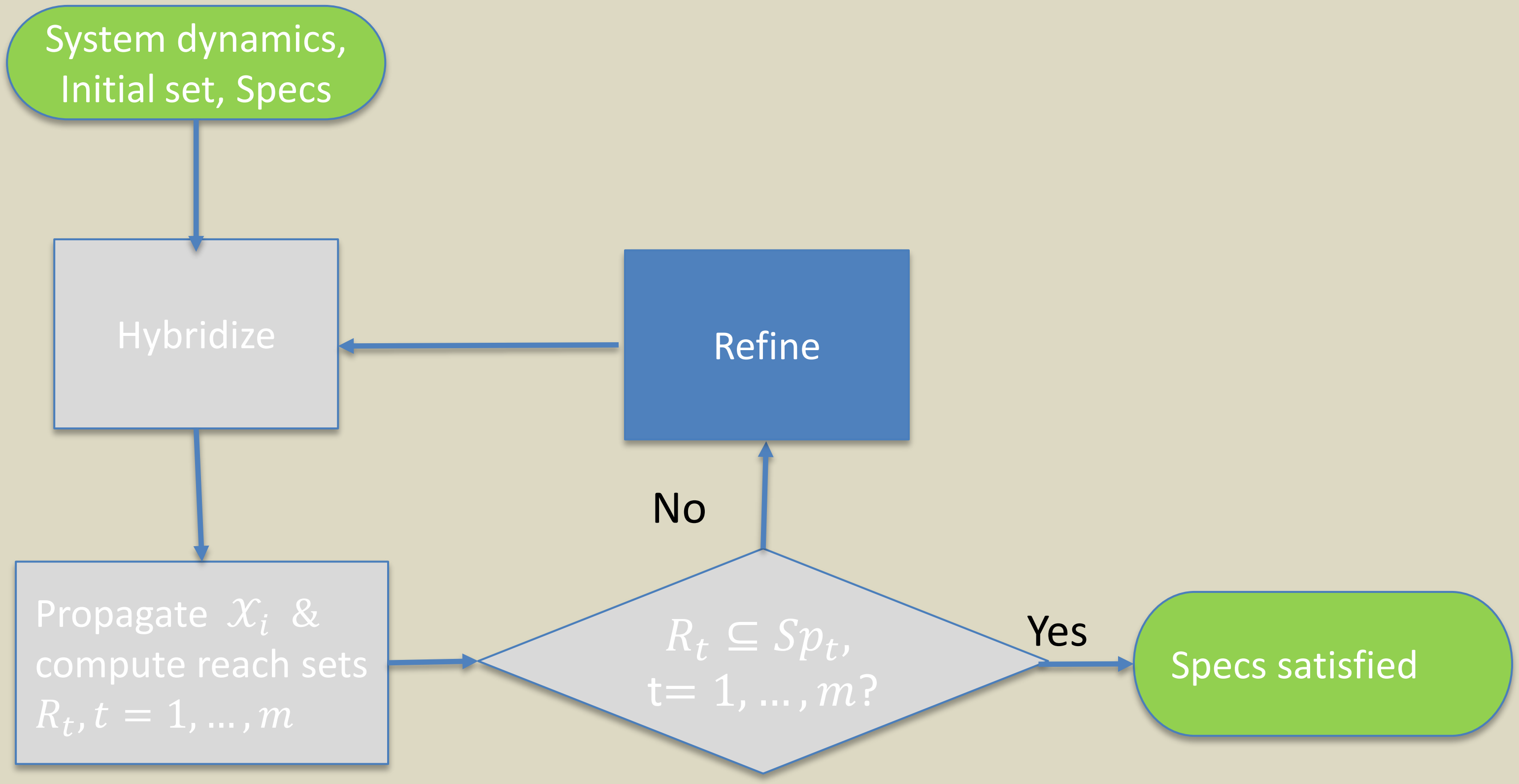
Model checker



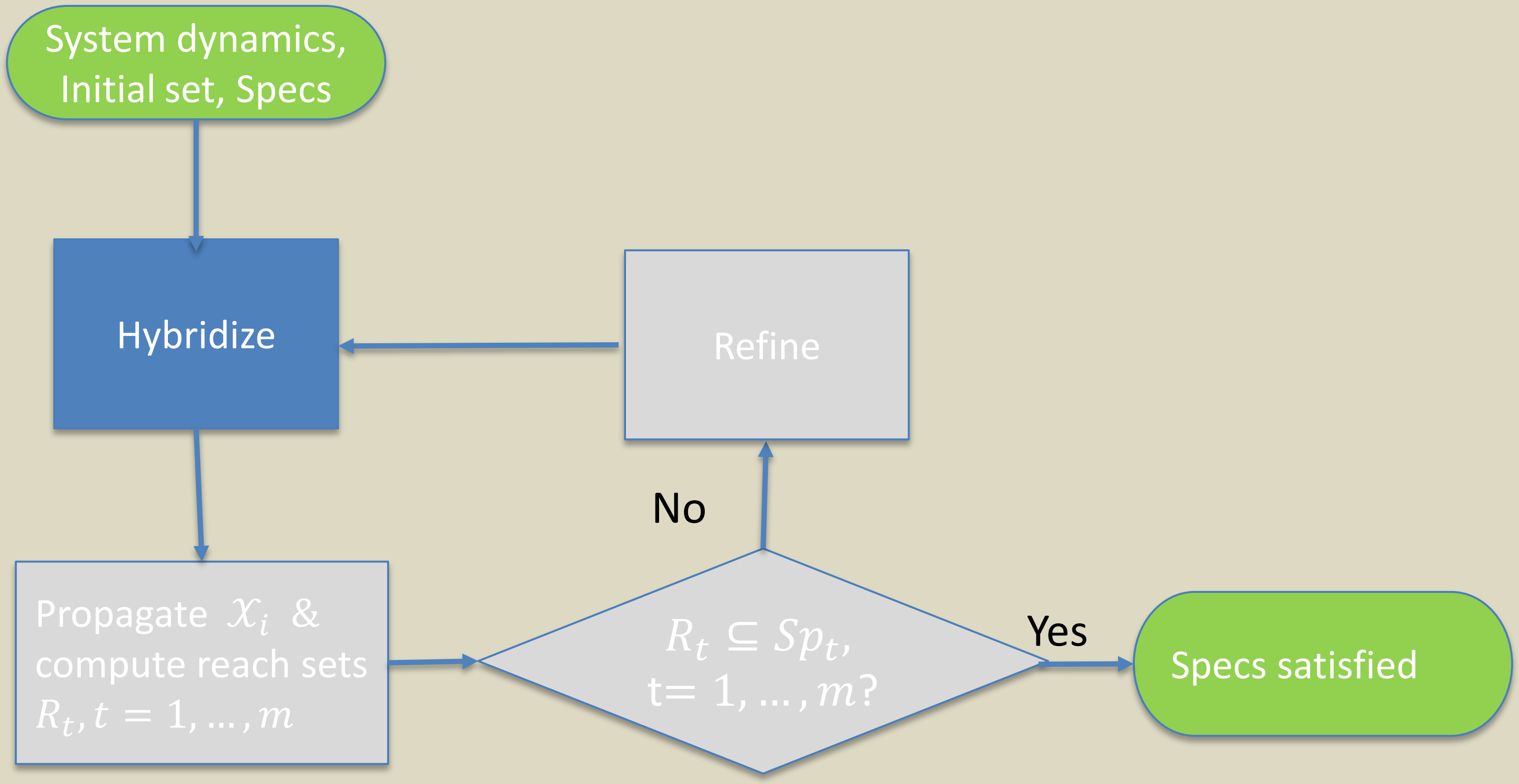
Model checker



Model checker



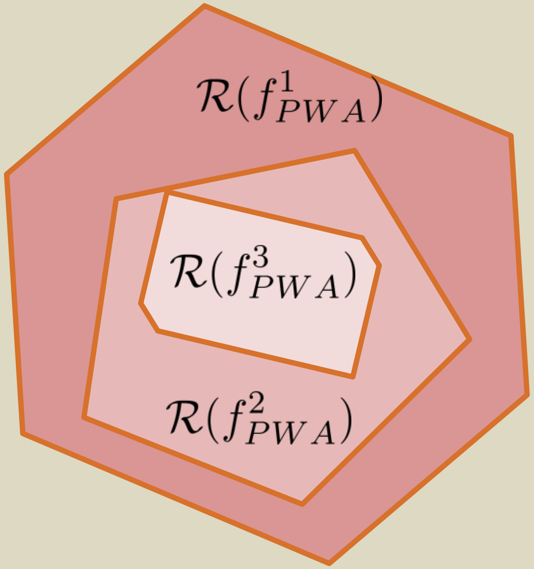
Model checker



Refinement inclusion

reach sets of an abstraction contain reach sets of its refinements

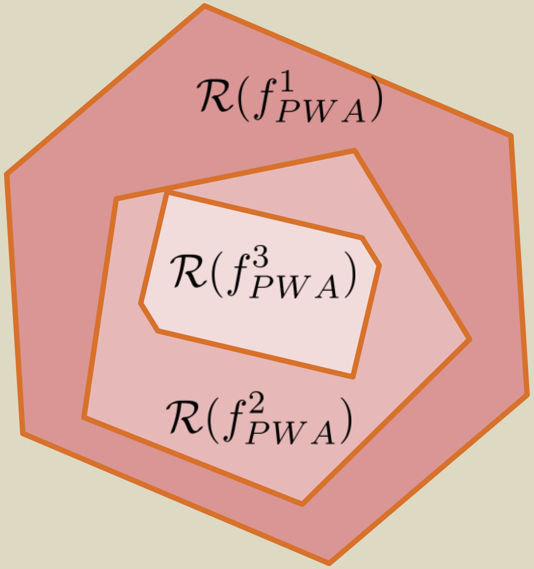
$$\mathcal{R}(f^1_{PWA}) \supseteq \mathcal{R}(f^2_{PWA}) \supseteq \mathcal{R}(f^3_{PWA}) \supseteq \cdots \supseteq \mathcal{R}(f)$$



Refinement inclusion

reach sets of an abstraction contain reach sets of its refinements

$$\mathcal{R}(f^1_{PWA}) \supseteq \mathcal{R}(f^2_{PWA}) \supseteq \mathcal{R}(f^3_{PWA}) \supseteq \cdots \supseteq \mathcal{R}(f)$$

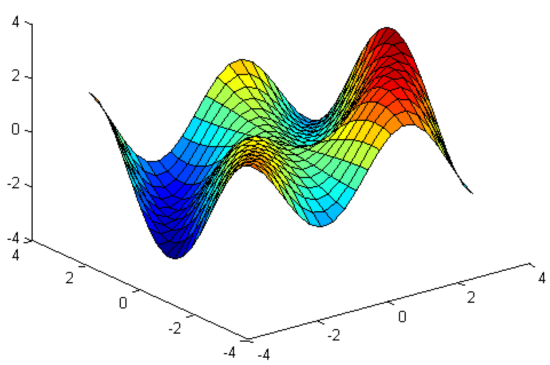


no additional spurious behaviors introduced and a progressively tighter abstraction of the original system obtained → iterative approach is sound

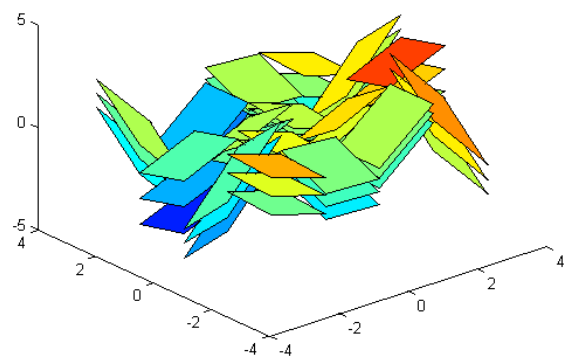
Hybridization: Refinement Inclusion

Orthogonal projection of the non-linear function in the space of PWA functions

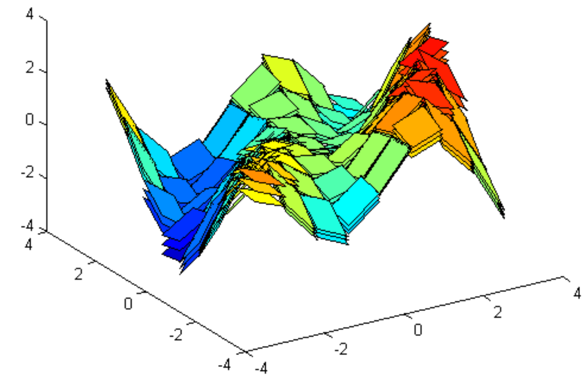
Non linear



First iteration



Second iteration

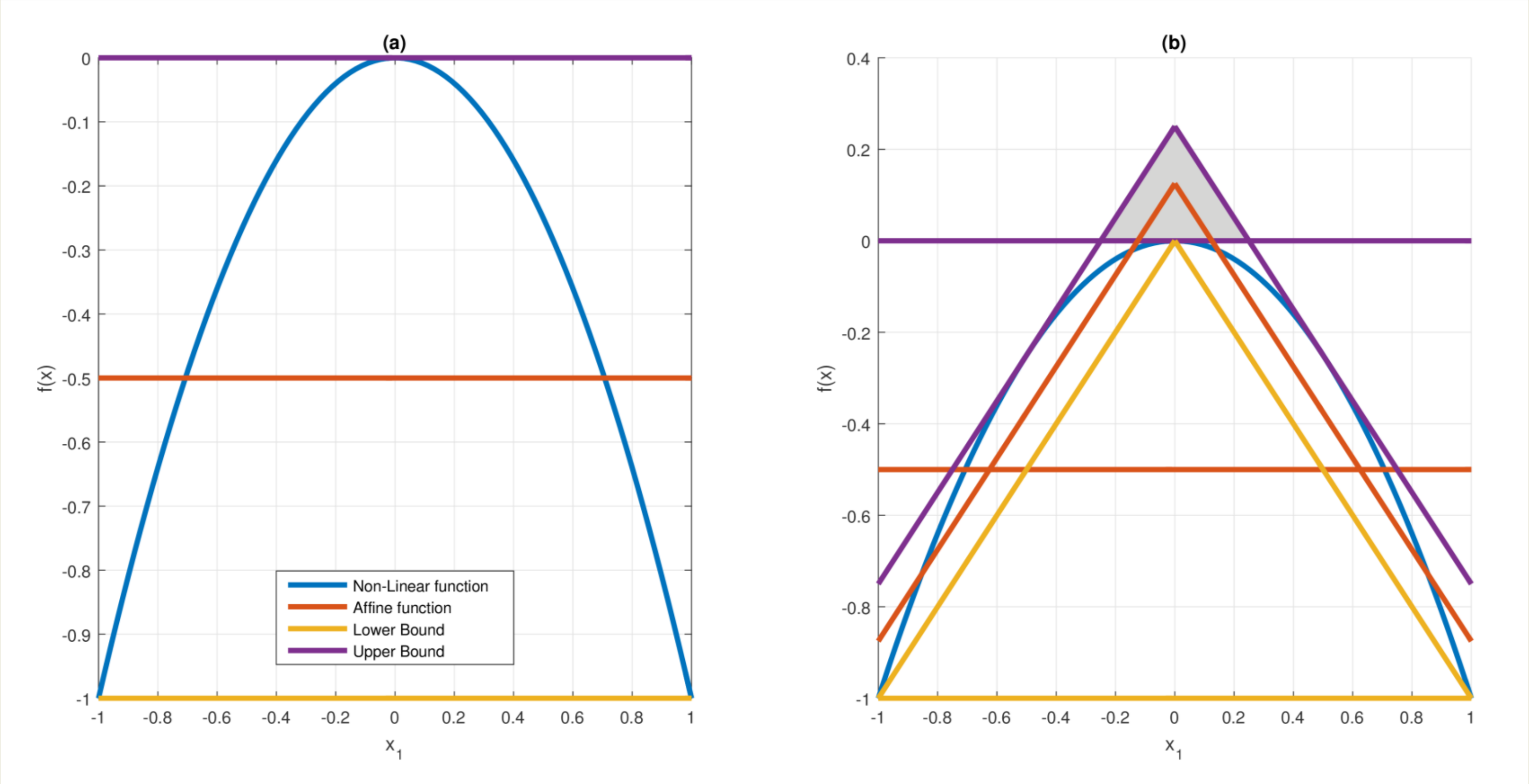


Approximation error: smallest

Reachable set reduction: not guaranteed

Hybridization: Refinement Inclusion

counter-example



Hybridization: Affine Approximation

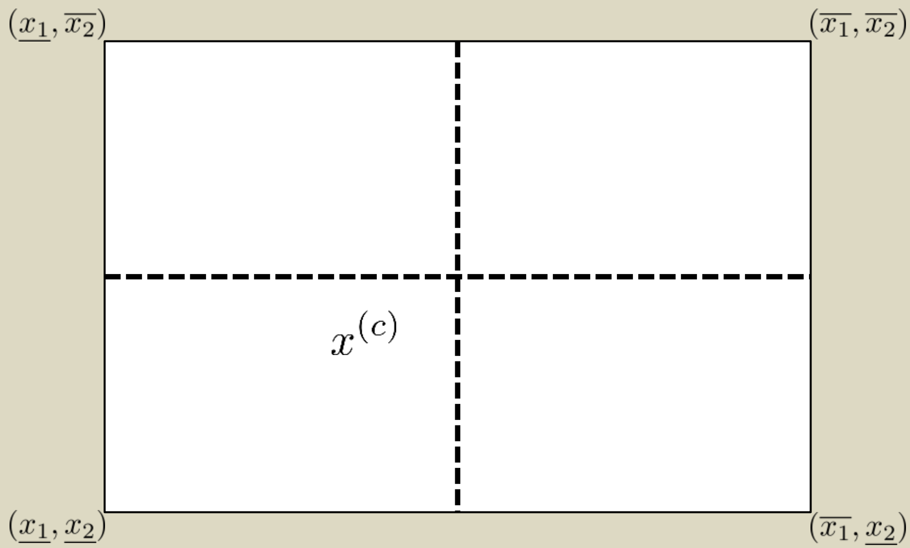
Scalar valued function

$$f(x) : \Re^2 \rightarrow \Re$$

find $f_{PWA}(x) = A^{(i)}x + f^{(i)} + w,\ w \in W^{(i)},\ x \in X^{(i)}$

that satisfies the refinement inclusion property

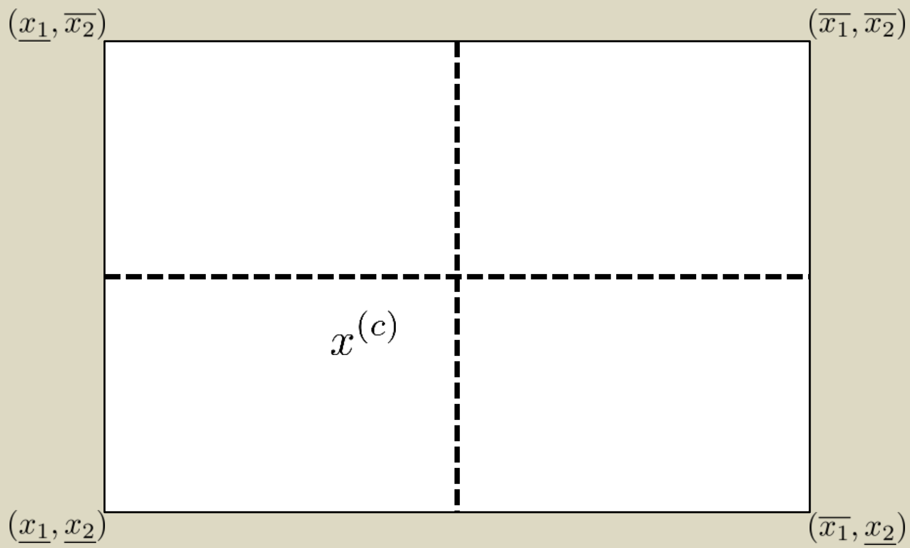
Hybridization: Affine Approximation



$$A_j^{(i)} = \frac{f([\frac{\overline{x_1} + \underline{x_1}}{2} \; ... \; \overline{x_j} \; ... \; \frac{\overline{x_n} + \underline{x_n}}{2}]) - f([\frac{\overline{x_1} + \underline{x_1}}{2} \; ... \; \underline{x_j} \; ... \; \frac{\overline{x_n} + \underline{x_n}}{2}])}{L_j}$$

where L_j is the grid size along x_j

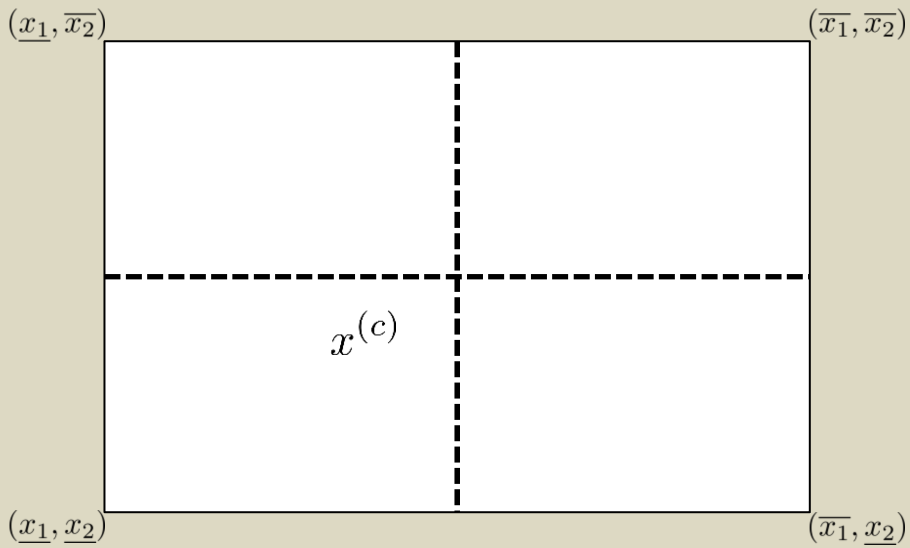
Hybridization: Affine Approximation



$$A_j^{(i)} = \frac{f([\frac{\overline{x_1} + \underline{x_1}}{2} \ldots \overline{x_j} \ldots \frac{\overline{x_n} + \underline{x_n}}{2}]) - f([\frac{\overline{x_1} + \underline{x_1}}{2} \ldots \underline{x_j} \ldots \frac{\overline{x_n} + \underline{x_n}}{2}])}{L_j}$$

$$f^{(i)} = f(\frac{\overline{x} + \underline{x}}{2}) - A^{(i)} \frac{\overline{x} + \underline{x}}{2}$$

