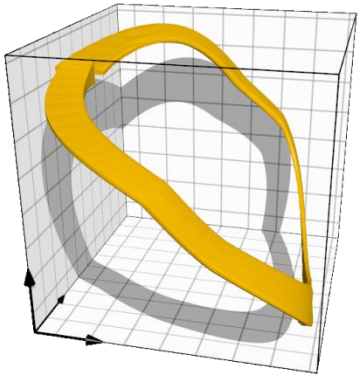
Verification of Hybrid Systems



Antoine Girard
Université Grenoble 1, Laboratoire Jean Kuntzmann

- with work from Thao Dang, Goran Frehse and Colas Le Guernic -

Ecole des JDMACS, 19 mars, 2009

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES – INFORMATIQUE

UNIVERSITE JOSEPH FOURIER
SCIENCES.TECHNOLOGIE.SANTÉ
GRENOBLE 1

Acknowledgments

- **Organizers: Luc Jaulin, Nacim Ramdani.**

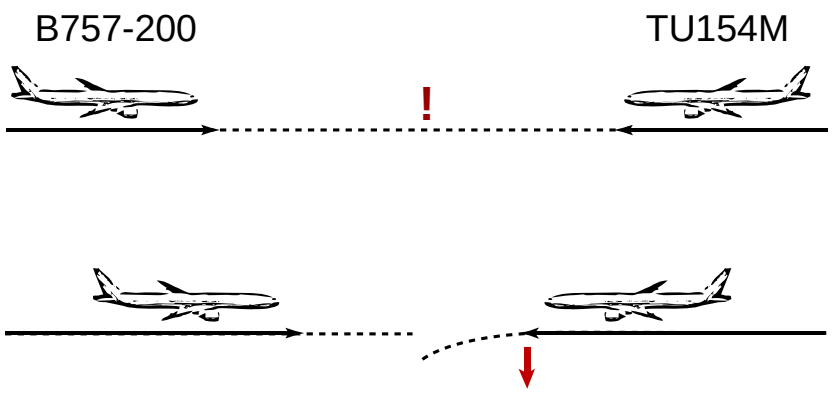
- **Collaborators:**
  - Thao Dang,
  - Goran Frehse,
  - Colas Le Guernic.

- **Special thanks to Goran Frehse for putting up together the first version of the slides.**

Boeing & Tupolew Collision

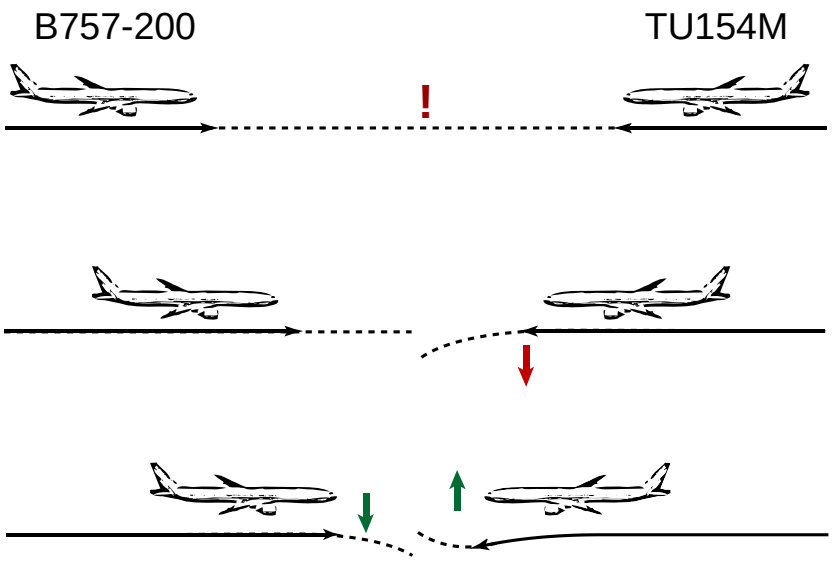
- **Überlingen, July 1, 2002**
- **21:33:03**
  - Alarm from Traffic Collision Avoidance System (TCAS)

Boeing & Tupolew Collision



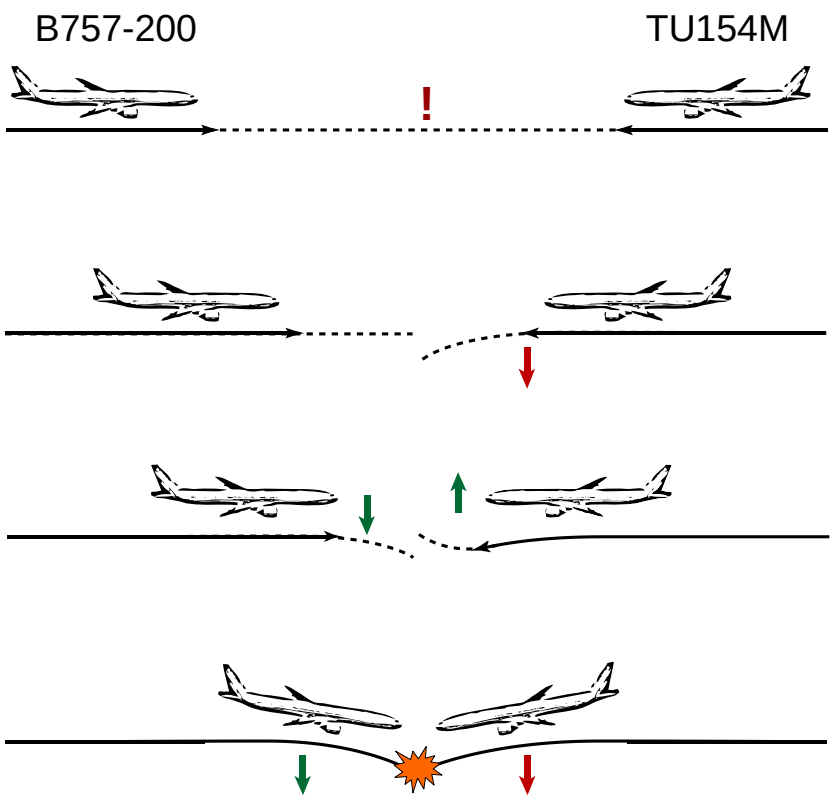
- **Überlingen, July 1, 2002**
- **21:33:03**
  - Alarm from Traffic Collision Avoidance System (TCAS)
- **21:34:49**
  - Human air traffic controller command