Hybridization: Affine Approximation

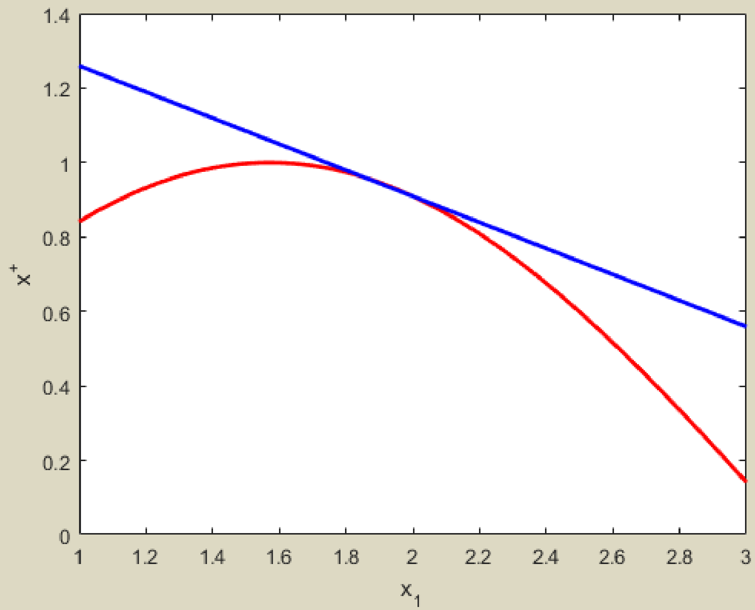


$$A_j^{(i)} = \frac{f([\frac{\overline{x_1}+\underline{x_1}}{2} \ldots \overline{x_j} \ldots \frac{\overline{x_n}+\underline{x_n}}{2}]) - f([\frac{\overline{x_1}+\underline{x_1}}{2} \ldots \underline{x_j} \ldots \frac{\overline{x_n}+\underline{x_n}}{2}])}{L_j}$$

$$f^{(i)} = f(\frac{\overline{x}+\underline{x}}{2}) - A^{(i)}\frac{\overline{x}+\underline{x}}{2}$$

$$\boxed{f_{PWA}(x^{(c)}) = A^{(i)}x^{(c)} + f^{(i)}}$$

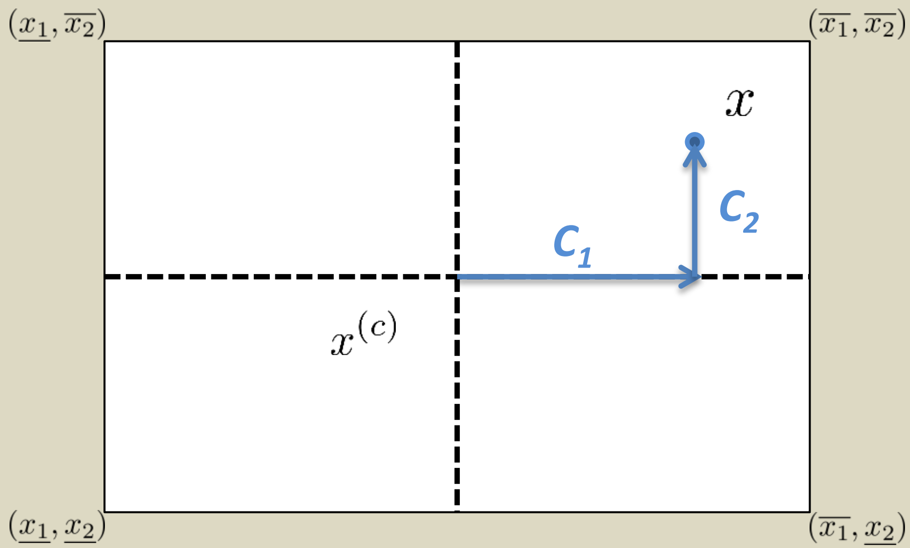
Hybridization: Affine Approximation



$$A_j^{(i)} = \frac{f([\frac{\overline{x_1}+\underline{x_1}}{2} \dots \overline{x_j} \dots \frac{\overline{x_n}+\underline{x_n}}{2}]) - f([\frac{\overline{x_1}+\underline{x_1}}{2} \dots \underline{x_j} \dots \frac{\overline{x_n}+\underline{x_n}}{2}])}{L_j}$$

$$f^{(i)} = f(\frac{\overline{x}+\underline{x}}{2}) - A^{(i)}\frac{\overline{x}+\underline{x}}{2} \qquad \boxed{f_{PWA}(x^{(c)}) = A^{(i)}x^{(c)} + f^{(i)}}$$

Hybridization: Error Bound



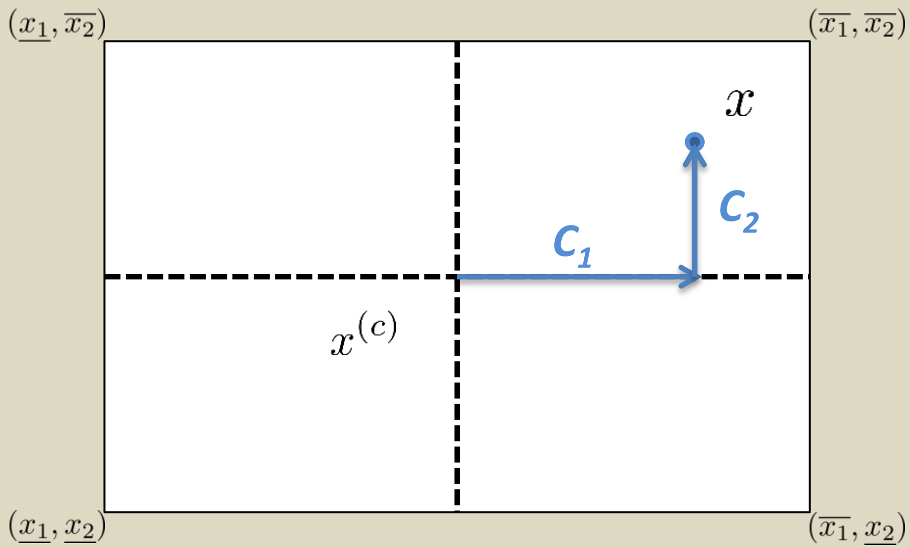
$$\varepsilon(x) = \sum_{j=1}^{n} \left(f^M_{x_j} \Big|_{C_j} - A_j \right)(x_j - x_j^{(c)})$$

Mean-value theorem : average first order partial derivative w.r.t. x_j along path C_j

Hybridization: Error Bound

Assumption:

$$\left| \frac{\partial^2 f}{\partial x_j \partial x_h} \right| \leq M_H$$

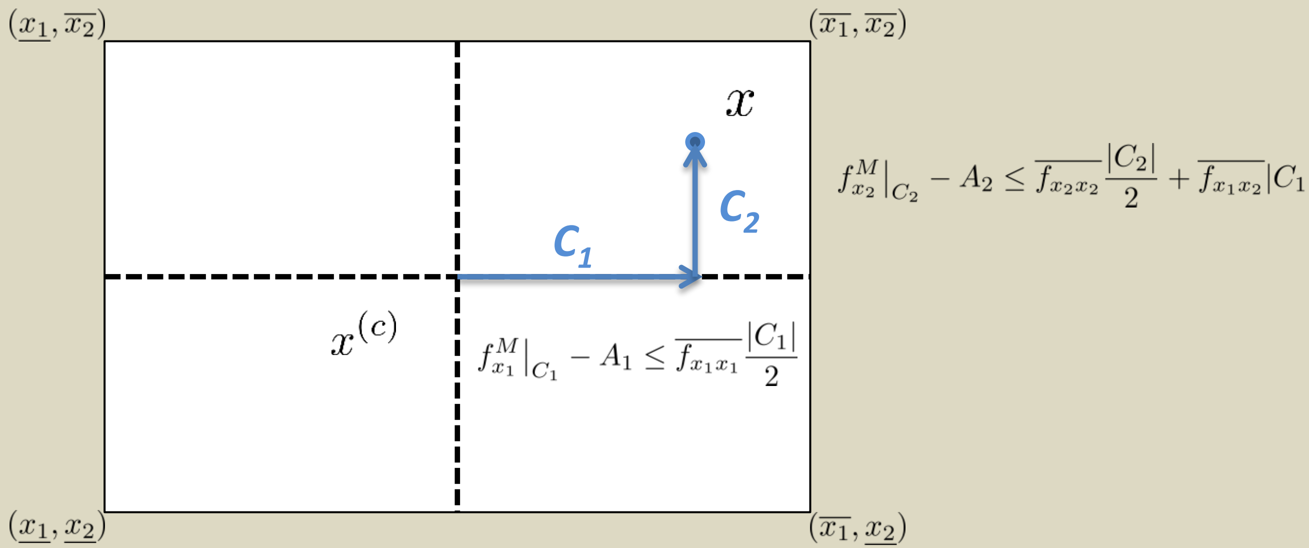


$$\varepsilon(x) = \sum_{j=1}^{n} \left(f_{x_j}^{M} \Big|_{C_j} - A_j \right)(x_j - x_j^{(c)})$$

Hybridization: Error Bound

Assumption:

$$\left| \frac{\partial^2 f}{\partial x_j \partial x_h} \right| \leq M_H$$

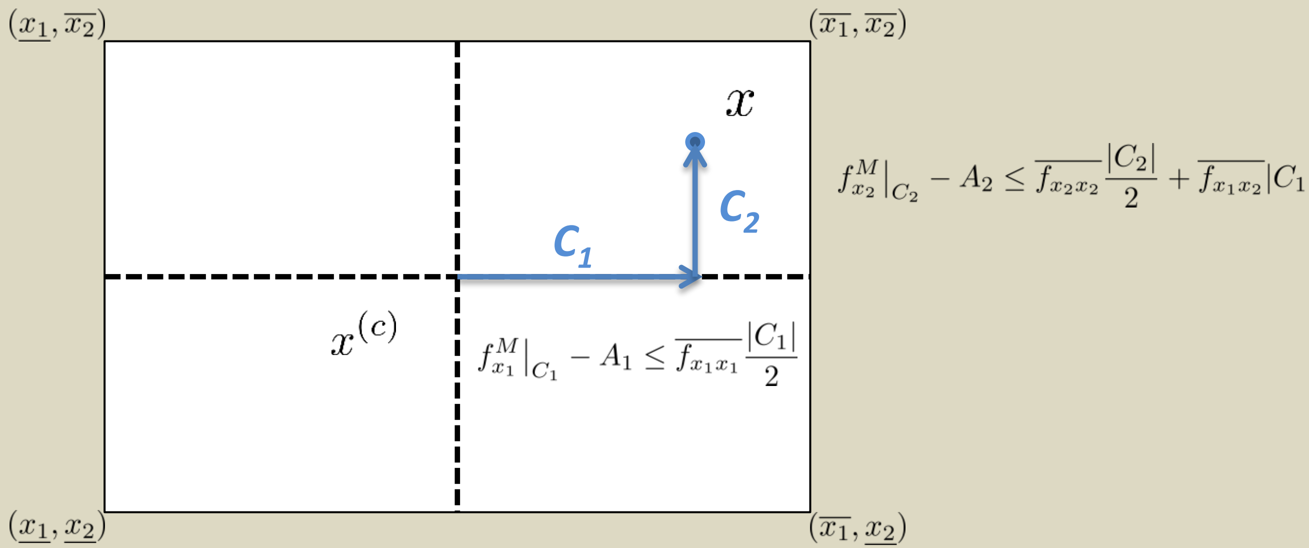


$$\varepsilon(x) = \sum_{j=1}^{n} \boxed{\left(f^M_{x_j} \Big|_{C_j} - A_j \right)} (x_j - x_j^{(c)})$$

Hybridization: Error Bound

Assumption:

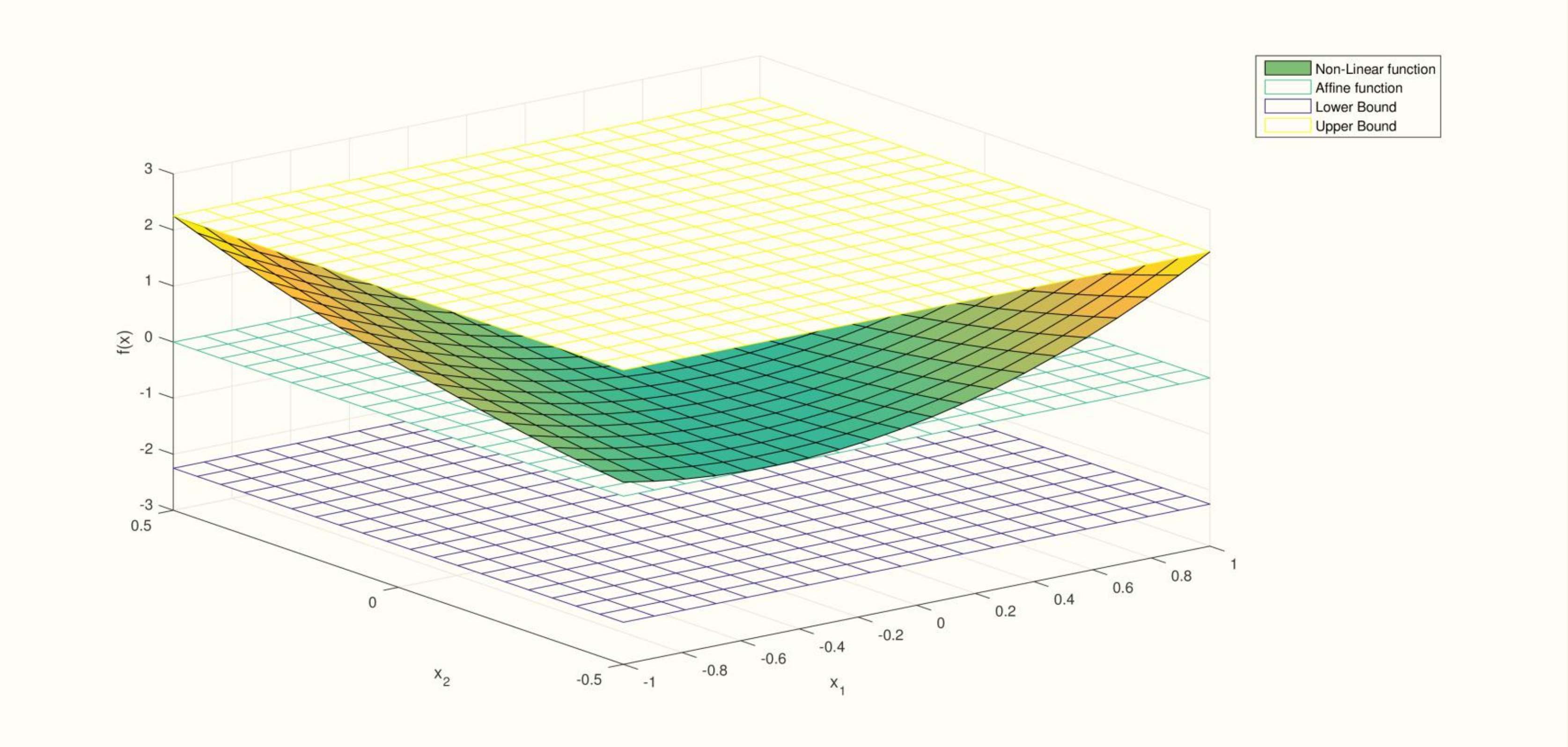
$$\left|\frac{\partial^2 f}{\partial x_j \partial x_h}\right| \le M_H$$



$$\varepsilon(x) = \sum_{j=1}^{n} \left(f^M_{x_j}\Big|_{C_j} - A_j\right)(x_j - x_j^{(c)}) \le \frac{M_H}{8} \sum_{j=1}^{n} \sum_{h=1}^{n} L_j L_h$$

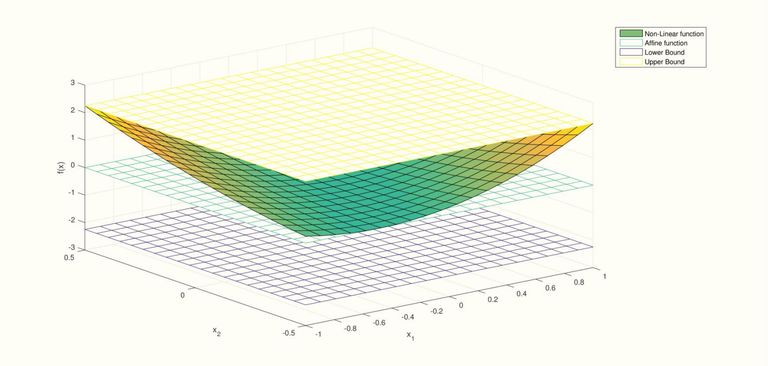
where L_j is the grid size along x_j and hence $|x_j - x_j^{(c)}| = |C_j| \le \dfrac{L_j}{2}$

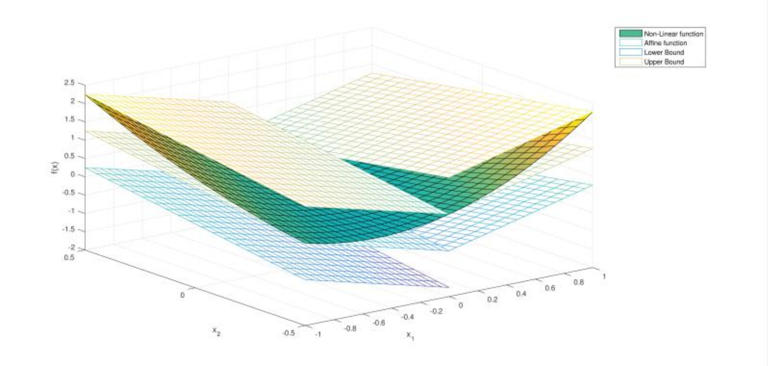
Hybridization: Refinement Inclusion

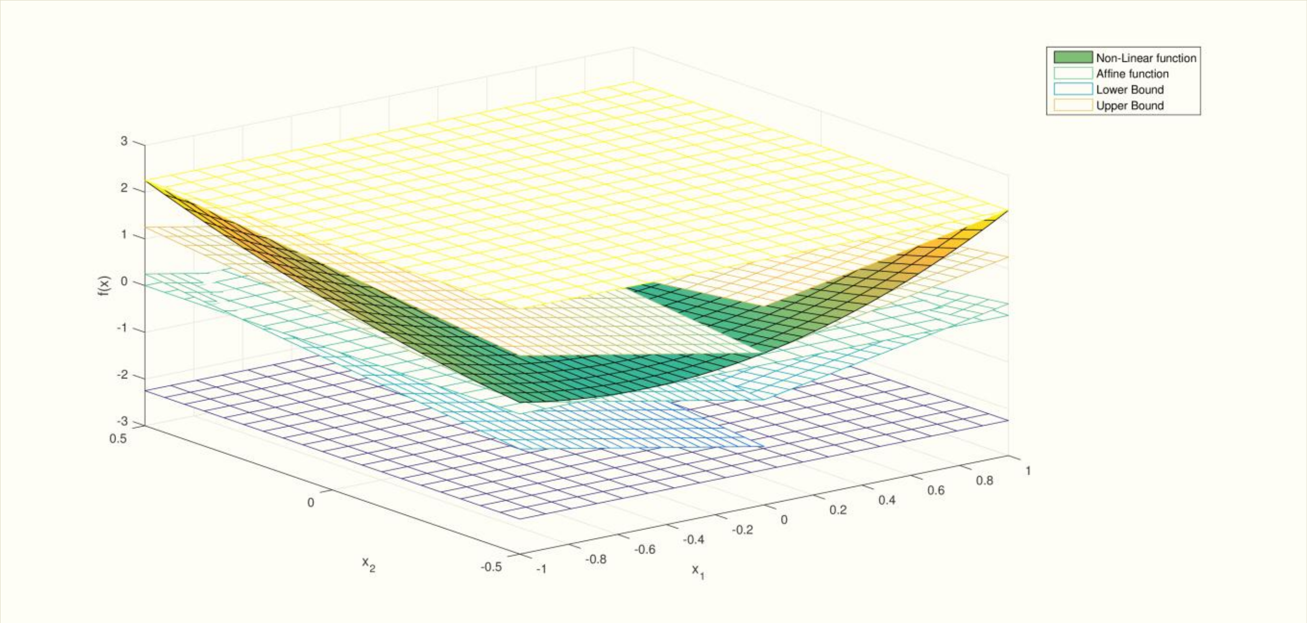


$$f(x) = (x_1 + x_2)^2$$

Hybridization: Refinement Inclusion







How to improve scalability?