Boeing & Tupolew Collision



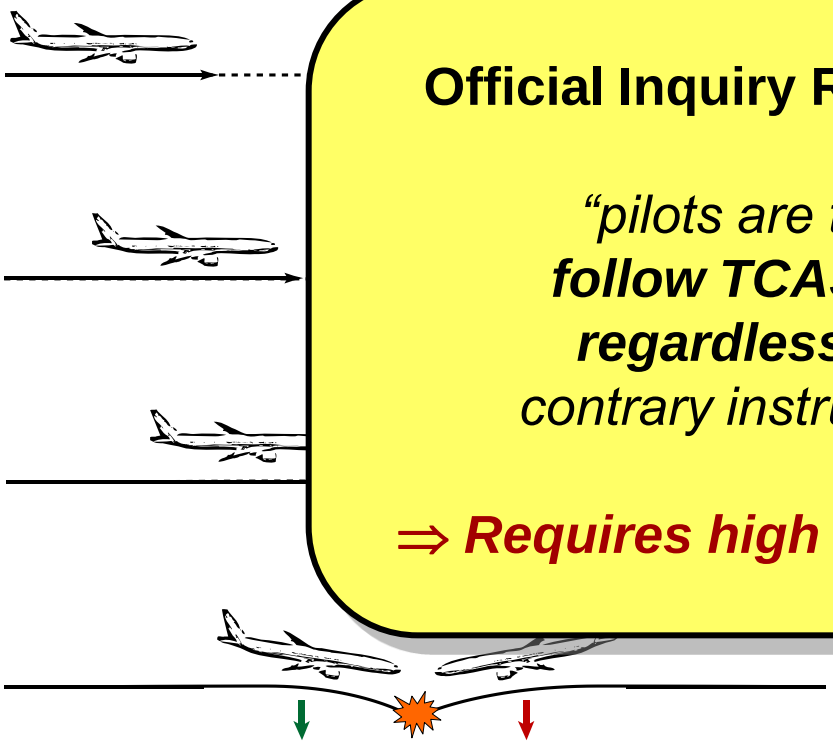
- **Überlingen, July 1, 2002**
- **21:33:03**
  - Alarm from Traffic Collision Avoidance System (TCAS)
- **21:34:49**
  - Human air traffic controller command
- **21:34:56**
  - TCAS recommendation

Boeing & Tupolew Collision



- **Überlingen, July 1, 2002**
- **21:33:03**
  - Alarm from Traffic Collision Avoidance System (TCAS)
- **21:34:49**
  - Human air traffic controller command
- **21:34:56**
  - TCAS recommendation
- **21:35:32**
  - Collision

Boeing & Tupolew Collision



- **Überlingen, July 1, 2002**

Official Inquiry Recommendation:

"pilots are to obey and
***follow TCAS** advisories,*
***regardless** of whether*
contrary instruction is given"

⇒ ***Requires high confidence design***

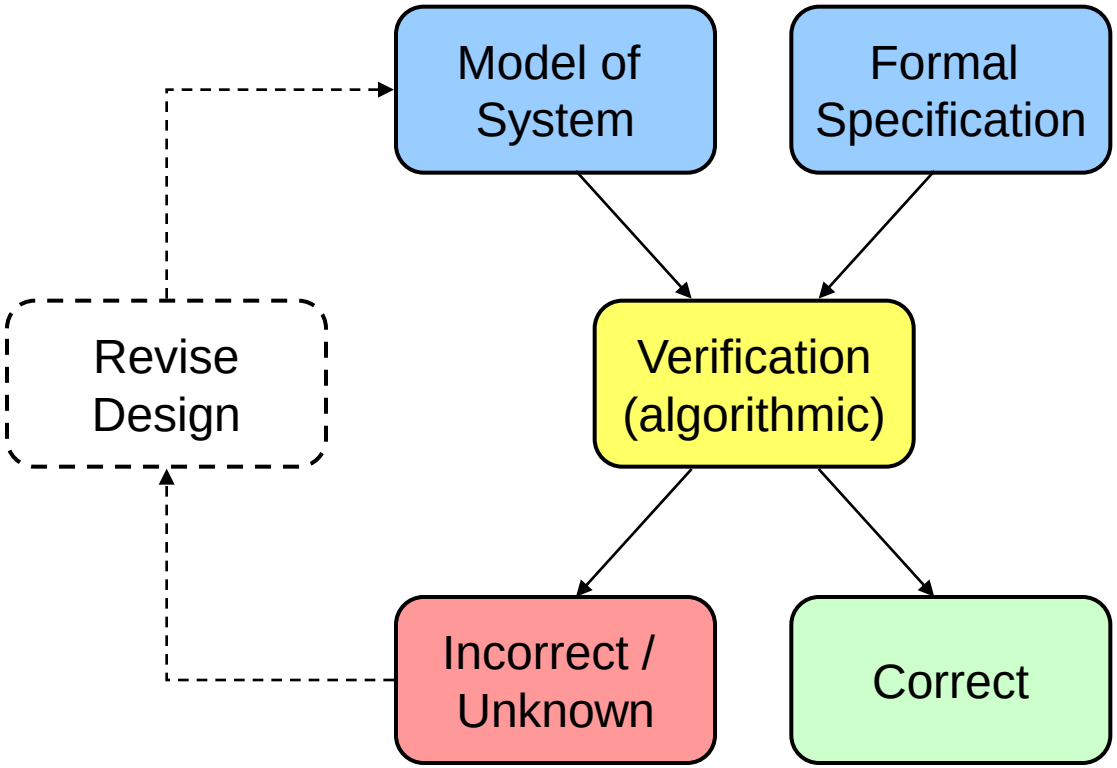
c Collision
m (TCAS)

controller

ndation

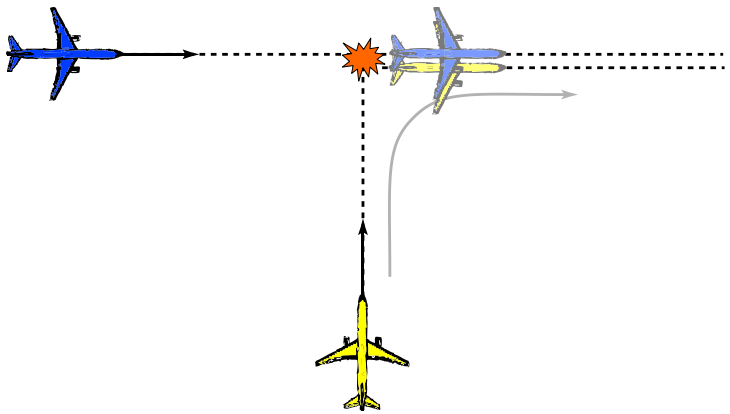
- **21:35:32**
  - Collision

Formal Verification



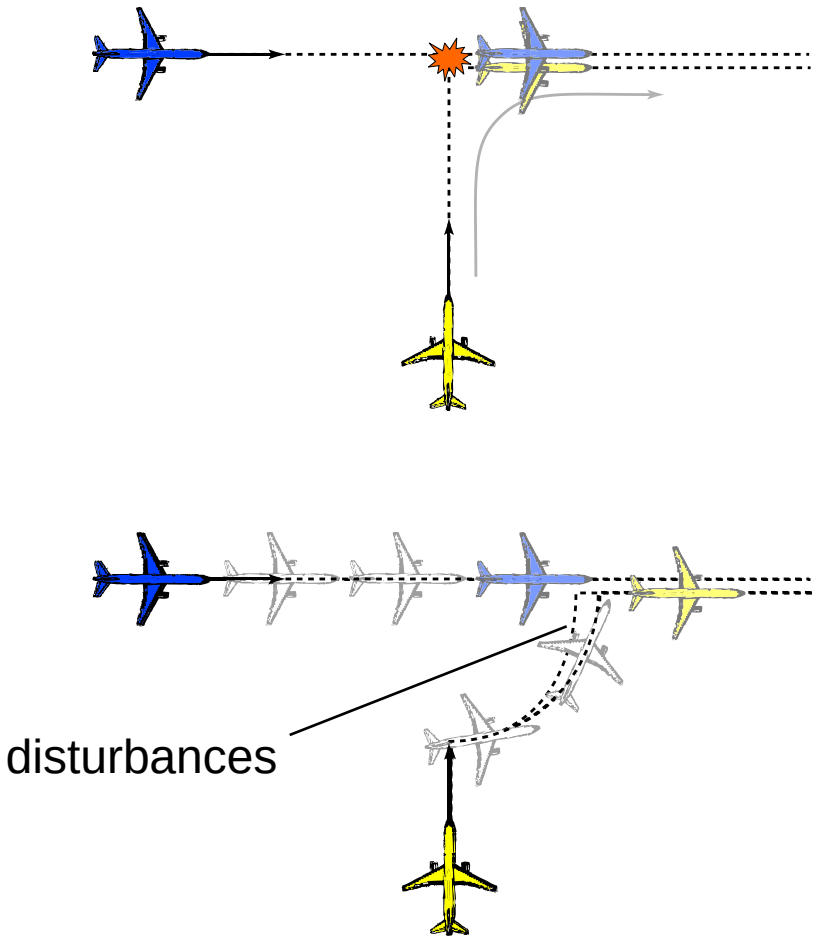
TCAS verified in part
[Livadas, Lygeros, Lynch, ’00]

Join Maneuver [Tomlin et al.]



- **Traffic Coordination Problem**
  - join paths at different speed
- **Goals**
  - avoid collision
  - join with sufficient separation

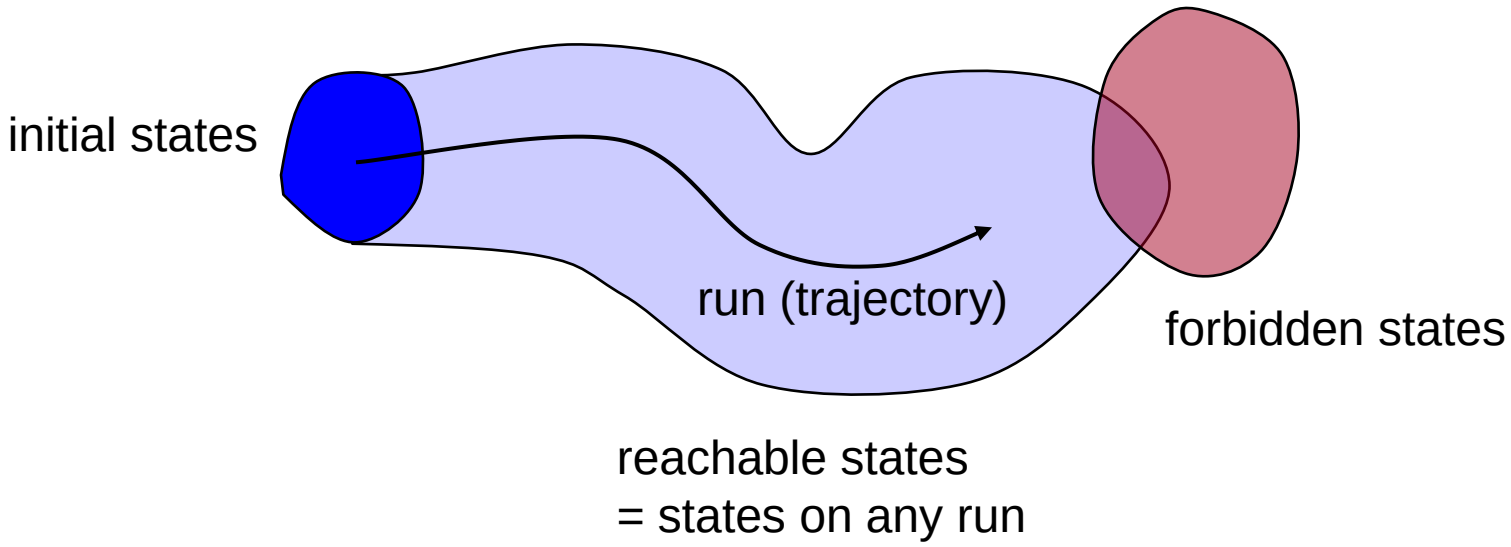
Join Maneuver [Tomlin et al.]