- adaptive gridding
- guided refinement
- exploit the interconnection structure and decouple in lower dimensional sub-problems
- PWA model reduction

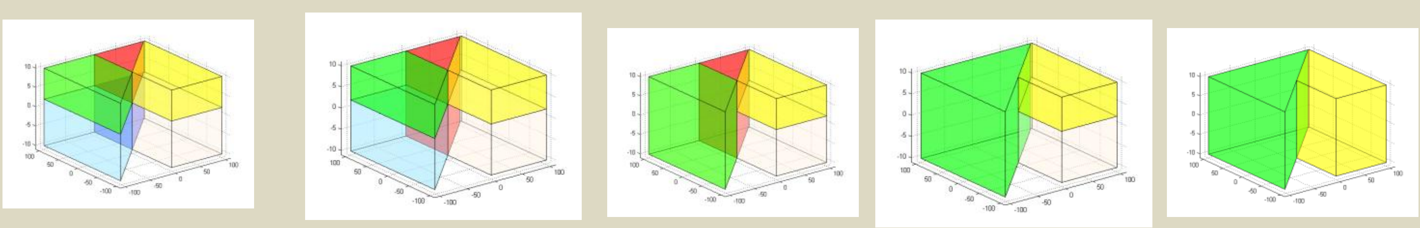
How to improve scalability?

- adaptive gridding
- guided refinement
- exploit the interconnection structure and decouple in lower dimensional sub-problems
- PWA model reduction

Model reduction for PWA systems

- if the specs are on some output variable, one can eliminate the input and state variables that do not affect the output (model reduction)

- Contributions:
  - introduction of a structural approach to model reduction based on observability properties of PWA systems
  - modes merging in the resulting PWA models



Summary

- reachability can be used for safety verification
- reachability is in general hard for hybrid systems due to the continuous component
- we described a set-based approach to reachability computations for nonlinear continuous dynamics, which is based on
  - conformant model approximation with refinement inclusion
  - reachability analysis via zonotopic set propagation

Summary

Extend the approach to

- enforcement of safety

when the system evolution can be affected by some control input

[from analysis to design]

An example

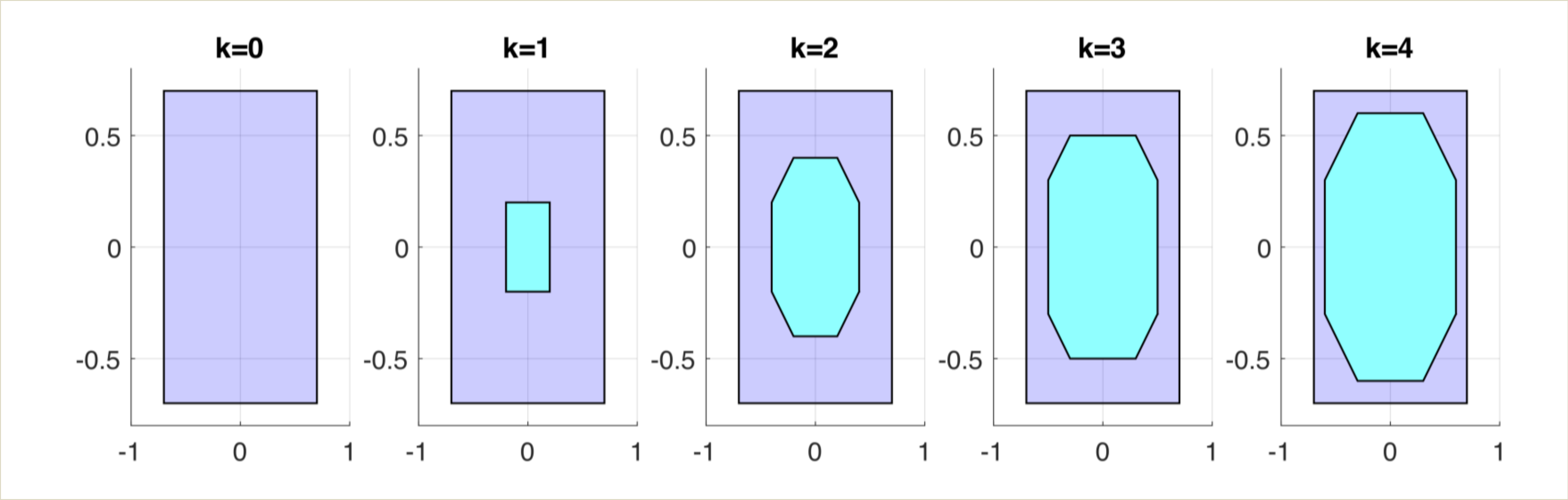
$$x^+ = \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} x + w,\; w \in [-0.2, 0.2]^2 \qquad \mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

$$Sp_k = \{\|x\|_\infty \leq 0.7\},\; k = 1, \ldots, 9$$

An example

$$x^+ = \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} x + w,\; w \in [-0.2, 0.2]^2 \qquad \mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

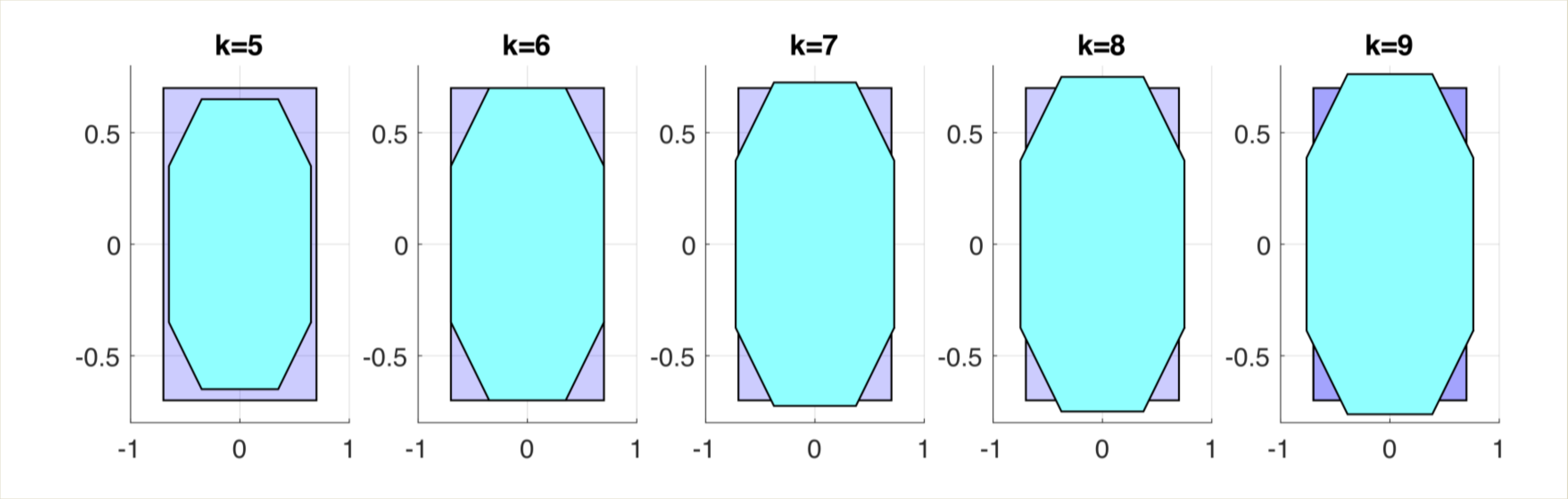
$$Sp_k = \{\|x\|_\infty \leq 0.7\},\; k = 1, \dots, 9$$



An example

$$x^+ = \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} x + w,\ w \in [-0.2, 0.2]^2 \qquad \mathcal{X}_i = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}$$

$$Sp_k = \{\|x\|_\infty \leq 0.7\},\ k = 1, \ldots, 9$$



An example

$$x^+ = \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} x + \boxed{\begin{bmatrix} 1 \\ 0 \end{bmatrix} u} + w,$$

control input
that can be set to enforce the specs

An example

$$x^+ = \begin{bmatrix} 0.5 & -0.5 \\ 0.5 & 0.5 \end{bmatrix} x + \boxed{\begin{bmatrix} 1 \\ 0 \end{bmatrix} u} + w,$$

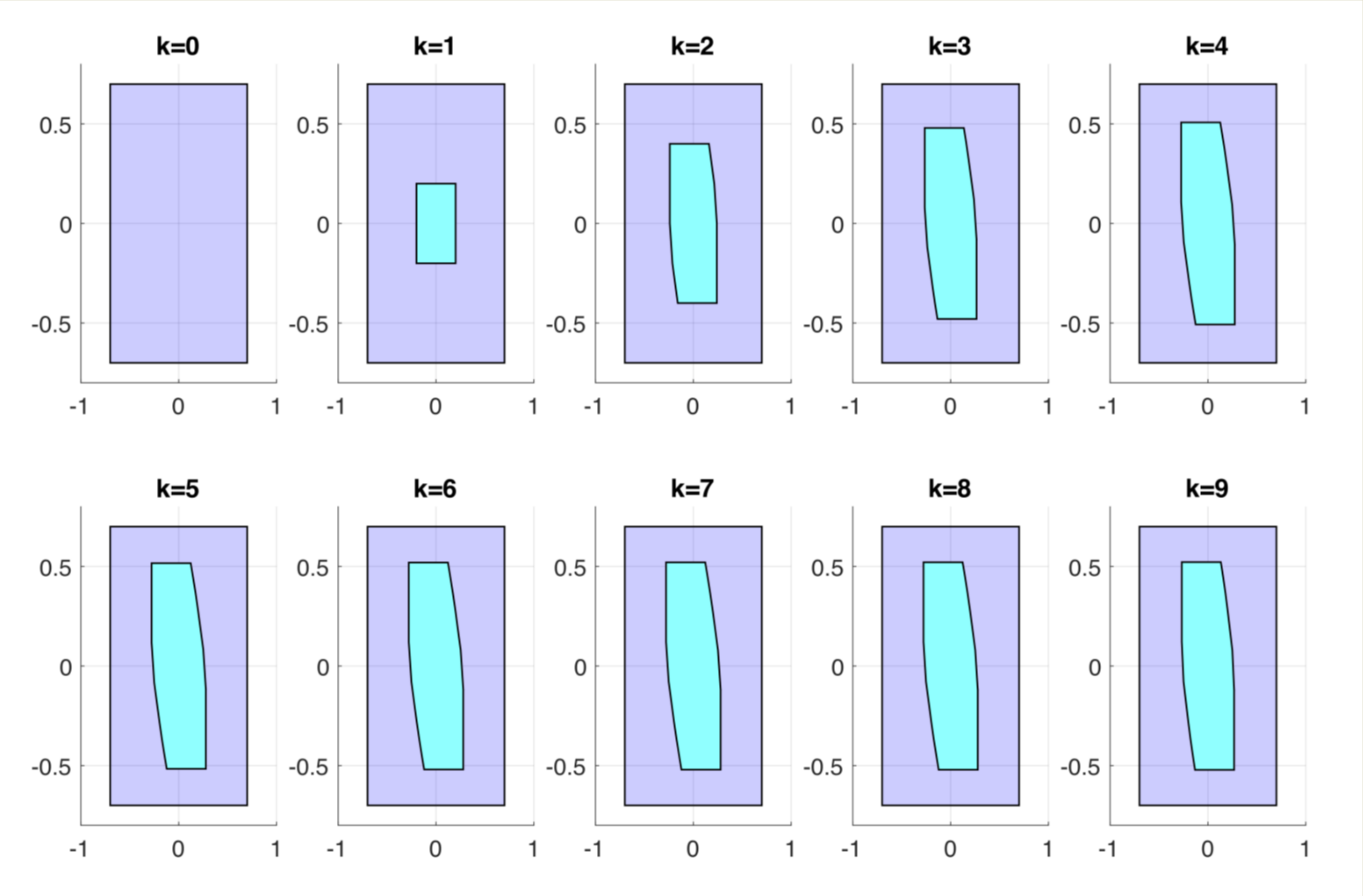
control input
that can be set to enforce the specs

$$\text{minimize } \mathbb{E}_w \left[\sum_{k=0}^{8} \Big(x(k+1)^T Q x(k+1) + u(k)^T R u(k) \Big) \right]$$

subject to:

$$\begin{cases} \|u_i\|_\infty \leq \bar{u} & i = 0, \ldots, 8 \\ \|x(k)\|_\infty \leq \bar{y} & k = 1, \ldots, 9 \end{cases},$$

Results with control

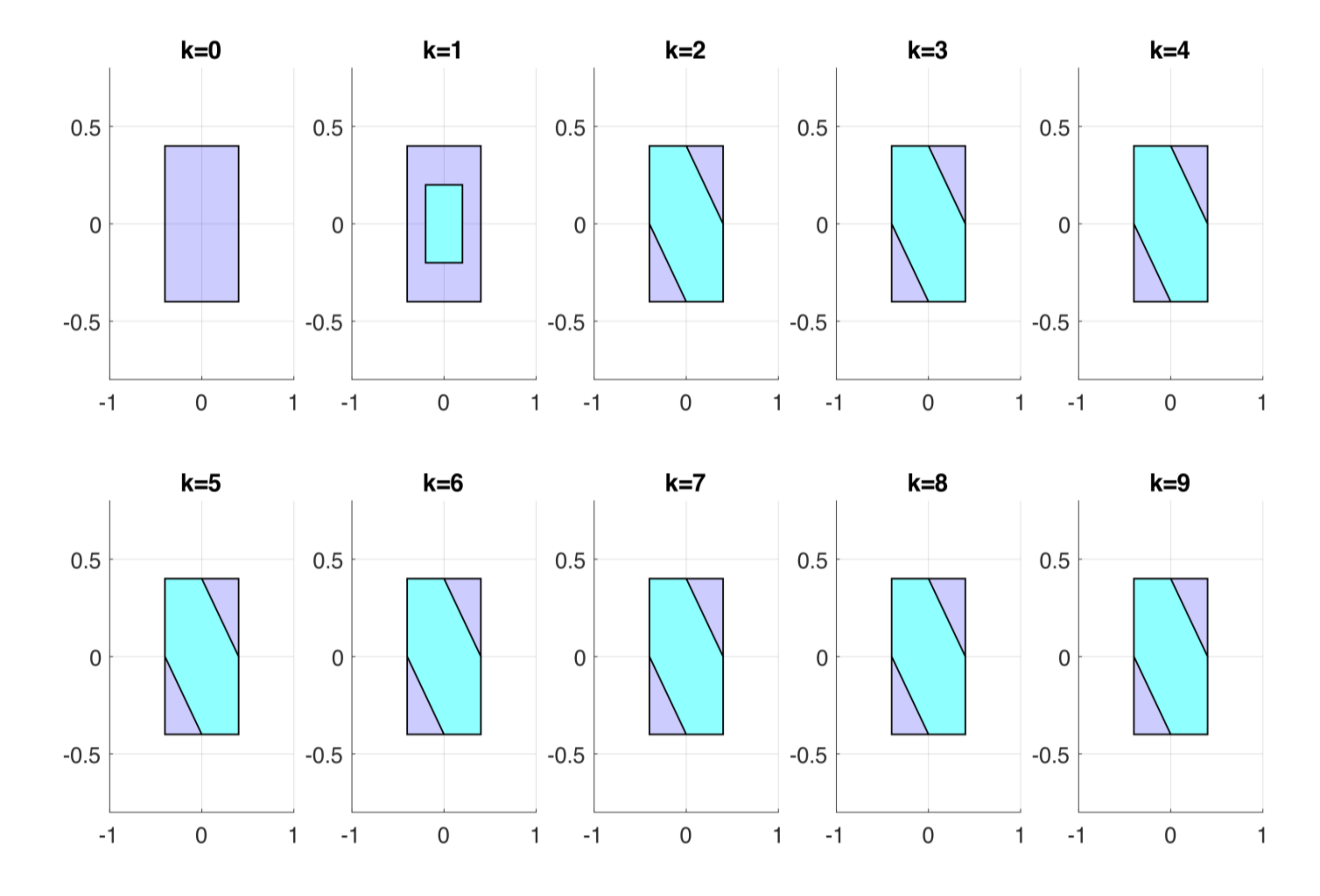


$$Q = I_2$$

$$R = 0.1$$

$$\bar{y} = \bar{u} = 0.7$$

Results with control

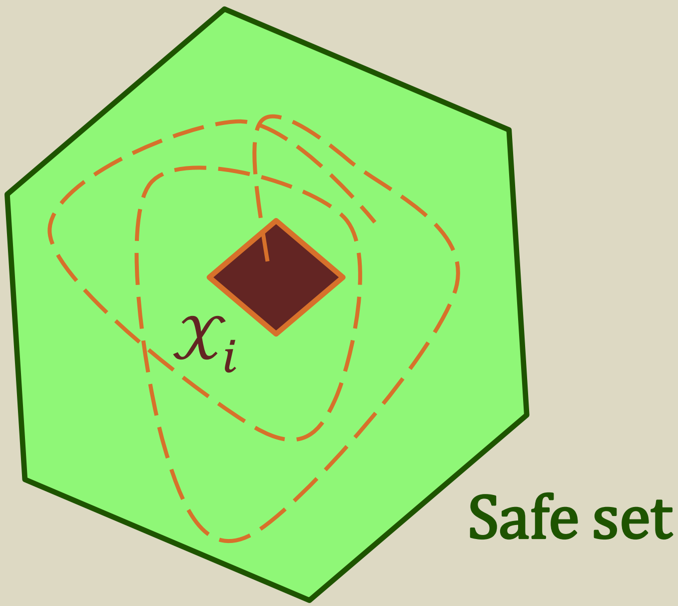


$$Q = I_2$$
$$R = 0.1$$

$$\bar{y} = \bar{u} = 0.4$$

Safety control problem

design a controller that keeps the state of the system within some given safe set indefinitely



Safety control problem

Set-up

- discrete time system with affine in the input nonlinear dynamics

$$x^+ = f(x,u) = f_1(x) + f_2(x)u$$

- a safe set $\mathcal{H}$ described via a set of linear constraints that should always remain true

$$\mathcal{H} = \{x : H_A x \leq H_B\}$$

- an initial set $\mathcal{X}_i$

Safety control problem

Find a state feedback control law $u = \sigma(x)$ and a set $\mathcal{I}$ such that:

$$f(x, \sigma(x)) \in \mathcal{I}, \quad \forall x \in \mathcal{I}$$

- the control law makes $\mathcal{I}$ controlled invariant

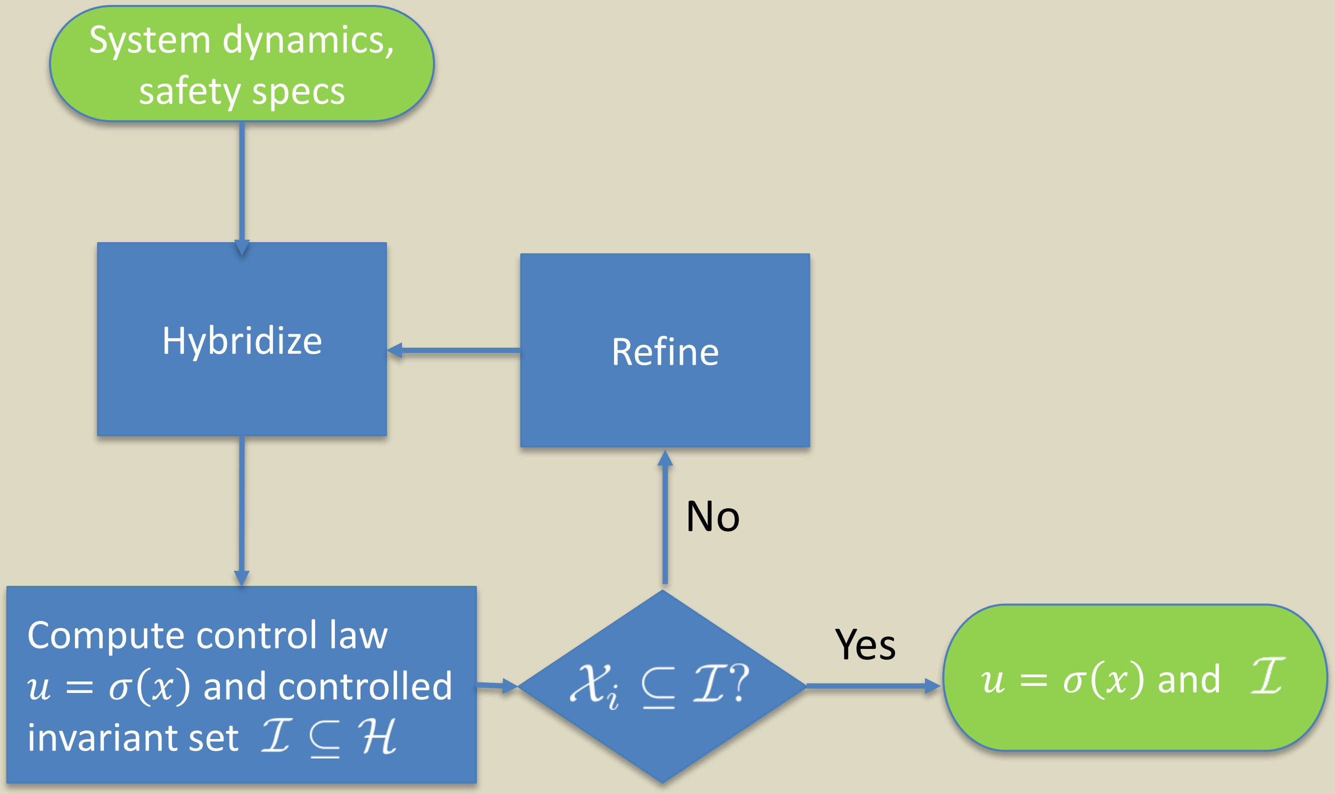
$$\mathcal{I} \subseteq \mathcal{H}$$

- $\mathcal{I}$ satisfies the safety specification

$$\mathcal{X}_i \subseteq \mathcal{I}$$

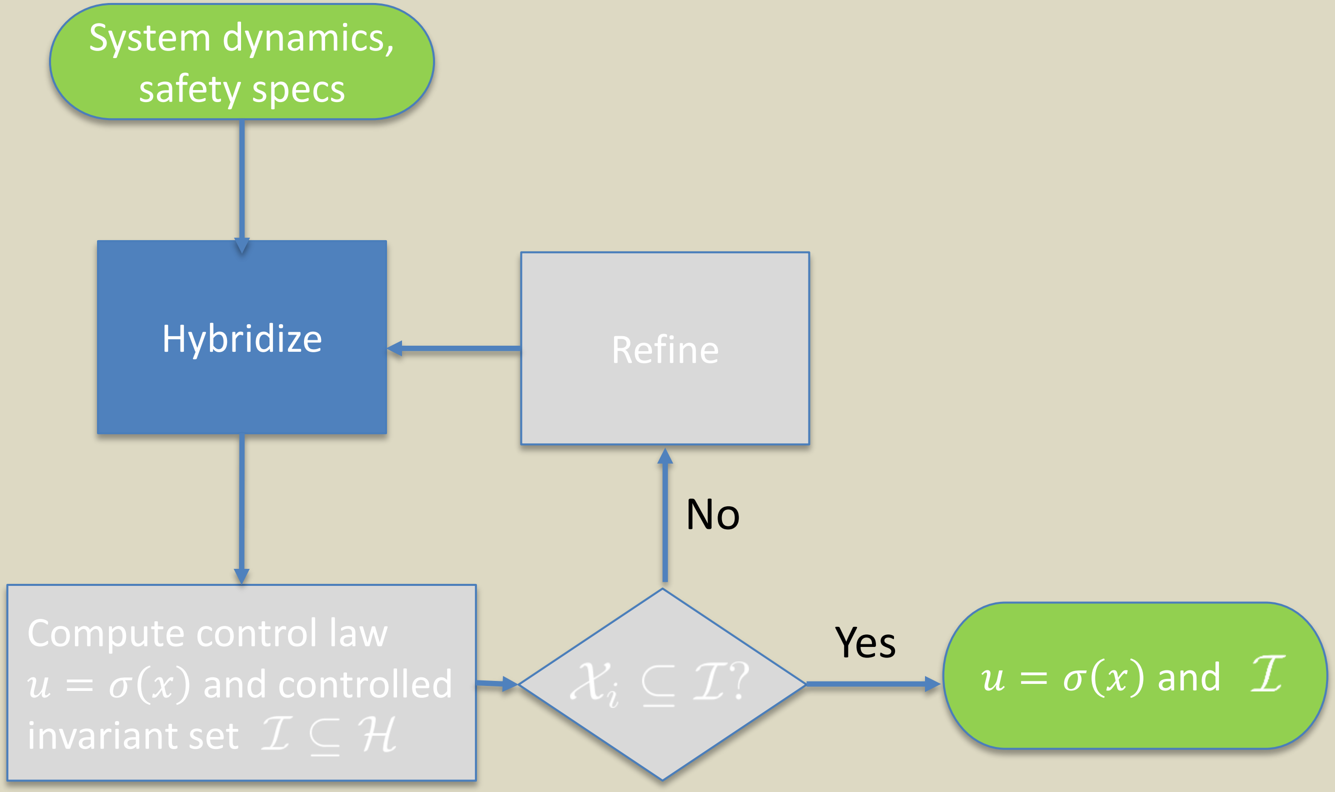
- $\mathcal{I}$ covers the set of initial states

Approach



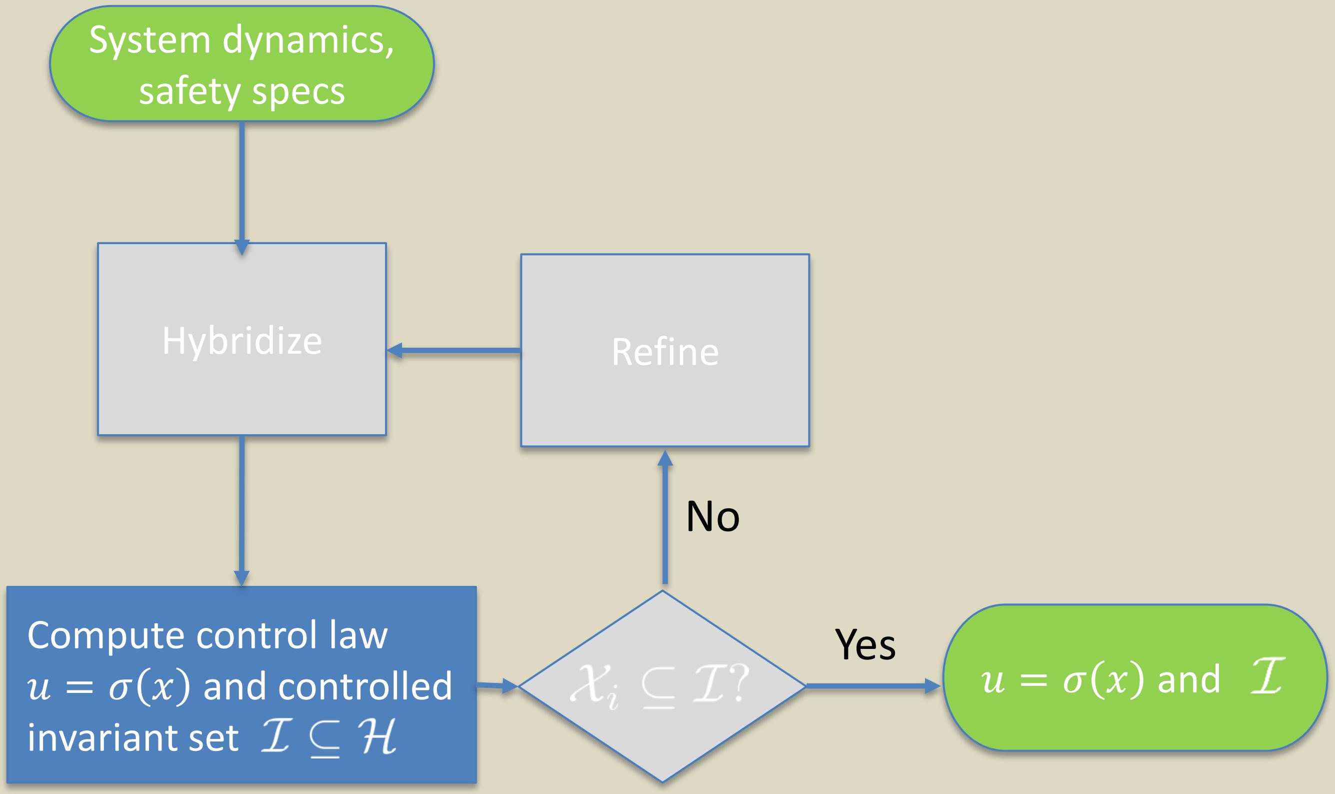
System dynamics, safety specs
Hybridize
Refine
No
Compute control law $u = \sigma(x)$ and controlled invariant set $\mathcal{I} \subseteq \mathcal{H}$
$\mathcal{X}_i \subseteq \mathcal{I}$?
Yes
$u = \sigma(x)$ and $\mathcal{I}$

Approach



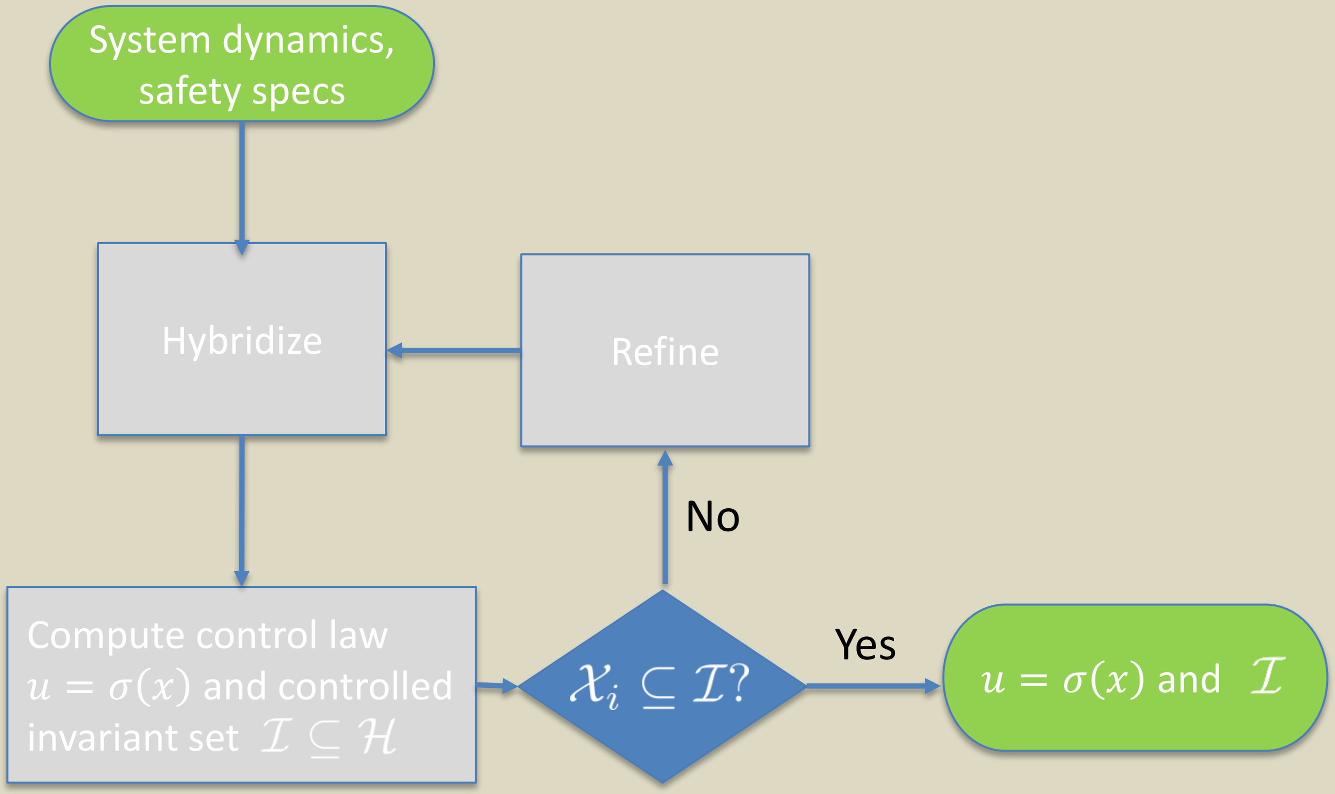
System dynamics, safety specs
Hybridize
Refine
No
Compute control law $u = \sigma(x)$ and controlled invariant set $\mathcal{I} \subseteq \mathcal{H}$
$\mathcal{X}_i \subseteq \mathcal{I}$?
Yes
$u = \sigma(x)$ and $\mathcal{I}$

Approach



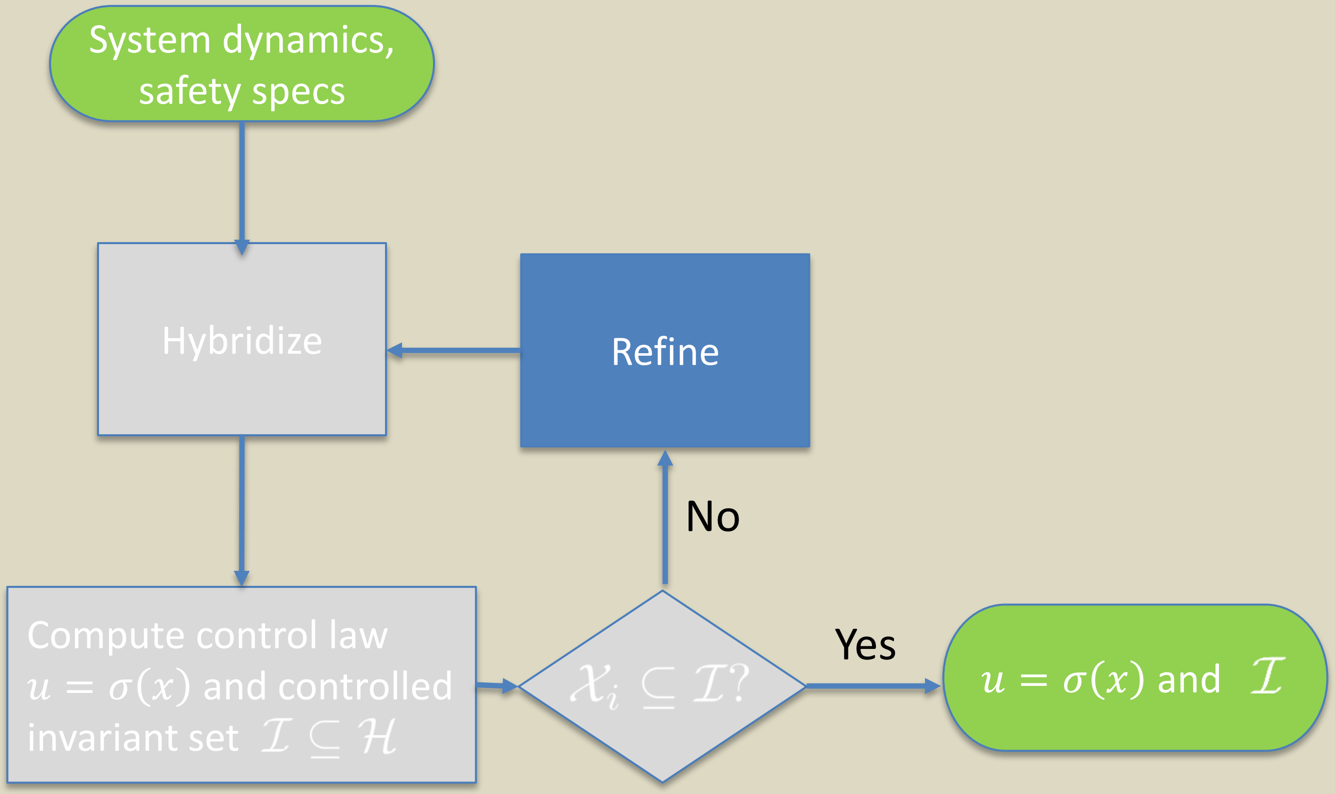
System dynamics, safety specs
Hybridize
Refine
No
Compute control law $u = \sigma(x)$ and controlled invariant set $\mathcal{I} \subseteq \mathcal{H}$
$\mathcal{X}_i \subseteq \mathcal{I}?$
Yes
$u = \sigma(x)$ and $\mathcal{I}$

Approach



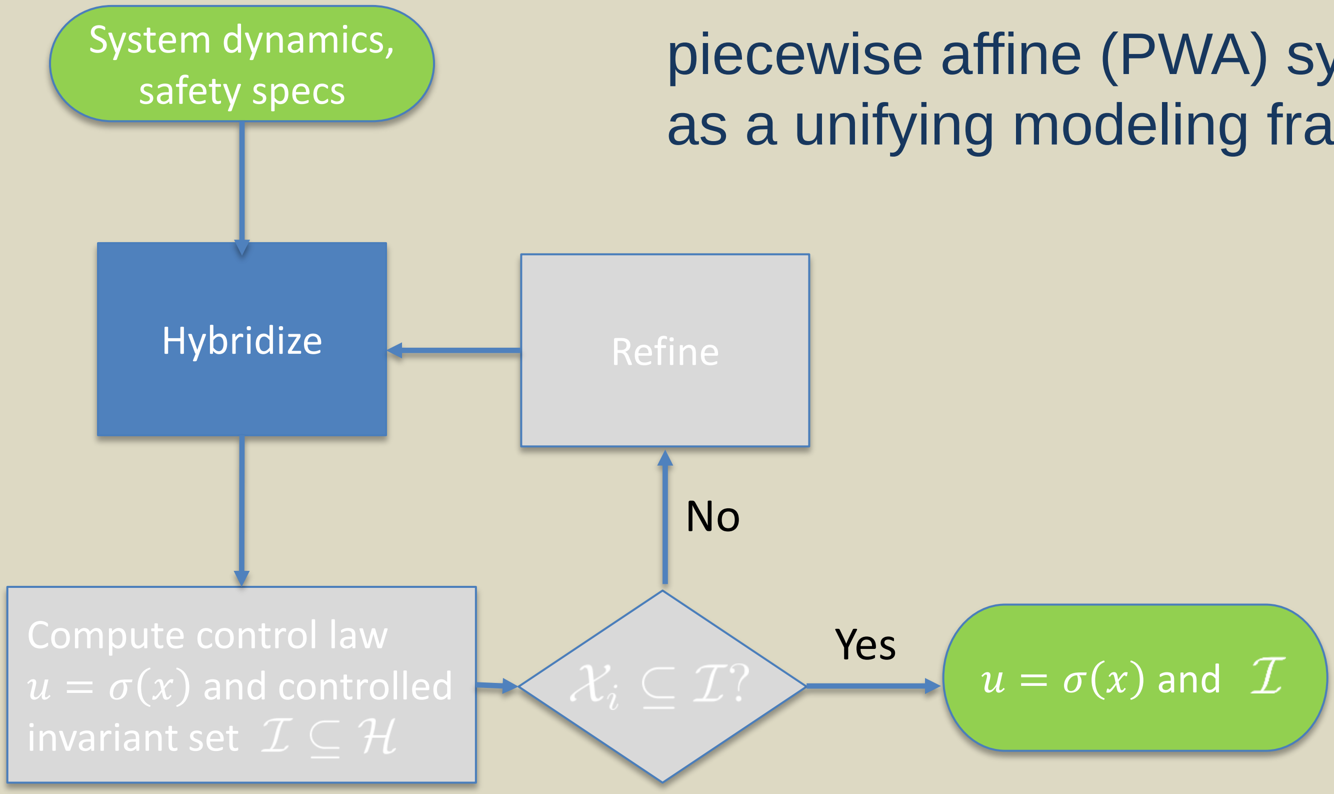
System dynamics, safety specs
Hybridize
Refine
No
Compute control law $u = \sigma(x)$ and controlled invariant set $\mathcal{I} \subseteq \mathcal{H}$
$\mathcal{X}_i \subseteq \mathcal{I}$?
Yes
$u = \sigma(x)$ and $\mathcal{I}$

Approach

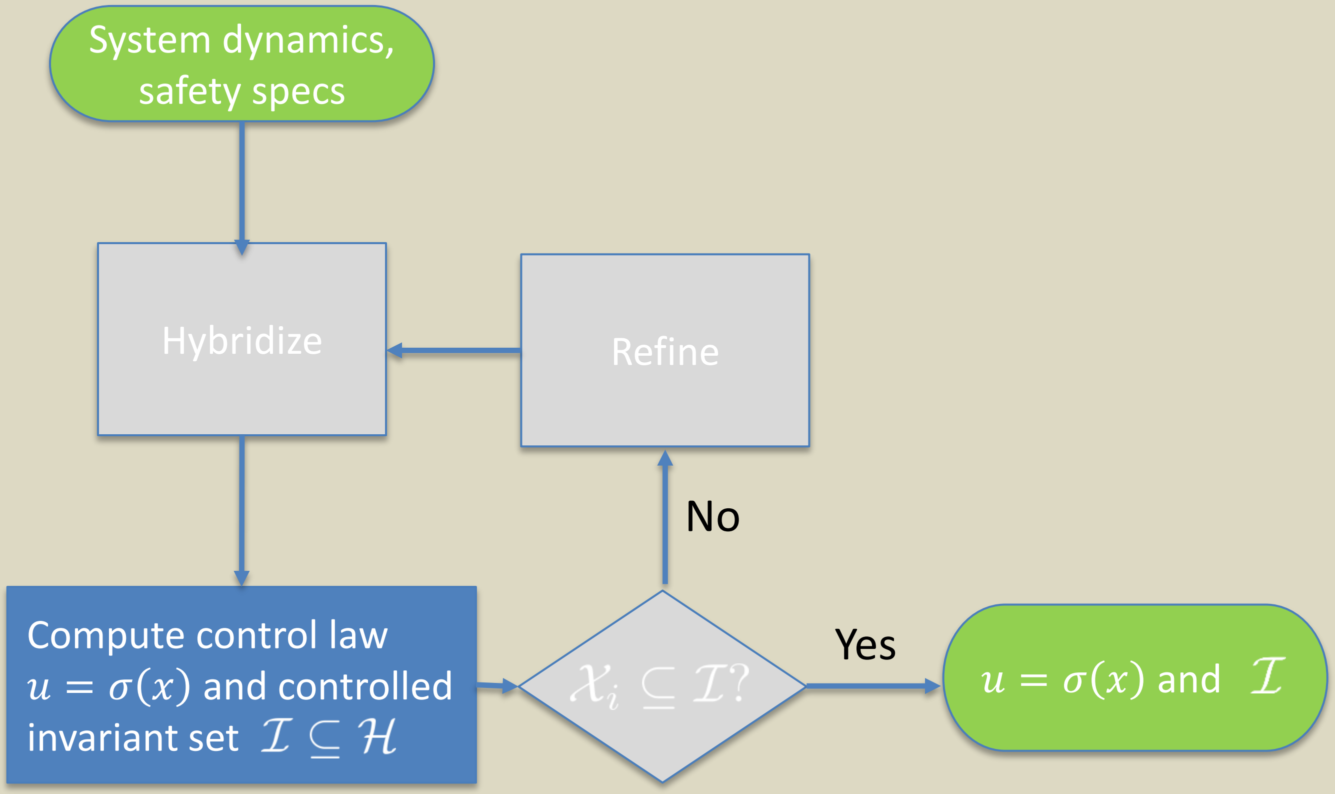

System dynamics, safety specs
Hybridize
Refine
No
Compute control law $u = \sigma(x)$ and controlled invariant set $\mathcal{I} \subseteq \mathcal{H}$
$\mathcal{X}_i \subseteq \mathcal{I}$?
Yes
$u = \sigma(x)$ and $\mathcal{I}$