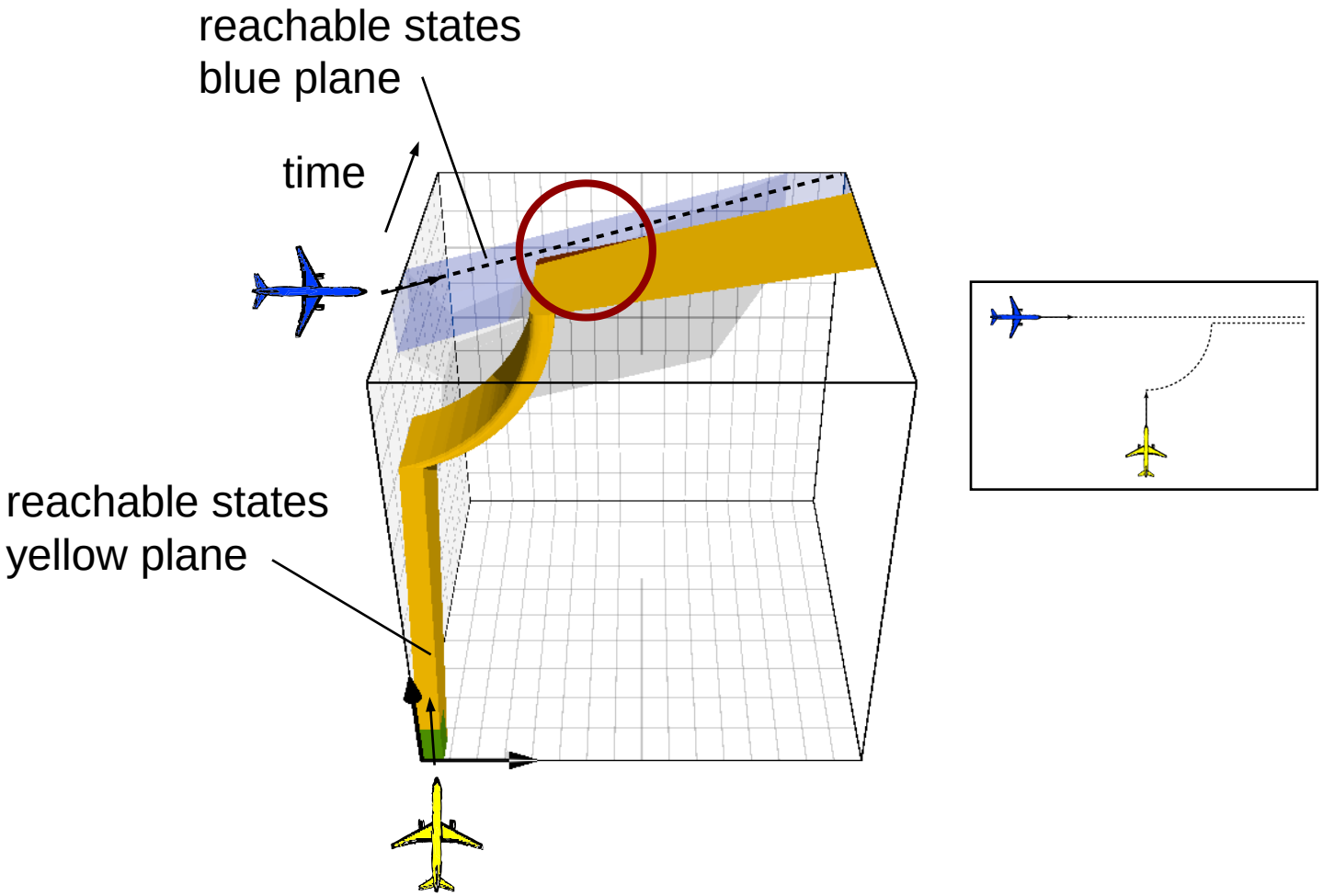
- **Traffic Coordination Problem**
  - join paths at different speed
- **Goals**
  - avoid collision
  - join with sufficient separation
- **Models**
  - Environment: Planes
  - Software: Controller
    - switches fast/slow
- **Specification**
  - keep min. distance

Formal Verification

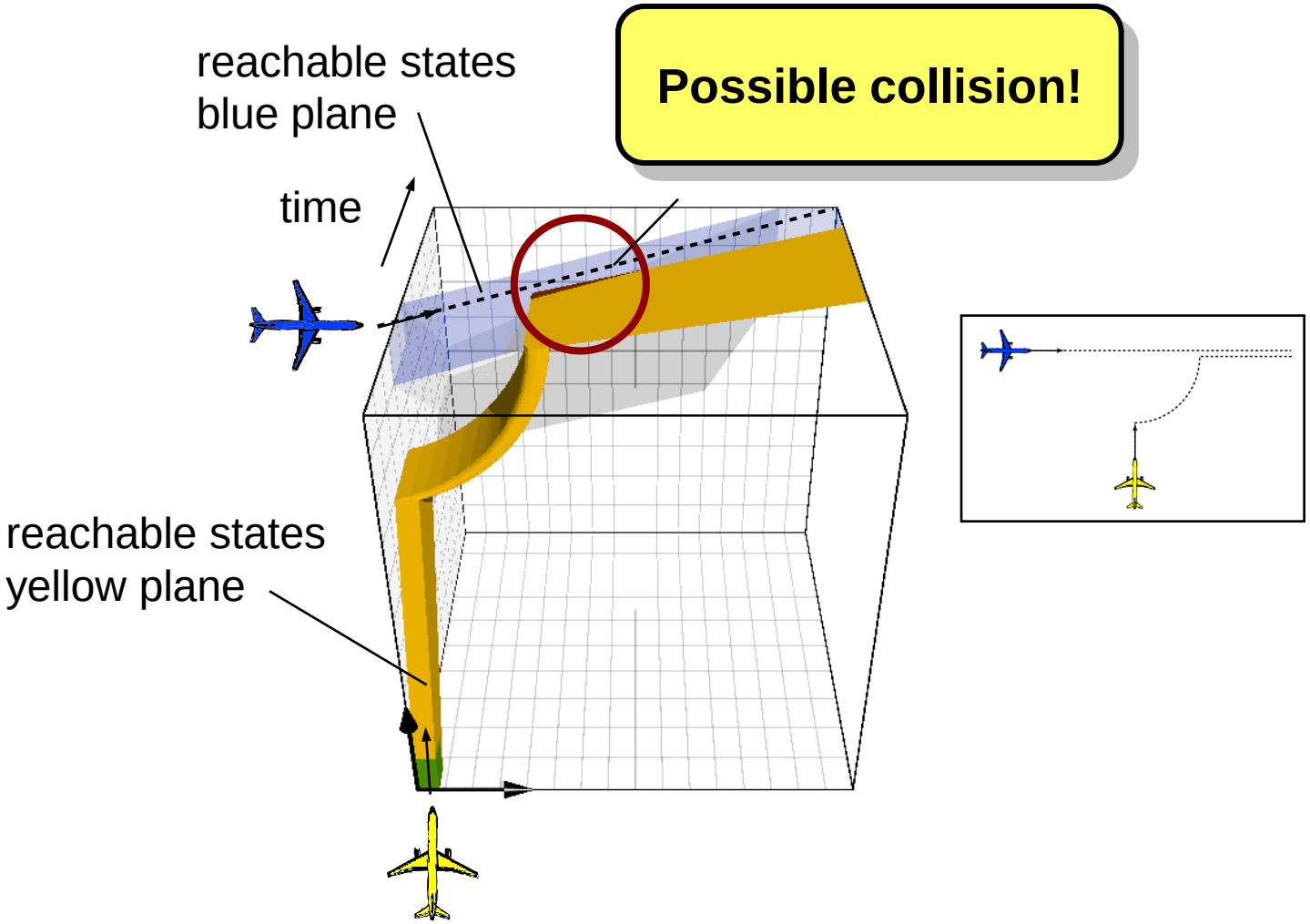
- **Characteristics**
  - mathematical rigor (sound proofs & algorithms)
  - exhaustive
- **In this talk: Reachability Analysis**



Join Maneuver [Tomlin et al.]



Join Maneuver [Tomlin et al.]



Formal Verification

- **Key Problems**
  - computable (decidable) only for simple dynamics
  - computationally expensive
  - representation of / computation with continuous sets

Formal Verification

- **Fighting complexity with overapproximations**
  - simplify dynamics
  - set representations
  - set computations
- **Overapproximations should be**
  - conservative
  - easy to derive and compute with
  - accurate (not too many false positives)

Outline

I. **Hybrid Automata and Reachability**

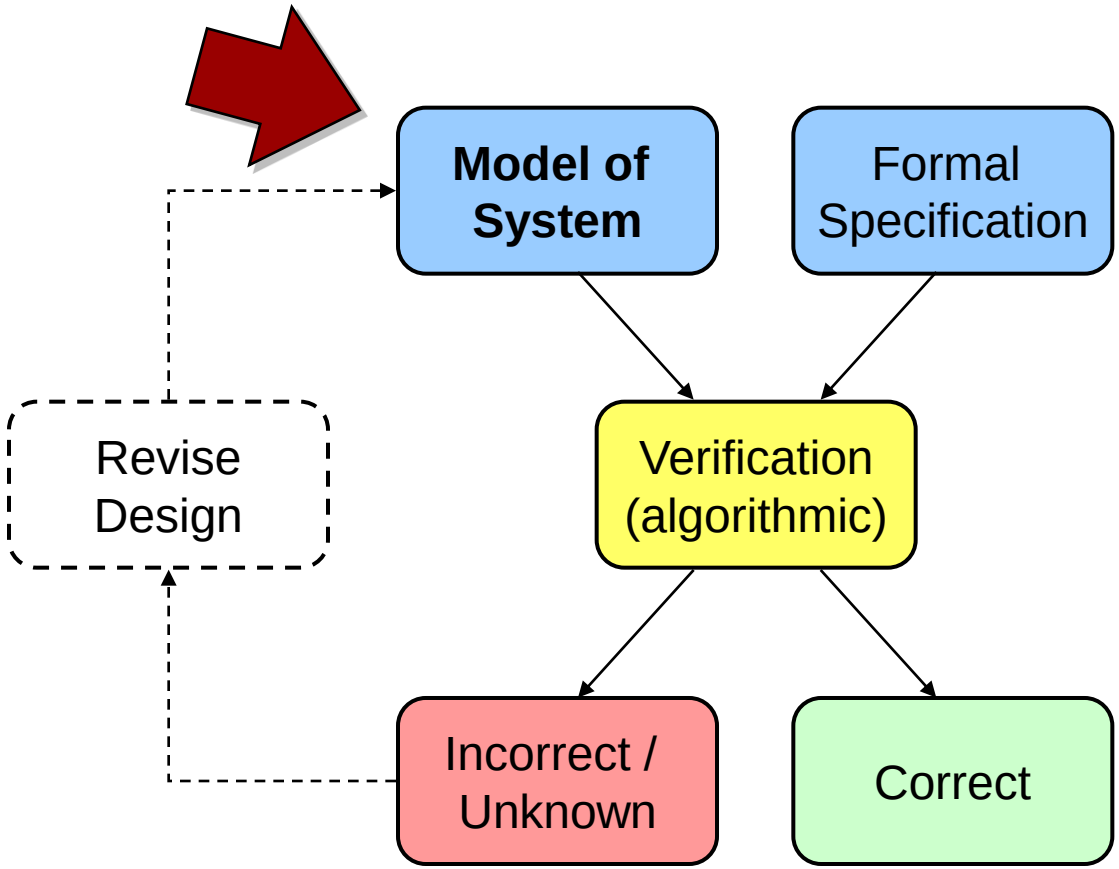
II. **Reachability for Simple Dynamics**