Hybridization

piecewise affine (PWA) systems
as a unifying modeling framework



Invariant set and control law



Invariant set and control law

Given

- hybridization

$$x^+ = A^{(i)}x + B^{(i)}u + f^{(i)} + w,\ w \in [\underline{W}^{(i)}, \overline{W}^{(i)}],\ \ x \in [\underline{X}^{(i)}, \overline{X}^{(i)}]$$

- safe set $\mathcal{H} = \{x : H_A x \leq H_B\}$

Invariant set and control law

Given

- hybridization

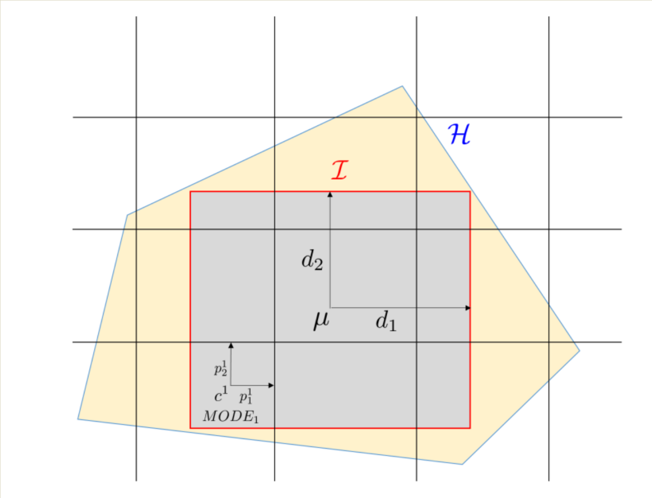
$$x^+ = A^{(i)}x + B^{(i)}u + f^{(i)} + w,\ w \in [\underline{W}^{(i)}, \overline{W}^{(i)}],\ \ x \in [\underline{X}^{(i)}, \overline{X}^{(i)}]$$

- safe set $\mathcal{H} = \{x : H_A x \leq H_B\}$

Design

- largest controlled invariant set in the form of a box
  $\mathcal{I} = [\mu - d, \mu + d]$
  that is contained within the safe set
- associated state feedback control law

$$u = \sigma(x)$$



Invariant set and control law

Linear system:

$$x^+ = Ax + Bu$$

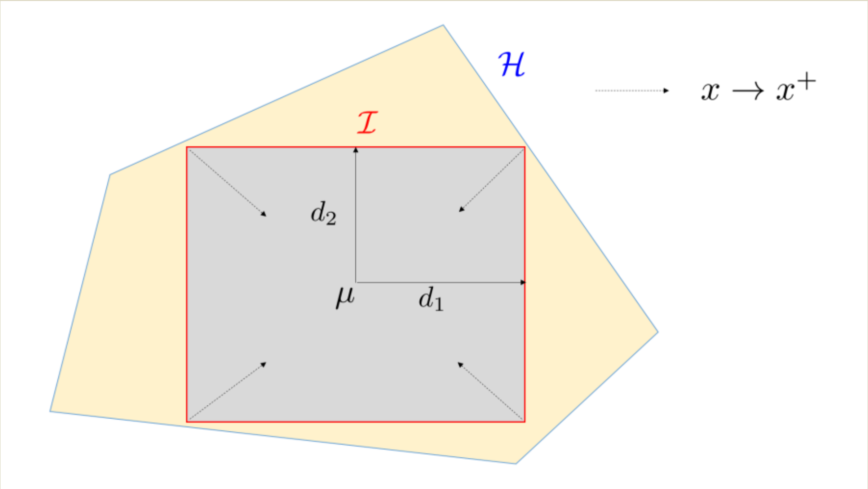
Invariant set and control law

Linear system:

$$x^+ = Ax + Bu$$

Largest invariant set in form of a box

$$x \in \mathcal{I} = [\mu - d, \mu + d] \Leftrightarrow x = \mu + diag(d)\alpha(x), \alpha(x) \in [-1, 1]^n$$



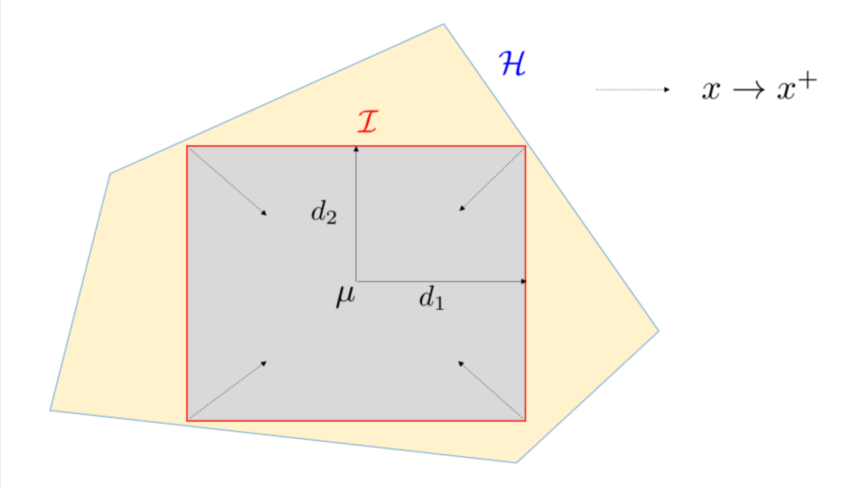
Invariant set and control law

Linear system:

$$x^+ = Ax + Bu$$

Largest invariant set in form of a box

$$x = \mu + diag(d)\alpha(x), \alpha(x) \in [-1, 1]^n$$



Invariant set and control law

Linear system:

$$x^+ = Ax + Bu$$

Largest invariant set in form of a box

$$x = \mu + diag(d)\alpha(x), \alpha(x) \in [-1, 1]^n$$

Control law:

$$u = \sigma(x) = u_\mu + U_G\alpha(x)$$

Invariant set and control law

Linear system:

$$x^+ = Ax + Bu$$

Largest invariant set in form of a box

$$x = \mu + diag(d)\alpha(x), \alpha(x) \in [-1, 1]^n$$

Control law:

$$u = \sigma(x) = u_\mu + U_G\alpha(x)$$

Closed-loop system:

$$x^+ = f(x, \sigma(x)) = A\mu + Bu_\mu + [Adiag(d) + BU_G]\alpha(x)$$

Invariant set and control law

Linear system:

$$x^+ = Ax + Bu$$

Largest invariant set in form of a box

$$x = \boxed{\mu} + diag\boxed{(d)}\alpha(x), \alpha(x) \in [-1, 1]^n$$

Control law:

$$u = \sigma(x) = \boxed{u_\mu} + \boxed{U_G}\alpha(x)$$

Closed-loop system:

$$x^+ = f(x, \sigma(x)) = A\mu + Bu_\mu + [Adiag(d) + BU_G]\alpha(x)$$

Invariant set and control law

Linear system:

$$x^+ = Ax + Bu$$

Largest invariant set in form of a box

$$x = \boxed{\mu} + diag\boxed{(d)}\alpha(x), \alpha(x) \in [-1, 1]^n$$

Control law:

$$u = \sigma(x) = \boxed{u_\mu} + \boxed{U_G}\alpha(x)$$

Performance index $$\max_{\mu, d, u_\mu, U_G} \sum_{i=1}^{n} d_i$$

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I}, \, \forall x \in \mathcal{I}$$

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I}, \ \forall x \in \mathcal{I}$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ \leq \mu + d$$

$$\min_{x \in [\mu - d, \mu + d]} x^+ \geq \mu - d$$

to be interpreted componentwise

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I}, \forall x \in \mathcal{I}$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ \leq \mu + d$$

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I},\ \forall x \in \mathcal{I}$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ \leq \mu + d$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ = \max_{\alpha(x) \in [-1, 1]^n} [A\mu + Bu_\mu + [Adiag(d) + BU_G]\alpha(x)]$$

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I}, \forall x \in \mathcal{I}$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ \leq \mu + d$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ = \max_{\alpha(x) \in [-1,1]^n} [A\mu + Bu_\mu + [Adiag(d) + BU_G]\alpha(x)]$$

$$A\mu + Bu_\mu + \|Adiag(d) + BU_G\|_1 \leq \mu + d$$

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I},\ \forall x \in \mathcal{I}$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ \leq \mu + d$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ = \max_{\alpha(x) \in [-1,1]^n} [A\mu + Bu_\mu + [Adiag(d) + BU_G]\alpha(x)]$$

$$A\mu + Bu_\mu + \|Adiag\boxed{(d)} + B\boxed{U_G}\|_1 \leq \mu + d$$

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I}, \, \forall x \in \mathcal{I}$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ \leq \mu + d$$

$$\max_{x \in [\mu - d, \mu + d]} x^+ = \max_{\alpha(x) \in [-1,1]^n} [A\mu + Bu_\mu + [Adiag(d) + BU_G]\alpha(x)]$$

$$A\mu + Bu_\mu + \|Adiag\boxed{(d)} + B\boxed{U_G}\|_1 \leq \mu + d$$

$$\|V\|_1 = \sum_{i=1}^{k} |v_i| \leq c \Leftrightarrow \sum_{i=1}^{k} h_i \leq c, \text{ with } |v_i| \leq h_i, i = 1, \ldots, k$$

Invariant set and control law

Invariance constraint:

$$x^+ = f(x, \sigma(x)) \in \mathcal{I}, \forall x \in \mathcal{I}$$

$$(A - I)\mu + Bu_\mu + \|Adiag(d) + BU_G\|_1 - d \leq 0$$
$$-(A - I)\mu - Bu_\mu + \|Adiag(d) + BU_G\|_1 - d \leq 0$$

Linear in the optimization variables

Invariant set and control law

Safety constraint:

$$H_A x \leq H_B, \; x \in \mathcal{I} = [\mu - d, \mu + d]$$

Invariant set and control law

Safety constraint:

$$H_A x \leq H_B, \; x \in \mathcal{I} = [\mu - d, \mu + d]$$

$$\max_{\alpha(x) \in [-1,1]^n} [H_A \mu + H_A diag(d) \alpha(x)] \leq H_B$$

Invariant set and control law

Safety constraint:

$$H_A x \leq H_B, \; x \in \mathcal{I} = [\mu - d, \mu + d]$$

$$\max_{\alpha(x) \in [-1,1]^n} [H_A \mu + H_A diag(d) \alpha(x)] \leq H_B$$

$$H_A \mu + \|H_A diag(d)\|_1 \leq H_B$$

Invariant set and control law

Safety constraint:

$$H_A x \leq H_B, \; x \in \mathcal{I} = [\mu - d, \mu + d]$$

$$\max_{\alpha(x) \in [-1,1]^n} [H_A \mu + H_A diag(d) \alpha(x)] \leq H_B$$

$$H_A \mu + \| H_A diag(d) \|_1 \leq H_B$$

Linear in the optimization variables

Invariant set and control law

For a linear system computing the largest invariant set in form of a box and the control law reduces to solving a Linear Programming (LP) problem

$$\max_{\mu, d, u_\mu, U_G} \sum_{i=1}^{n} d_i$$

subject to:

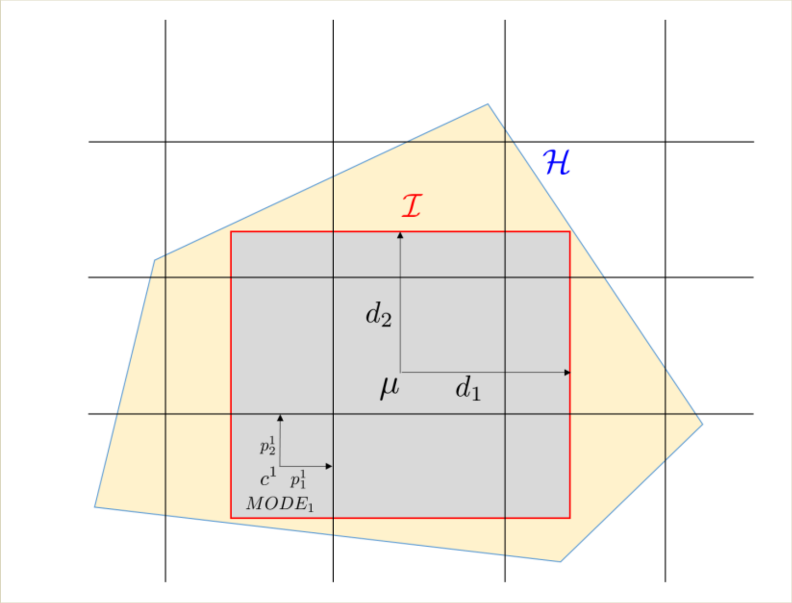
$$(A - I)\mu + Bu_\mu + \|Adiag(d) + BU_G\|_1 - d \leq 0$$
$$-(A - I)\mu - Bu_\mu + \|Adiag(d) + BU_G\|_1 - d \leq 0$$

$$H_A\mu + \|H_A diag(d)\|_1 \leq H_B$$

Invariant Set Computation

PWA system

- Largest invariant set in form of a box possibly covers various modes and splits (in boxes)
- Binary optimization variables identify modes that are covered
- Control law different per mode



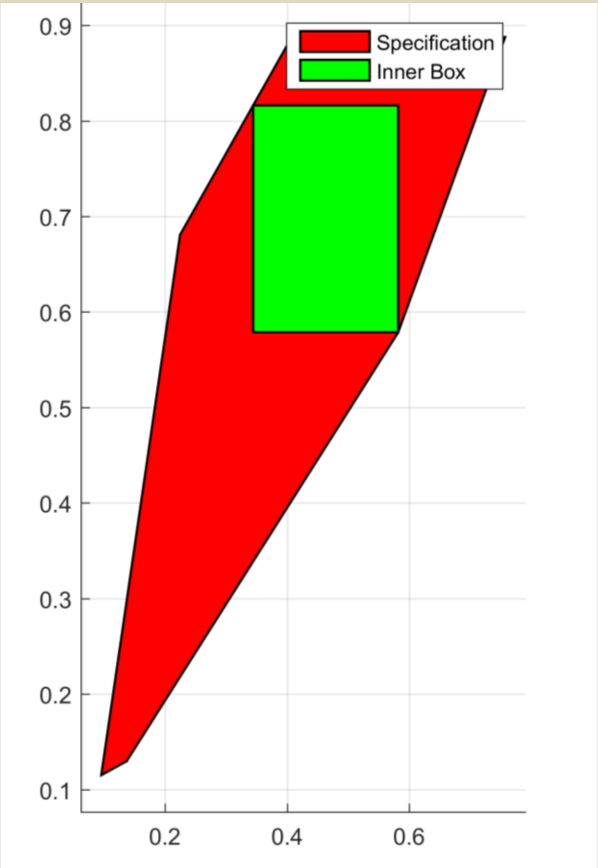
Invariant Set Computation

PWA system

- Largest invariant set in form of a box possibly covers various modes and splits (in boxes)
- Binary optimization variables identify modes that are covered
- Control law different per mode

For a PWA system computing the largest invariant set in form of a box and the control law reduces to solving a Mixed Integer Linear Programming (MILP) problem

Reclined Specification

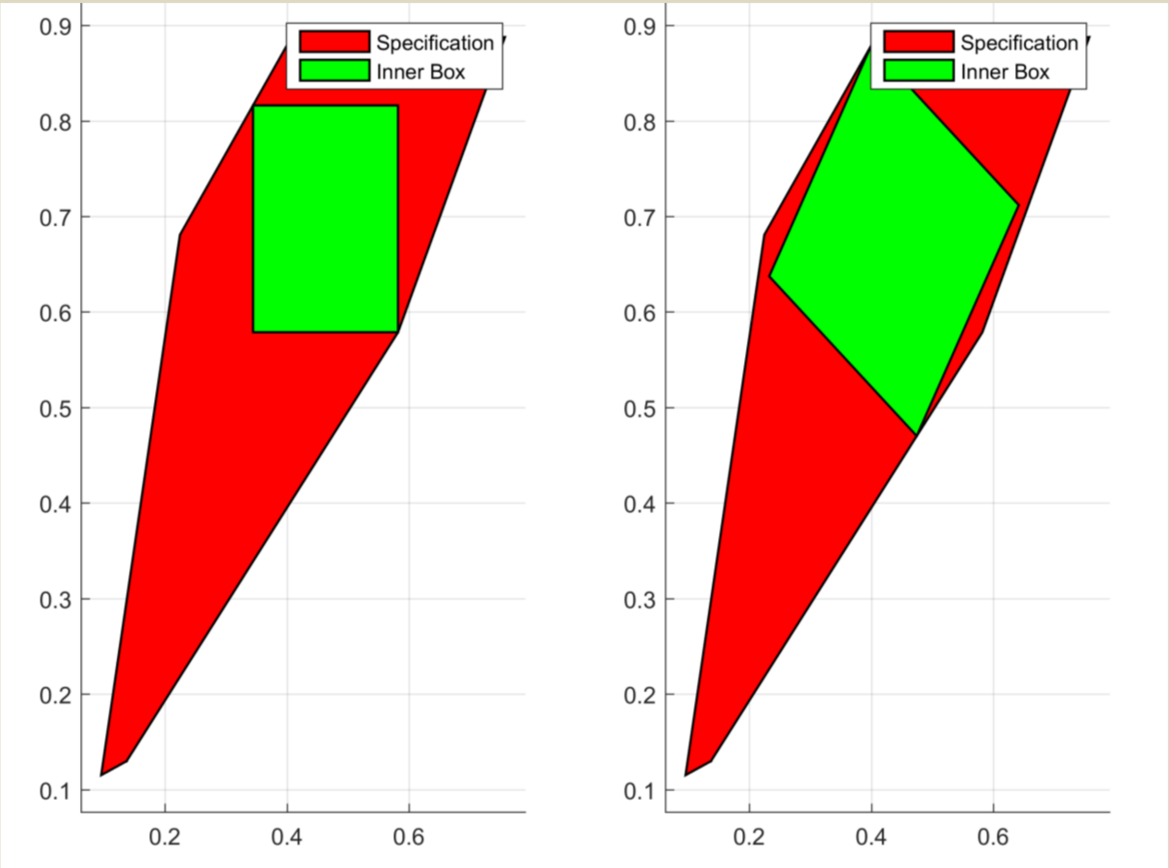


Reclined Specification

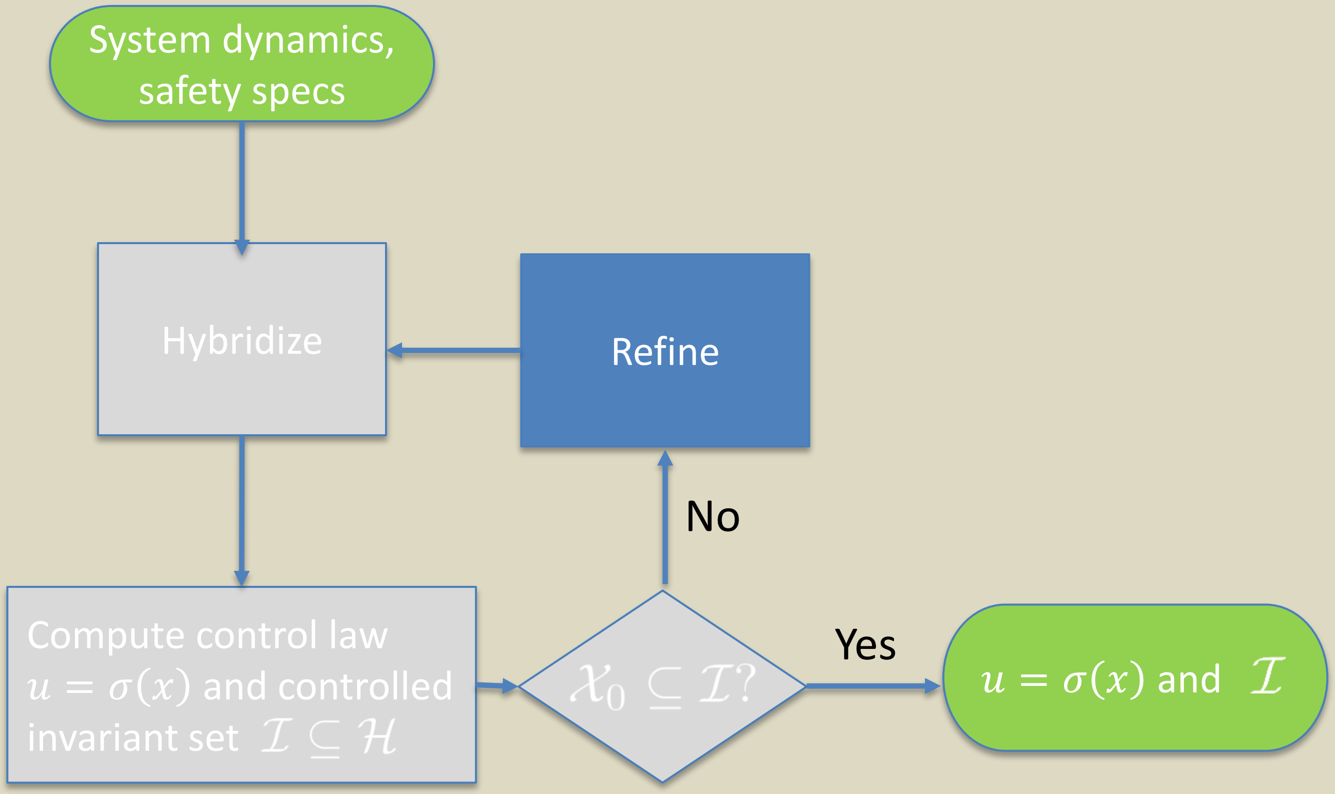
- Transform state space
- Principal component Analysis (PCA) on covariance matrix of vertices interpreted as data points