   a) Linear Hybrid Automata

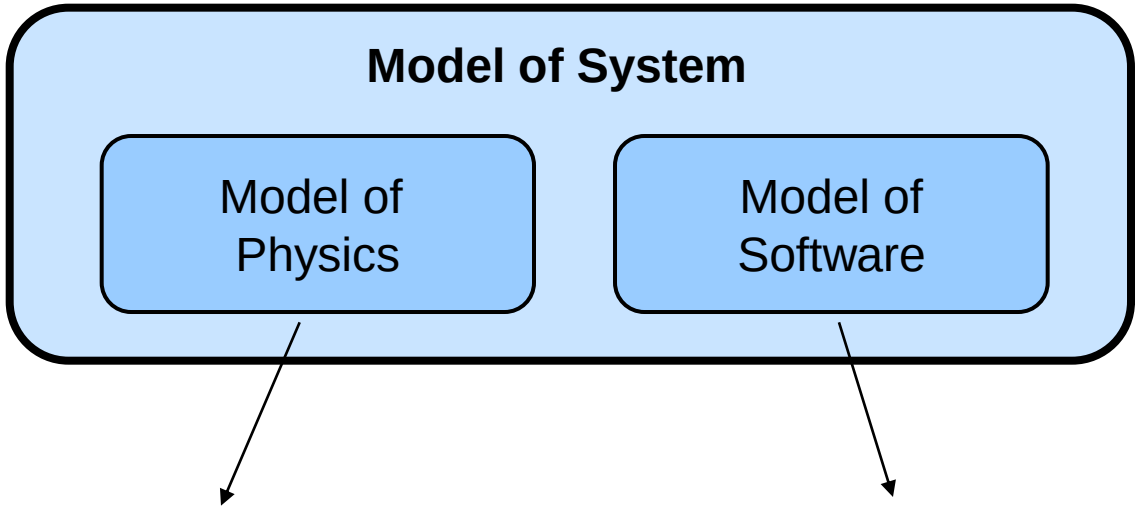
   b) Piecewise Affine Hybrid Systems

III. **Application to Complex Dynamics via Hybridization**

Formal Verification



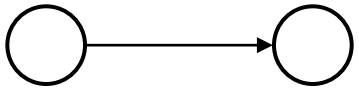
Formal Verification



continuous dynamics

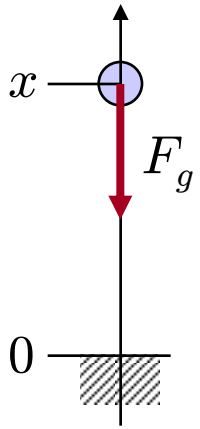
$$\dot{x} = f(x)$$

discrete dynamics



Modeling Hybrid Systems

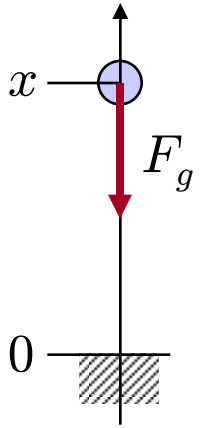
- **Example: Bouncing Ball**
  - ball with mass m and position x in free fall
  - bounces when it hits the ground at $x = 0$
  - initially at position x_0 and at rest



Part I – Free Fall

- **Condition for Free Fall**
  - ball above ground:

$$x \geq 0$$



- **First Principles (physical laws)**
  - gravitational force :

$$F_g = -mg$$

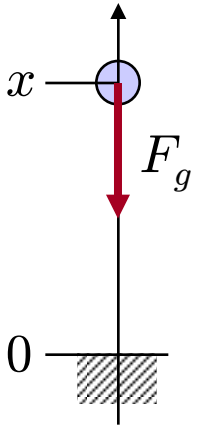
$$g = 9.81\mathrm{m/s}^2$$

  - Newton's law of motion :

$$m\ddot{x} = F_g$$

Part I – Free Fall

$$\begin{aligned} F_g &= -mg \\ m\ddot{x} &= F_g \end{aligned}$$



- **Obtaining 1st Order ODE System**
  - ordinary differential equation $\dot{x} = f(x)$
  - transform to 1st order by introducing variables for higher derivatives
  - here: $v = \dot{x}$:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

Part II – Bouncing

- **Conditions for “Bouncing”**
  - ball at ground position: $x = 0$
  - downward motion: $v < 0$
- **Action for “Bouncing”**
  - velocity changes direction
  - loss of velocity (deformation, friction)
  - $v := -cv$, $0 \leq c \leq 1$