$$cov(\overline{V}) = U\Sigma P^T$$
$$T = U = P$$

- Hybridization after transformation

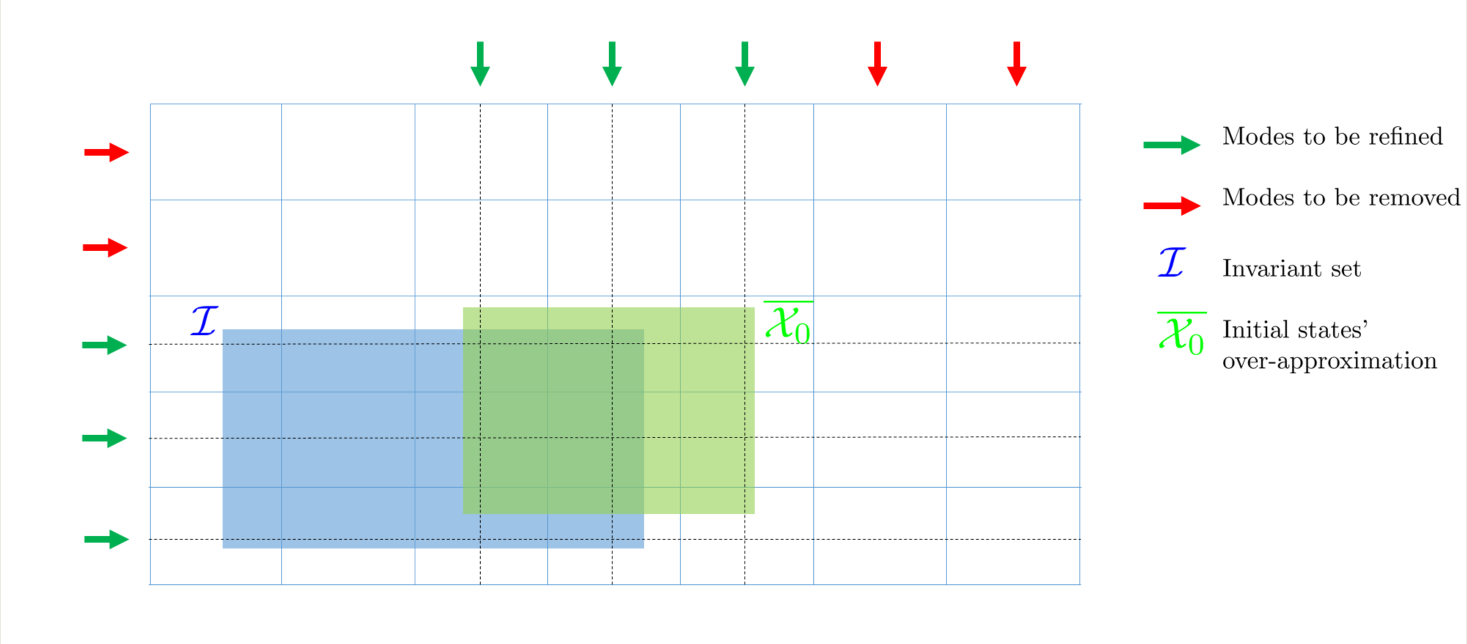


Guided Refinement

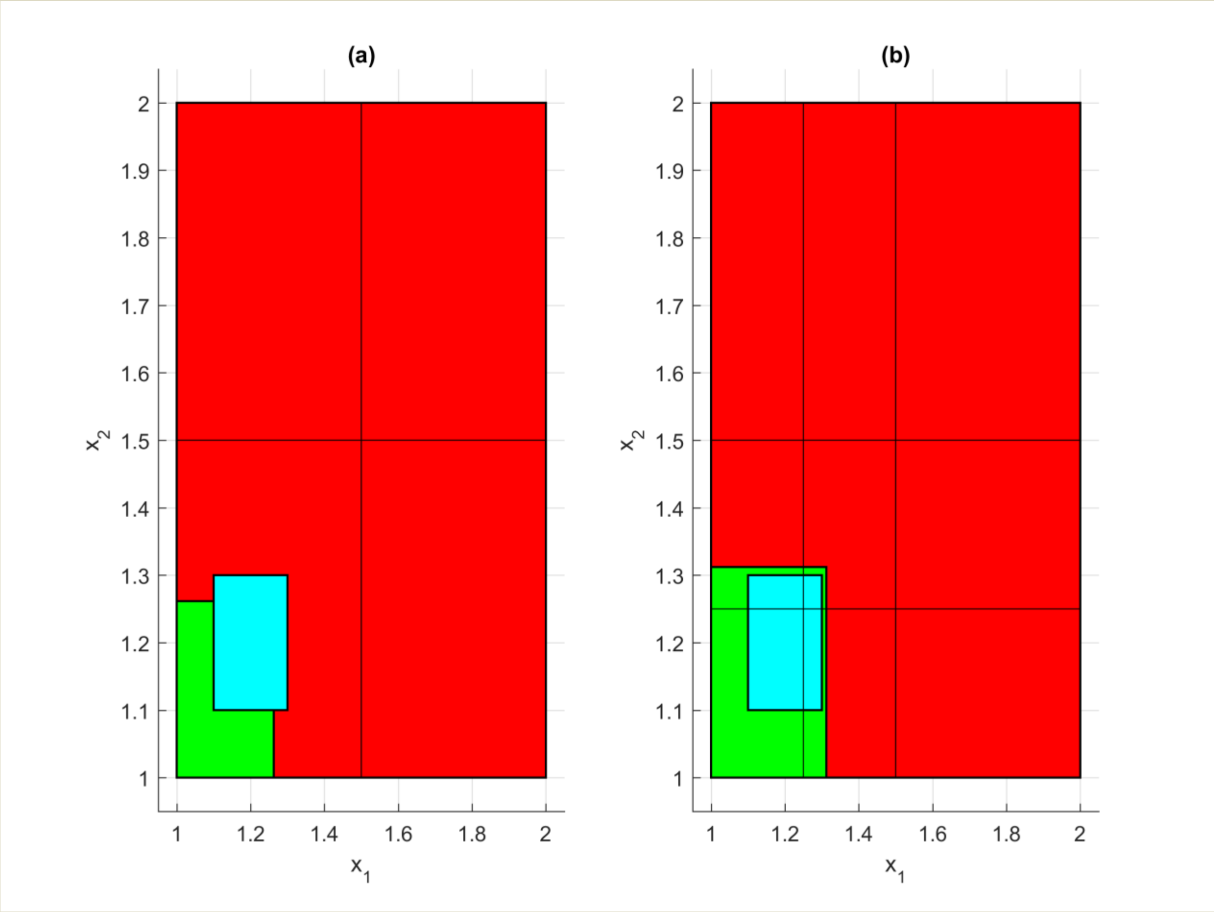


Guided Refinement

- Refine the modes where invariant set does not cover initial states.
- Remove (Merge) modes that neither intersect with invariant set, nor initial states.

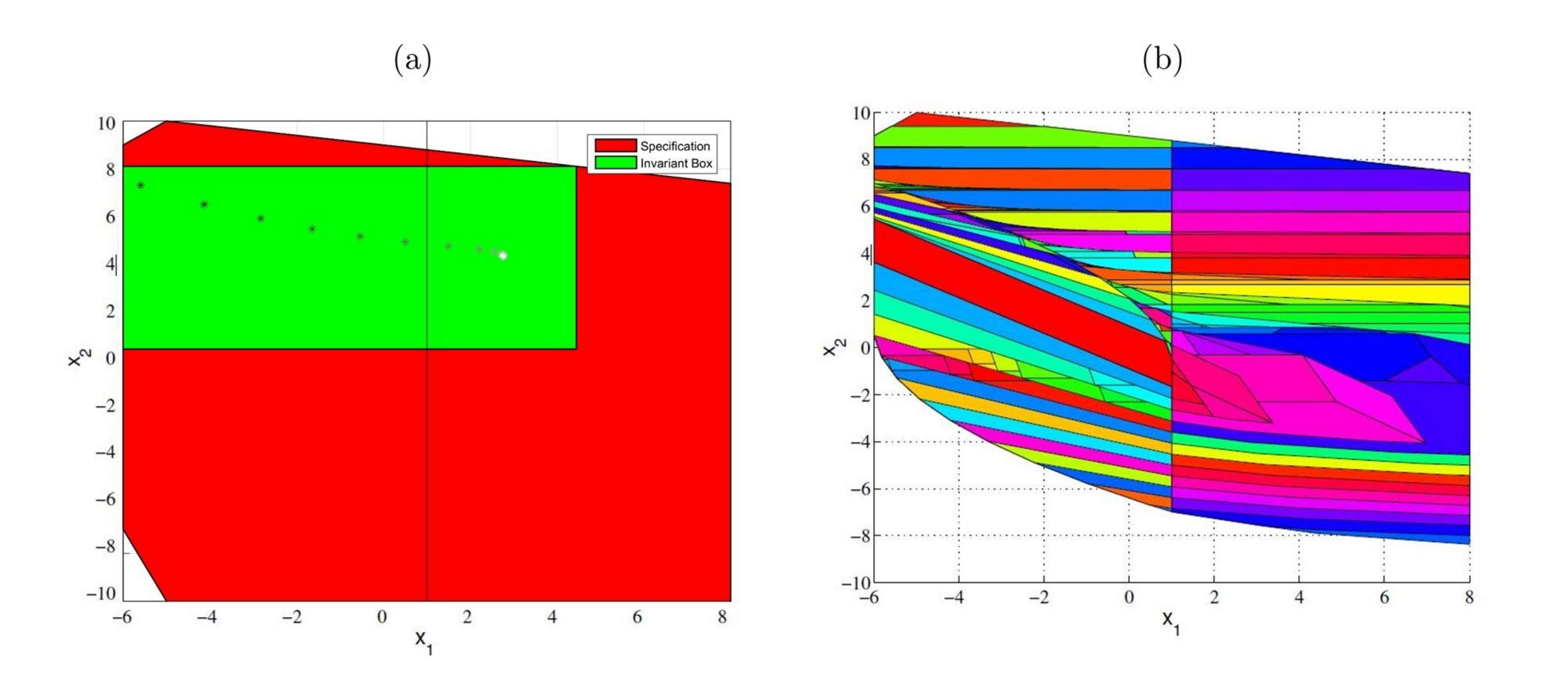


Guided Refinement



Performance and Computation Cost

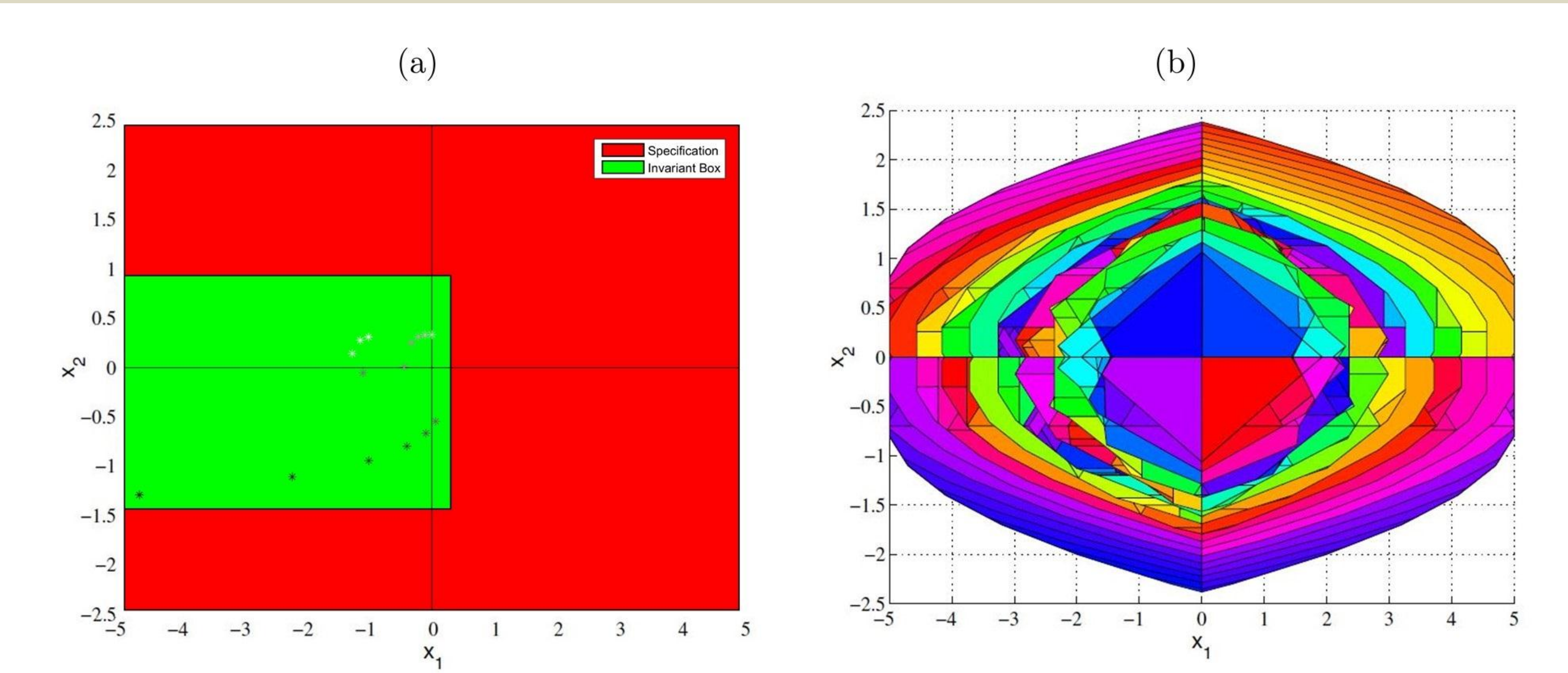
45ms **75s**



S. Rakovic, P. Grieder, M. Kvasnica, D. Mayne, and M. Morari, “Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances,”

Performance and Computation Cost

120ms **107s**



S. Rakovic, P. Grieder, M. Kvasnica, D. Mayne, and M. Morari, “Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances,”

Safety control

Achieved results

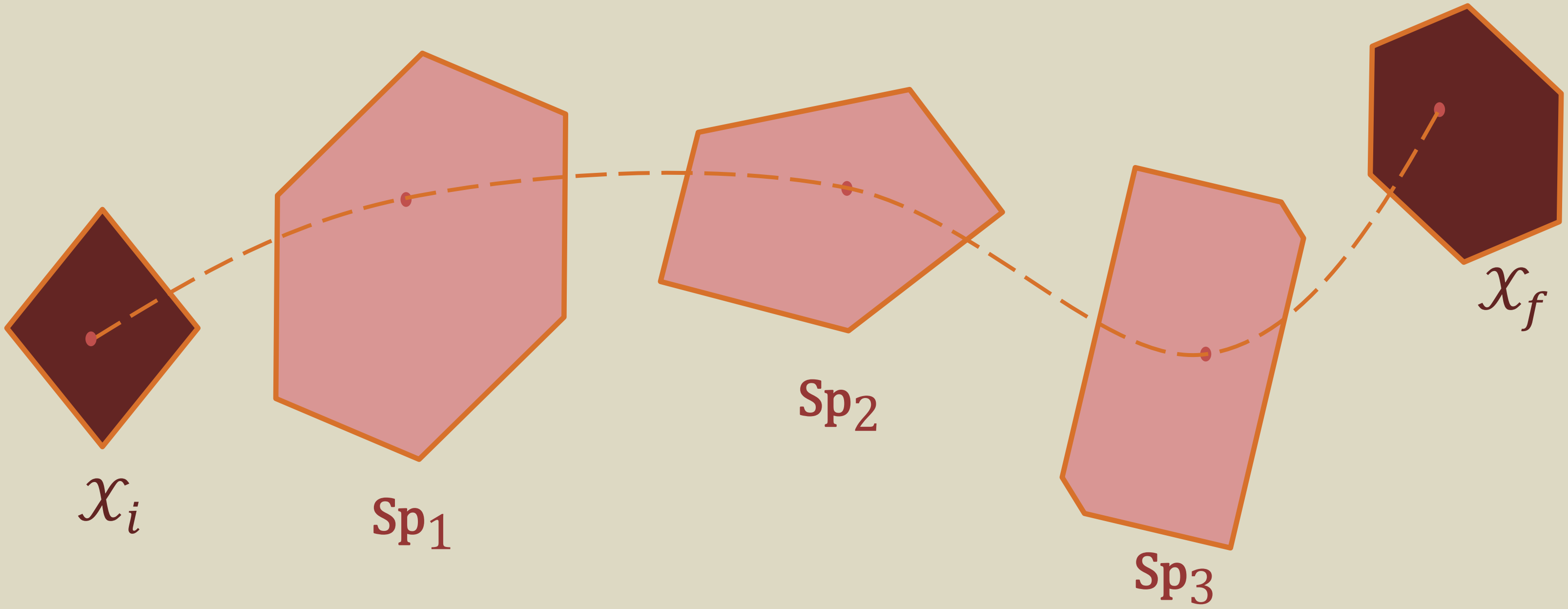
- Refinement inclusion
- Computation efficiency through rectangular sets
- State space transformation for improved results

Future direction

- Invariant sets as box collections
- Heuristics for abstraction refinement
- Stop criterion definition

Problem: Control under specs

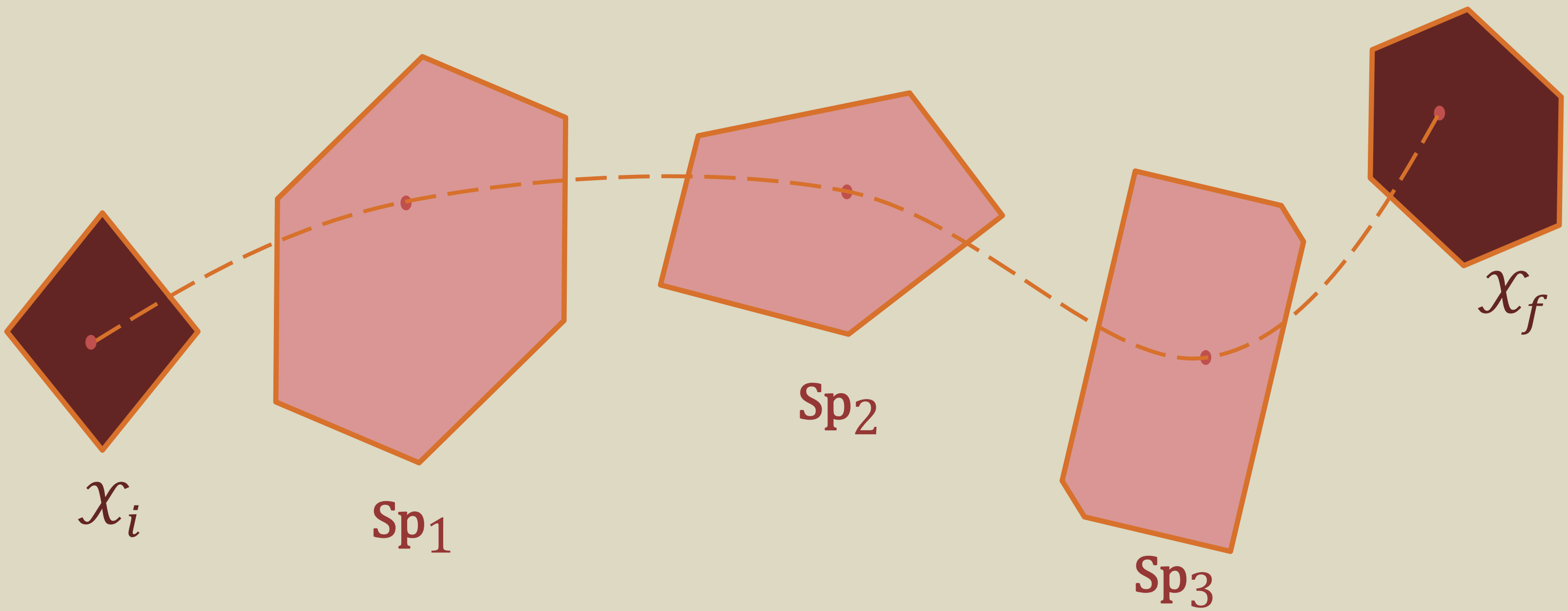
design a controller that steers the state of the system to some target region while keeping it within some possibly time-varying set along the way (safety specs)



POLITECNICO MILANO 1863

Problem: Control under specs

design a controller that steers the state of the system to some target region while keeping it within some possibly time-varying set along the way (safety specs)



No control input → safety verification problem

PWA model

$$\mathbf{x}(k+1) = A_i\mathbf{x}(k) + B_{Ui}\mathbf{u}(k) + B_{Wi}\mathbf{w}(k) + \mathbf{f}_i \quad \text{if } \mathbf{x}(k) \in \mathcal{P}_i, \quad i \in \mathbb{N}_s$$

Assumptions

Modes are polyhedral and constitute a partition of the state space

The system is affected by an additive bounded disturbance: $\mathbf{w}(k) \in [\mathbf{w}_{\min}, \mathbf{w}_{\max}]$

The initial state is uncertain: $\mathbf{x}(0) \in \mathcal{X}_i$ where $\mathcal{X}_i$ is a zonotope

Problem: Control under specs

Choose $\mathrm{u}(k)$ as a function of $\mathrm{x}(k)$ so as to robustly satisfy:

Actuation constraints

- $\mathbf{u}(k) \in [\mathbf{u}_{\min}, \mathbf{u}_{\max}] \quad \forall k \in [0, N)$

Safety & target specifications

- $\mathbf{x}(k) \in \mathbf{Sp}_k \quad \forall k \in (0, N]$

Proposed solution

Set-based approach

Reference trajectory

State-feedback controller

Mode recovery procedure

Reference trajectory computation

Open-loop control problem formulation for the nominal PWA system in the horizon [0,N]