Combining Part I and II

- **Free Fall**
  - while $x \geq 0$,

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

continuous dynamics

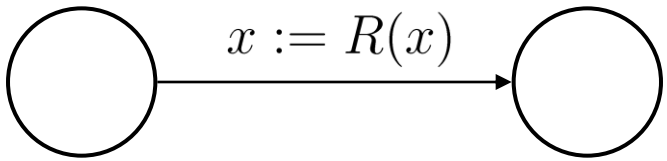
$$\dot{x} = f(x)$$

- **Bouncing**
  - if $x = 0$ and $v < 0$

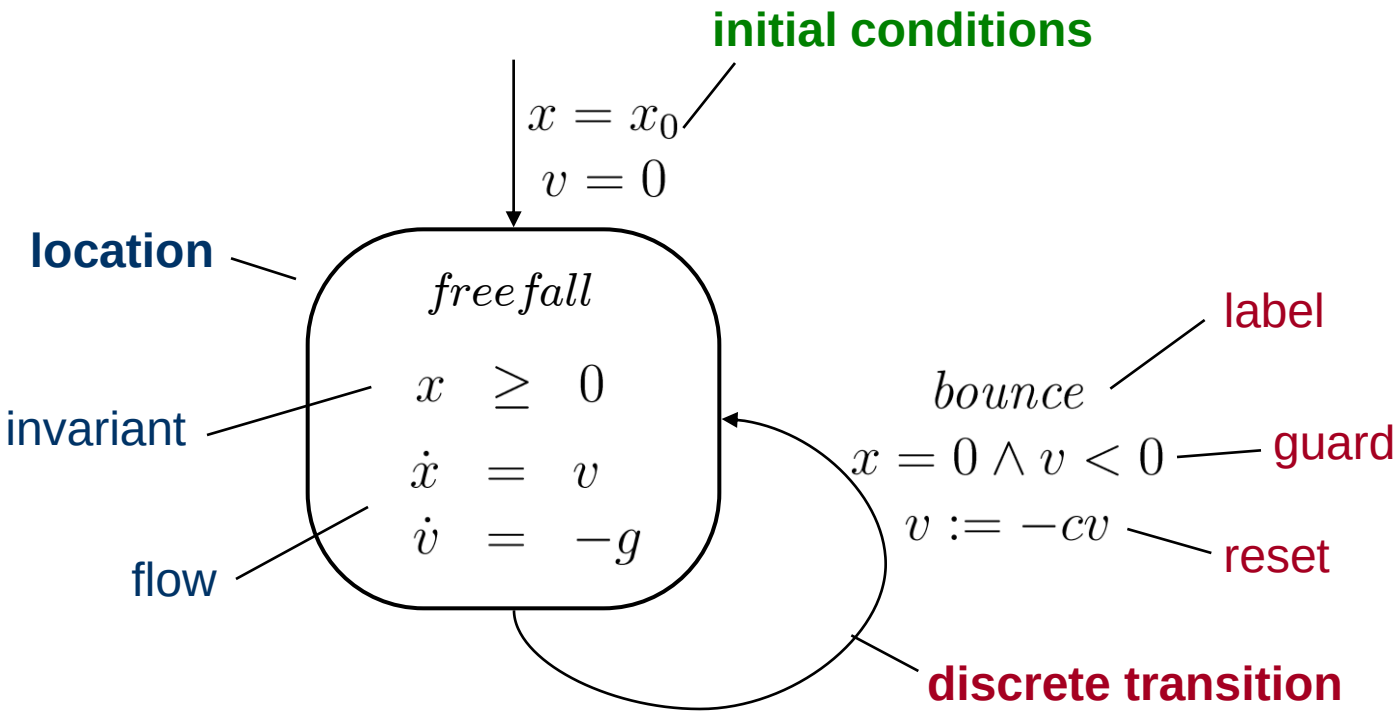
$$v := -cv$$

discrete dynamics

$$x \in G$$
$$x := R(x)$$



Hybrid Automaton Model



Hybrid Automata

$$H = (Loc, Var, Ini, Inv, Trans, Lab, Flow)$$

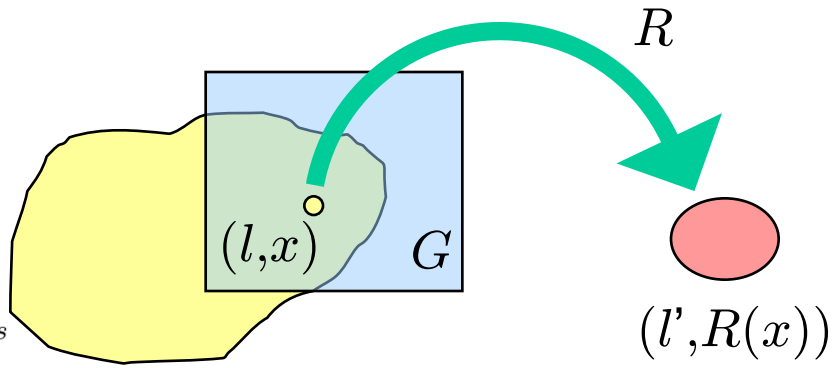
- **Defining Inhabited State Space:**
  - Locations Loc $\{freefall\}$
  - Variables Var $\{x, v\}$
    - Valuation: $x \in \mathbb{R}^{Vars}$ attributes a real value to each variable
    - State: $s = (l, x)$, with $l \in Loc$, $x \in \mathbb{R}^{Vars}$
  - Initial states $Ini \subseteq Loc \times \mathbb{R}^{Vars}$ $\{(freefall, (x = x_0, v = 0))\}$
  - Invariant $Inv \subseteq Loc \times \mathbb{R}^{Vars}$ $\{(freefall, (x \geq 0, v \in \mathbb{R}))\}$

Hybrid Automata – Discrete Dynamics

- **Defining Discrete Dynamics:** $Trans$

  $(l, \alpha, G, R, l') \in Trans$, with

  - label $\alpha \in Lab$,
  - guard $G \subseteq \mathbb{R}^{Vars}$,
  - reset $R : \mathbb{R}^{Vars} \rightarrow 2^{\mathbb{R}^{Vars}}$



- **Semantics: Discrete Transition**
  - can jump from (l,x) to (l', x') if $x \in G$ and $x' \in R(x)$

Hybrid Automata – Cont. Dynamics

- **Defining Continuous Dynamics:** $Flow$

$$Flow : Loc \times \mathbb{R}^{\text{Vars}} \rightarrow 2^{\mathbb{R}^{\text{Vars}}}$$

  - for each location l differential inclusion

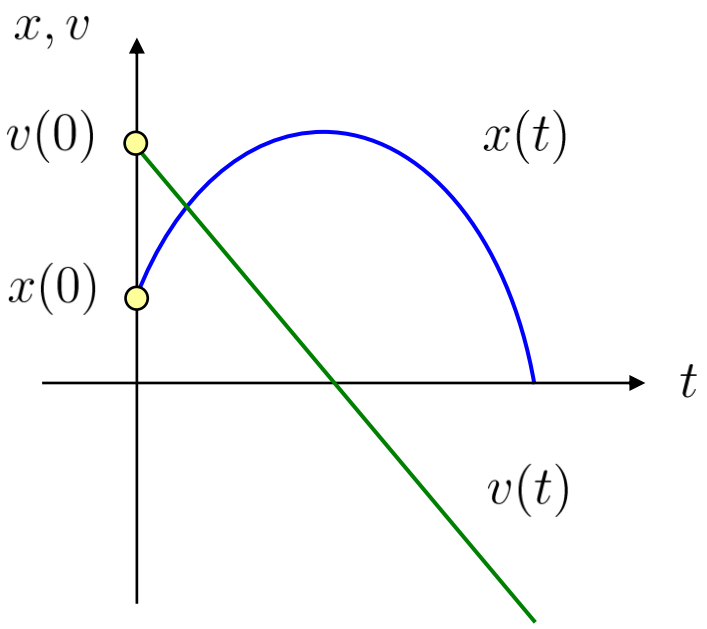
$$\dot{x} \in Flow(l, x)$$

- **Semantics: Time Elapse**
  - change state along $x(t)$ as time elapses
  - $x(t)$ must be in invariant Inv
  - $\dot{x}(t) \in Flow(l, x)$