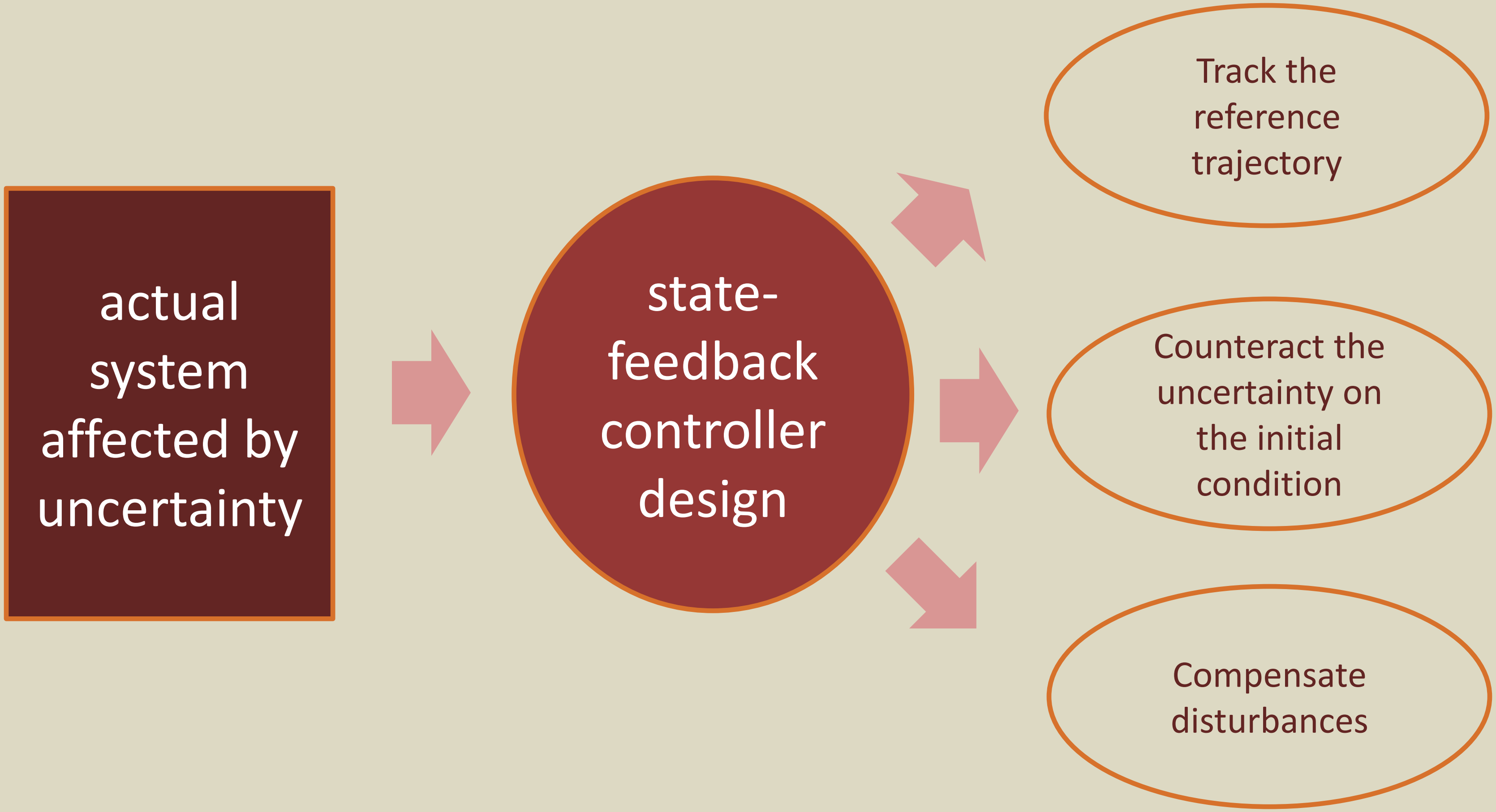
Mixed Integer Linear Program (MILP) solution

Optimization problem

Satisfy the specs and maximize the distance of the state from:

- the modes boundaries
- the safety specs boundaries

State-feedback controller design



State-feedback controller design

actual system affected by uncertainty

state-feedback controller design

Track the reference trajectory

Counteract the uncertainty on the initial condition

Compensate disturbances

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}(\mathbf{c}_Z) + \sum_{i=1}^{p} \alpha_i(\mathbf{x})\mathbf{u}(\mathrm{G}_{Z(i)})$$

Control law implementation

What is pre-computed offline?

- reference, possibly branched, trajectory and controlled reach sets associated with it
- control actions applied to centers and generators of each reach set

Control law implementation

What is pre-computed offline?

- reference, possibly branched, trajectory and controlled reach sets associated with it
- control actions applied to centers and generators of each reach set

What is computed online?

1. Determine to which reach set the current state belongs to and its α coefficients
2. Recover the control actions associated with center and generators of that reach set
3. Compute the control action associated to the given state value based on its α coefficients

Control law implementation

What is pre-computed offline?

- Reference, possibly branched, trajectory and controlled reach sets associated with it
- control actions applied to centers and generators of each reach set

MILP

What is computed online?

1. Determine to which reach set the current state belongs to and its α coefficients
2. Recover the control actions associated with center and generators of that reach set
3. Compute the control action associated to the given state value based on its α coefficients

LP

Numerical examples

Full mode recovery

- A reach set splits
- All of its subsets are steered back to the main mode in one step (mode recovery is successful, no branching)

Mode recovery failure

- A reach set splits
- A new reference trajectory is generated from one of its parts (mode recovery fails, branching)

Full mode recovery

$$\mathbf{x}(k+1) = A_i\mathbf{x}(k) + B_{Ui}\mathbf{u}(k) + B_{Wi}\mathbf{w}(k) + \mathbf{f}_i \quad \text{if } \mathbf{x}(k) \in \mathcal{P}_i, \quad i \in \mathbb{N}_s$$

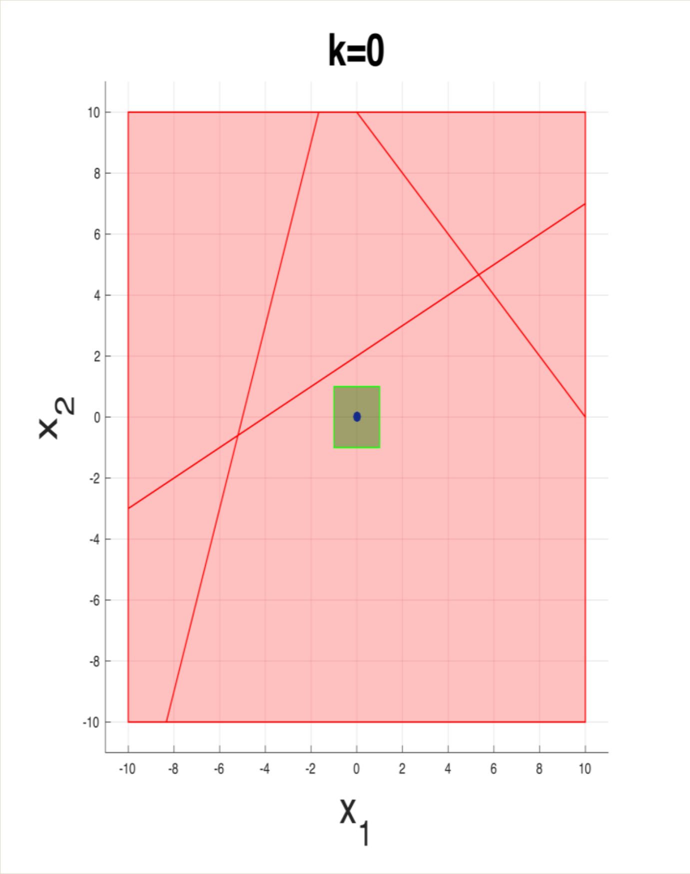
$$\mathbf{u}(k) \in [-1,1]^2 \qquad \mathbf{w}(k) \in [-1,1]^2 \qquad M = 3 \quad N = 6 \quad s = 6$$

$$A_1 = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} \end{pmatrix} \quad A_2 = \begin{pmatrix} -1 & 1 \\ \frac{4}{5} & -1 \end{pmatrix} \quad A_3 = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}$$

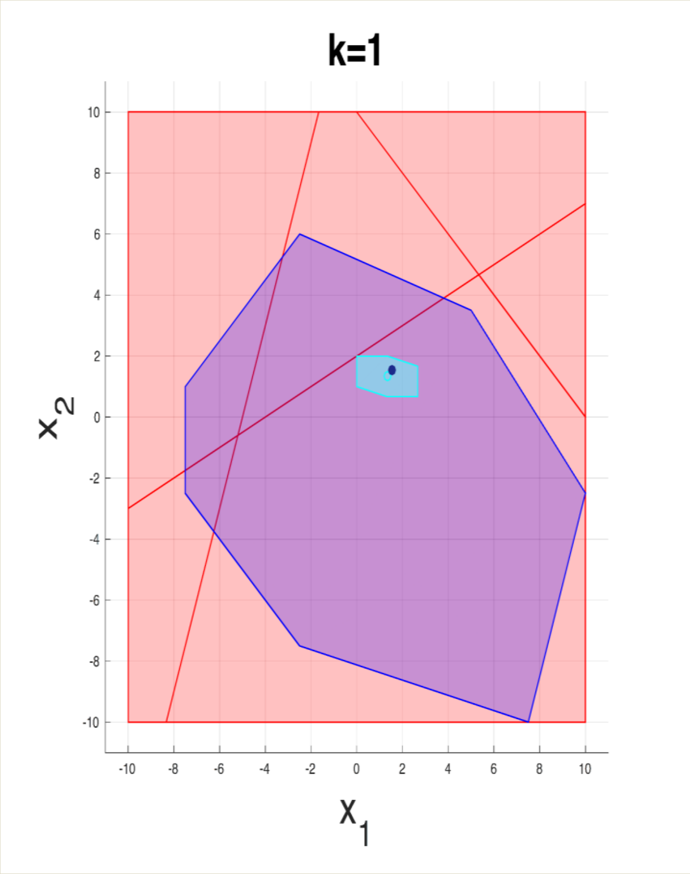
$$A_4 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad A_5 = \begin{pmatrix} \frac{1}{3} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{3} \end{pmatrix} \quad A_6 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$B_{Uj} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad B_{Wj} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \quad \mathbf{f}_j = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \forall j = 1,2,\dots,6$$

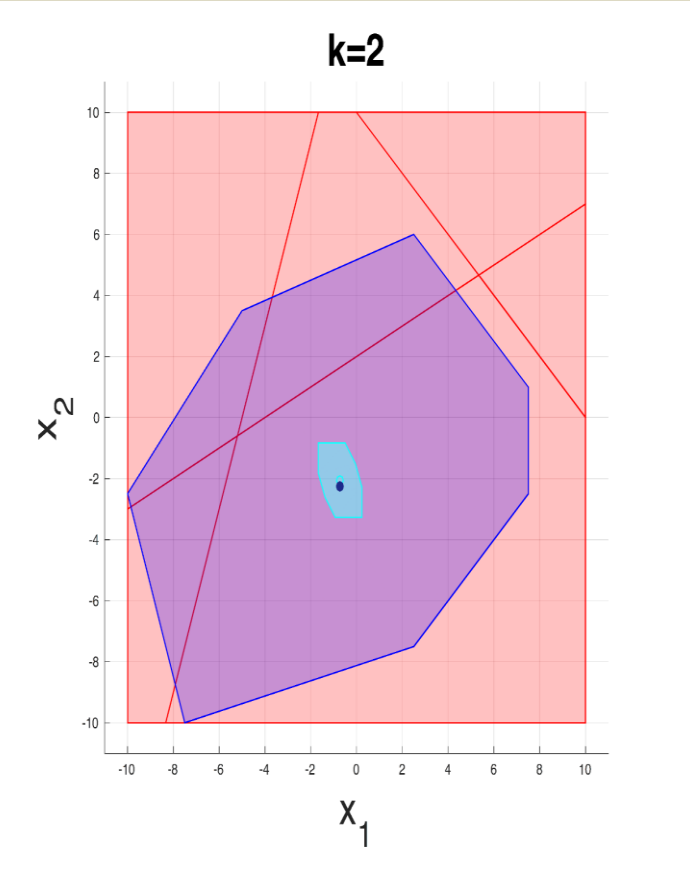
Full mode recovery



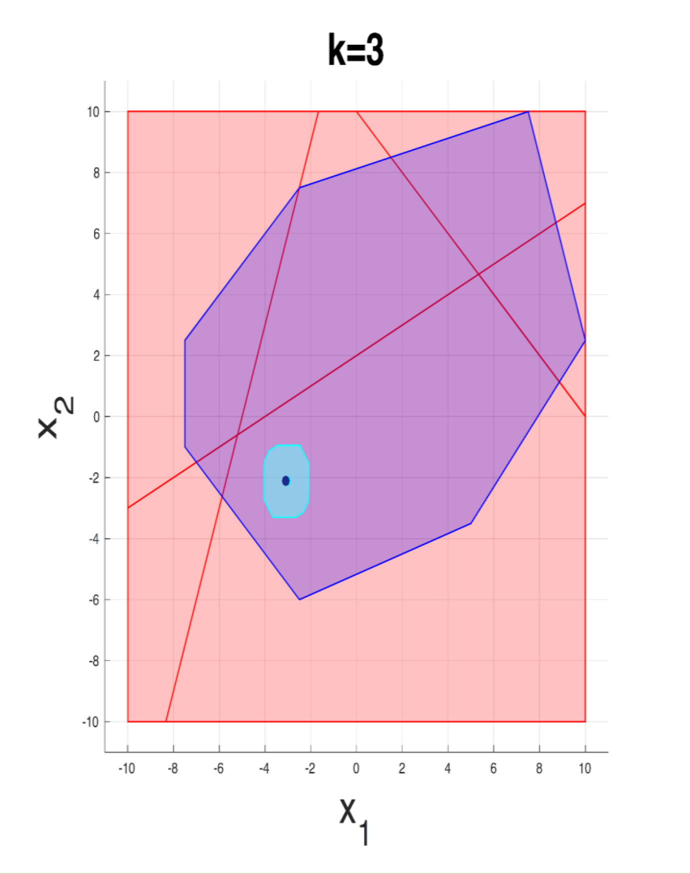
Full mode recovery



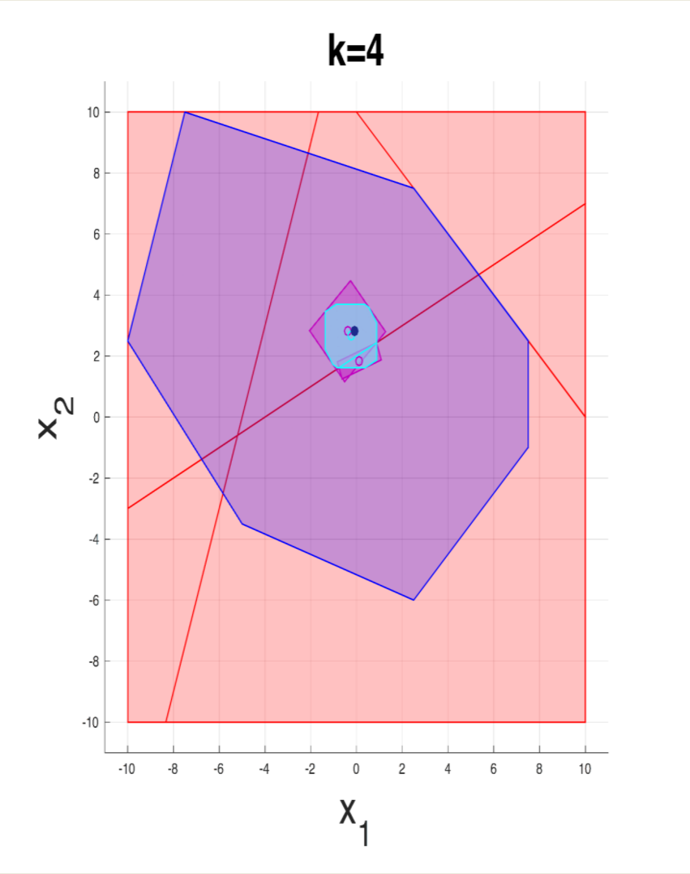
Full mode recovery

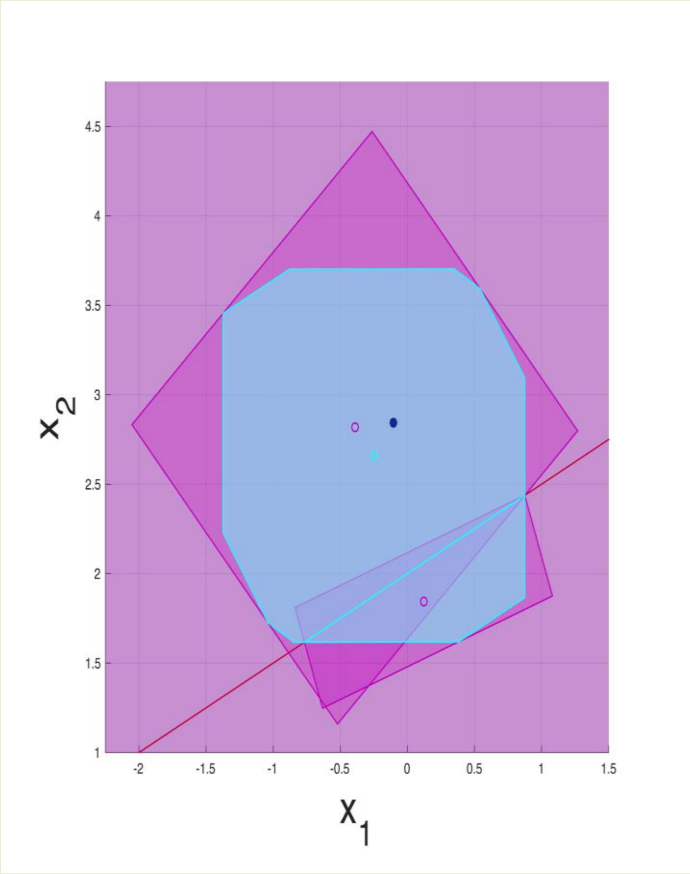


Full mode recovery

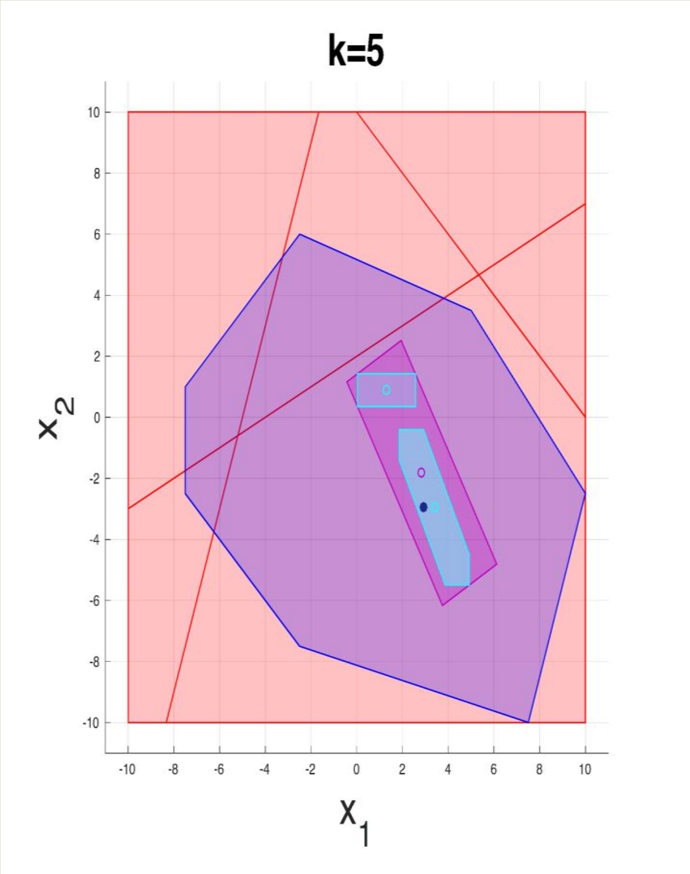


Full mode recovery

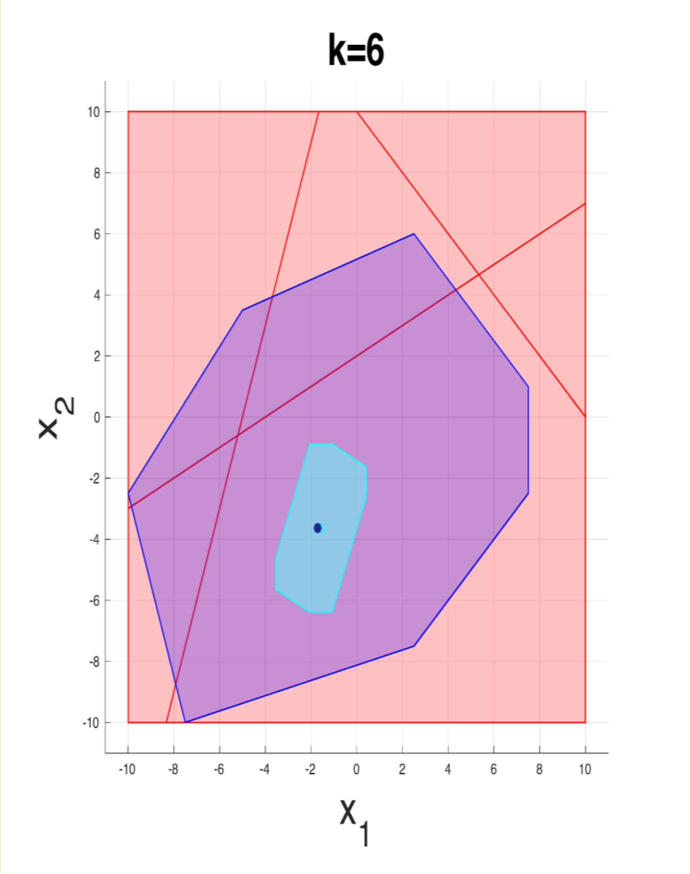




Full mode recovery



Full mode recovery



Mode recovery failure

$$\mathbf{x}(k+1) = A_i\mathbf{x}(k) + B_{Ui}\mathbf{u}(k) + B_{Wi}\mathbf{w}(k) + \mathbf{f}_i \quad \text{if } \mathbf{x}(k) \in \mathcal{P}_i, \quad i \in \mathbb{N}_s$$

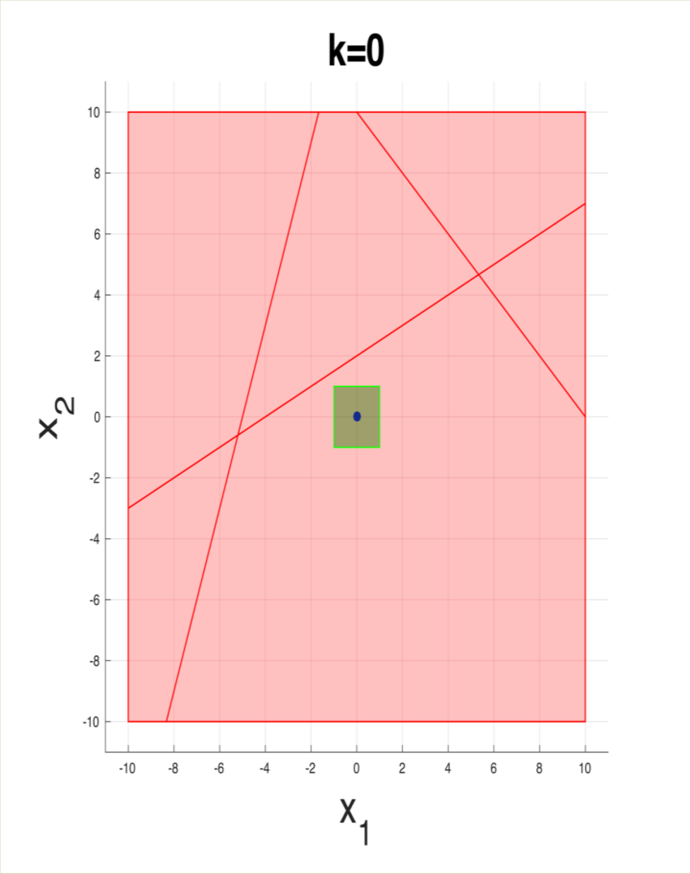
$$\mathbf{u}(k) \in [-10,10]^2 \qquad \mathbf{w}(k) \in [-2,2]^2 \qquad M = 3$$

$$A_1 = \begin{pmatrix} \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{1}{2} \end{pmatrix} \quad A_2 = \begin{pmatrix} -1 & 1 \\ \frac{4}{5} & 1 \end{pmatrix} \quad A_3 = \begin{pmatrix} 1 & 2 \\ 2 & -1 \end{pmatrix}$$

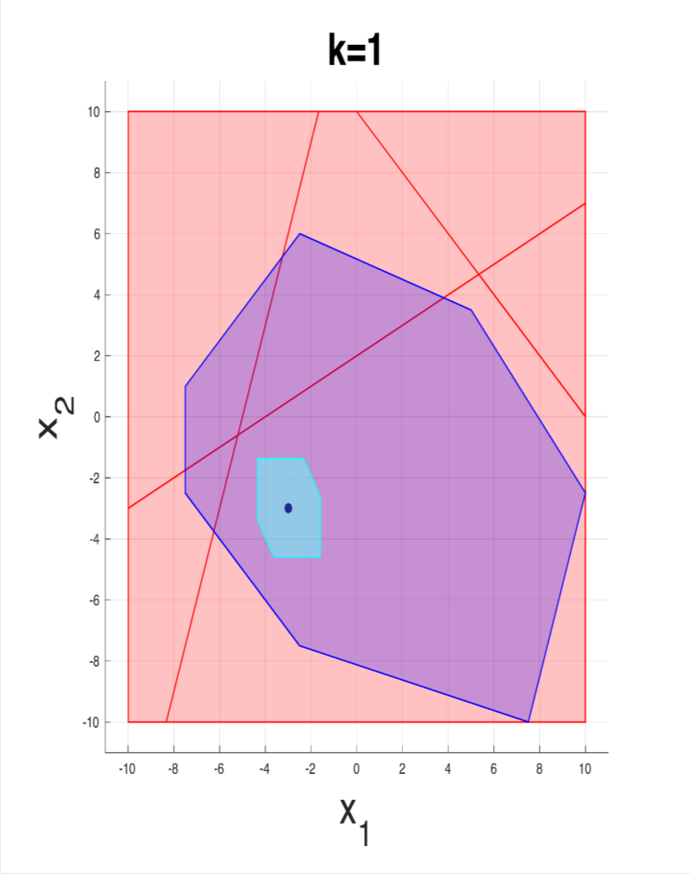
$$A_4 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad A_5 = \begin{pmatrix} \frac{1}{3} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{3} \end{pmatrix} \quad A_6 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$B_{Uj} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad B_{Wj} = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \quad \mathbf{f}_j = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \forall j = 1,2,\dots,6$$

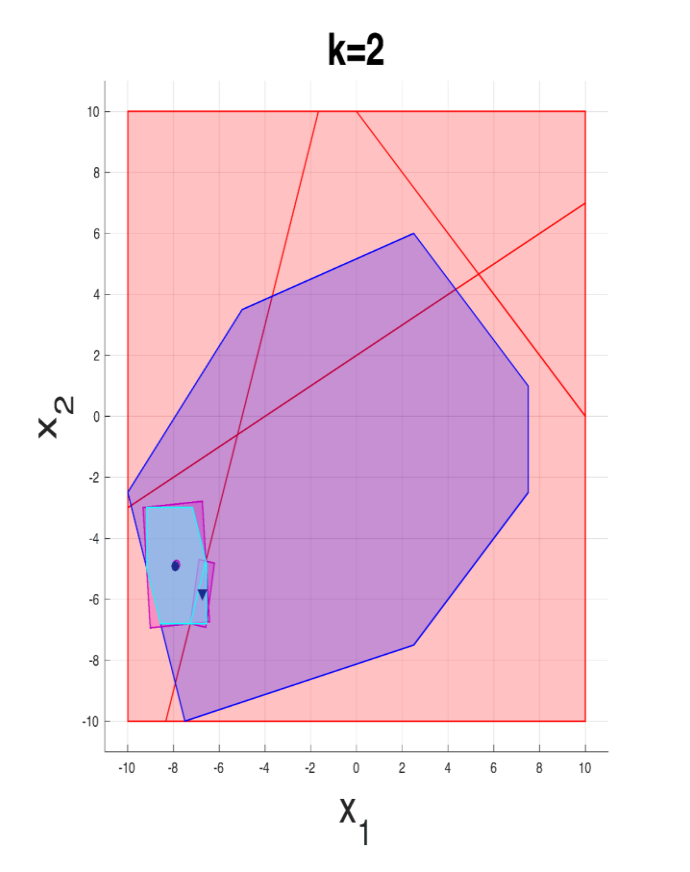
Mode recovery failure

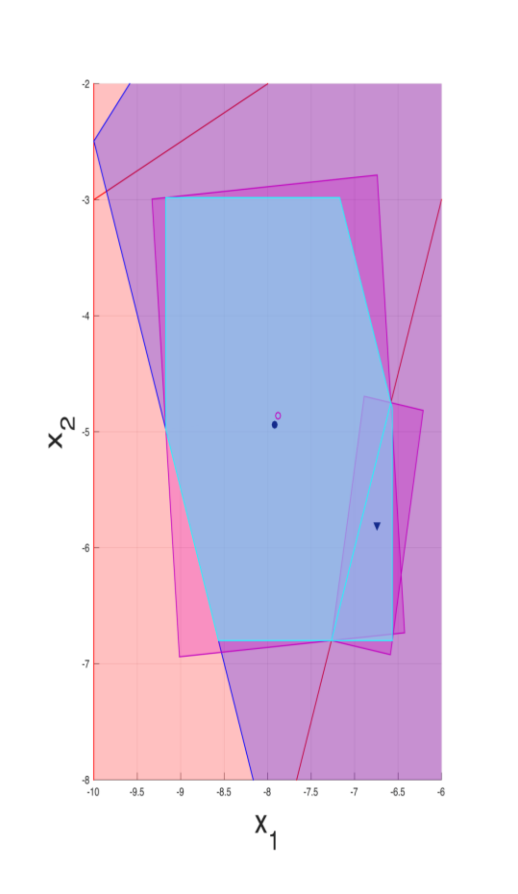


Mode recovery failure

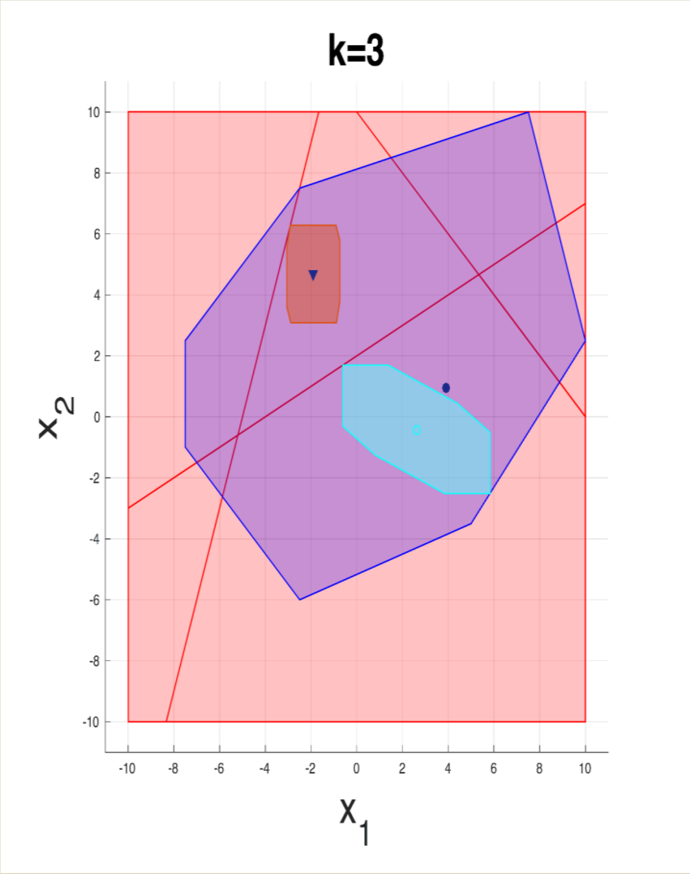


Mode recovery failure

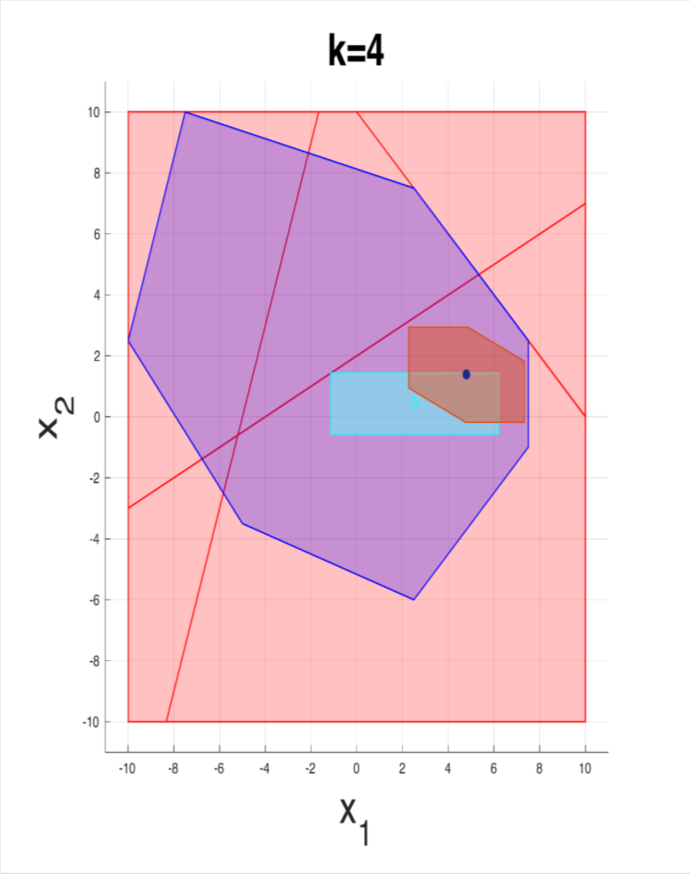




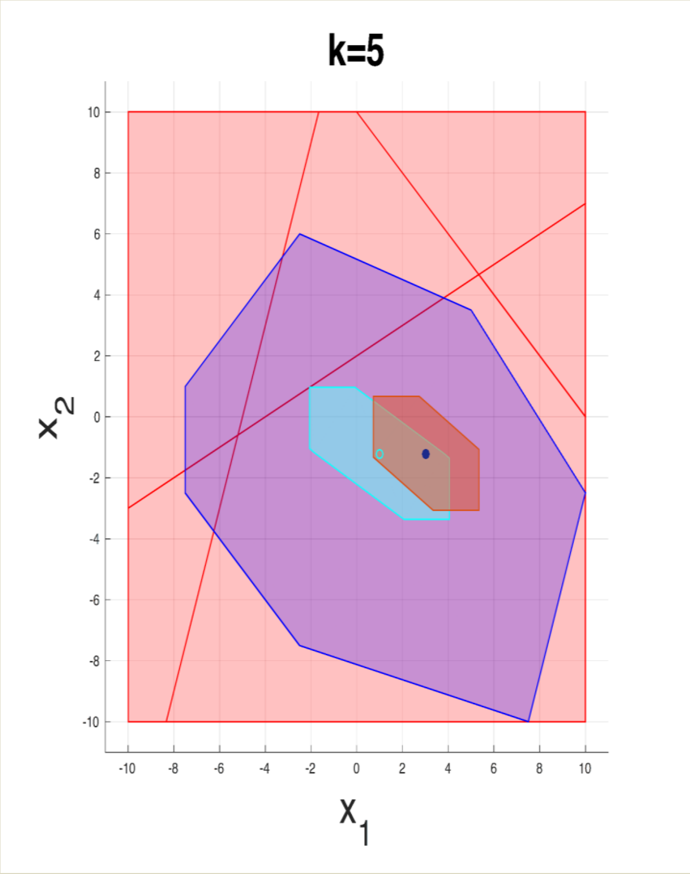
Mode recovery failure



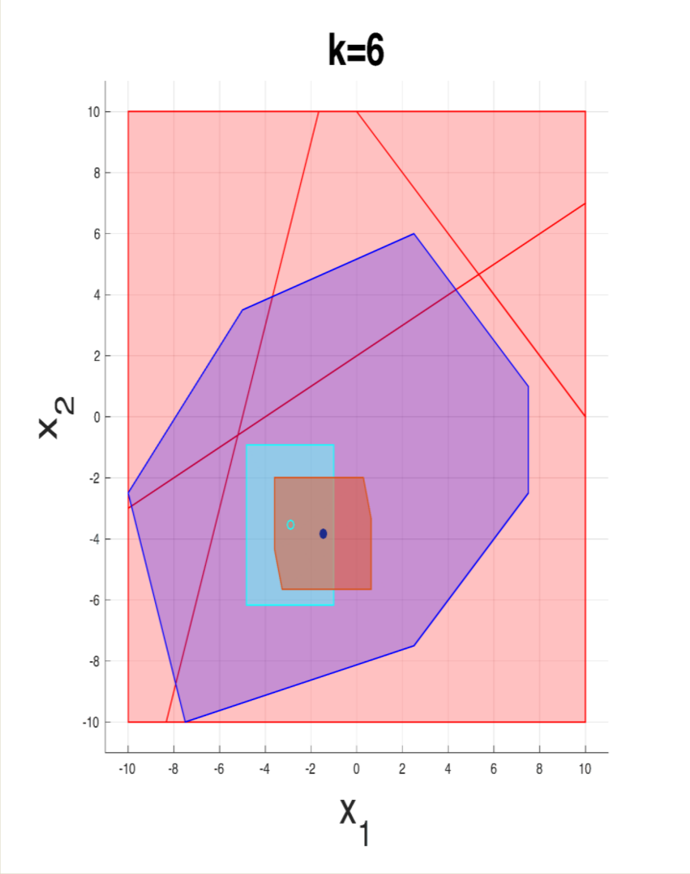
Mode recovery failure



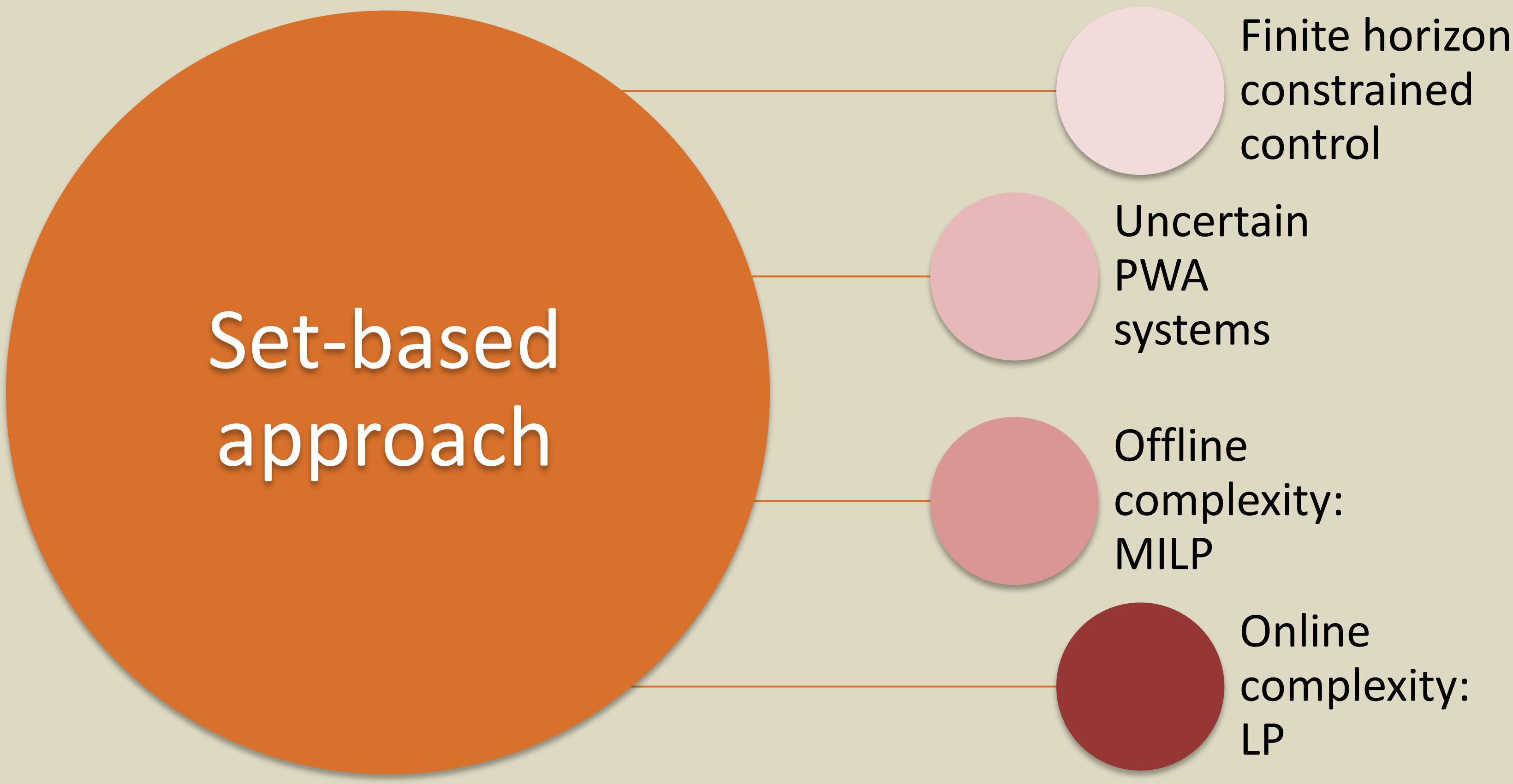
Mode recovery failure



Mode recovery failure



Enforcing bounded safety



Main references & credit

Master theses, Automation and Control Engineering, Politecnico di Milano:

- Verification of nonlinear systems through hybridization and invariant set computation
  Pouria Tajvar, 2017
- Robust constrained control of piecewise affine systems based on reach sets computation
  Riccardo Desimini, 2017

PhD thesis, Information Technology, Systems and Control area, Politecnico di Milano:

- Automatic verification and input design for dynamical systems: an optimization based approach to the detection of non-influential inputs
  Riccardo Vignali, 2015

Main references & credit

Master theses, Automation and Control Engineering, Politecnico di Milano:

- Verification of nonlinear systems through hybridization and invariant set computation
  Pouria Tajvar, 2017
- Robust constrained control of piecewise affine systems based on reach sets computation
  Riccardo Desimini, 2017

PhD thesis, Information Technology, Systems and Control area, Politecnico di Milano:

- Automatic verification and input design for dynamical systems: an optimization based approach to the detection of non-influential inputs
  Riccardo Vignali, 2015

Acknowledgements

UnCoVerCPS project 2015-18

Unifying Control and Verification of Cyber-Physical Systems

Horizon 2020 research and innovation programme

Grant agreement number 643921

Thank you for your attention!