Hybrid Automata – Cont. Dynamics

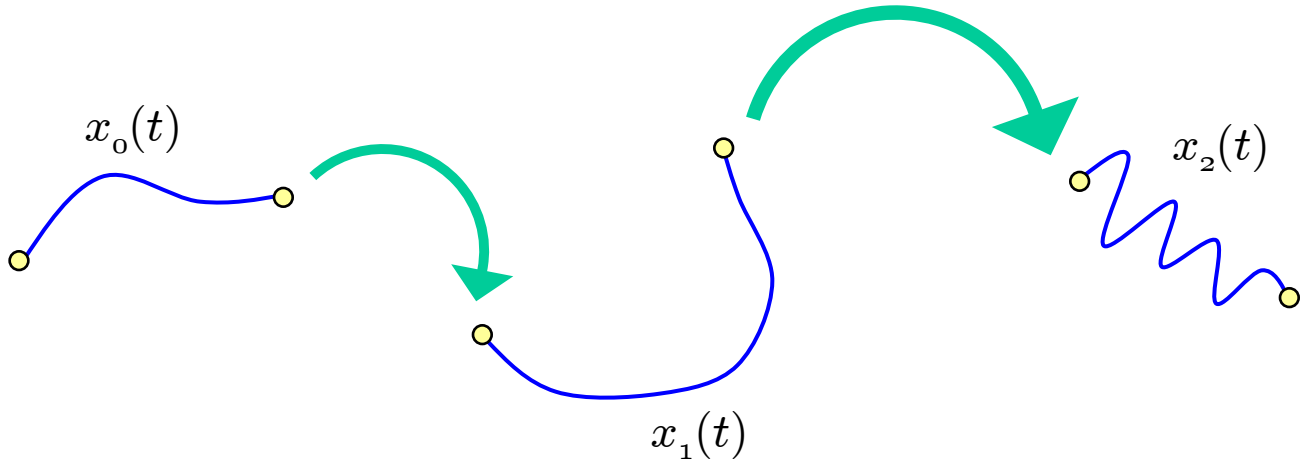
- **Bouncing Ball:**
  - Flow:

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

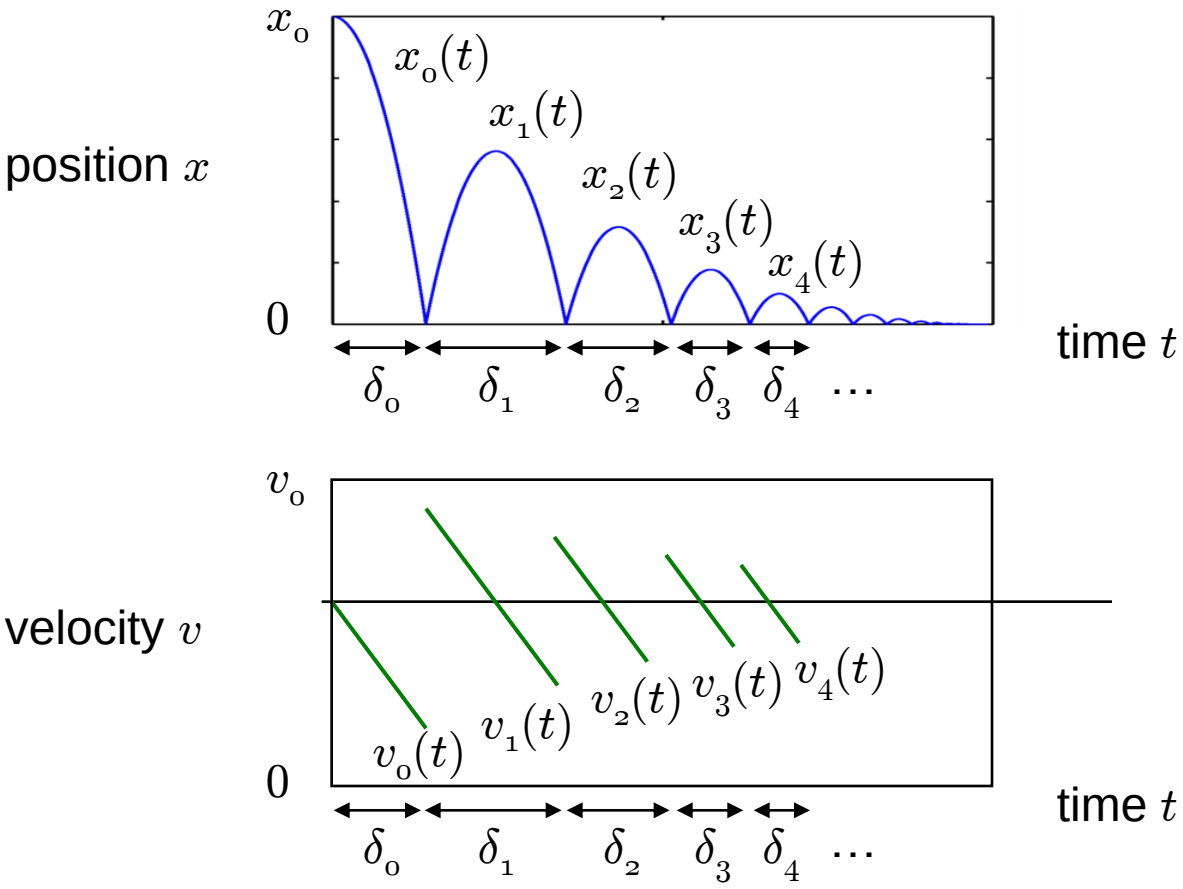


Hybrid Automata - Semantics

- **Run**
  - sequence of discrete transitions and time elapse
- **Execution**
  - run that starts in the initial states

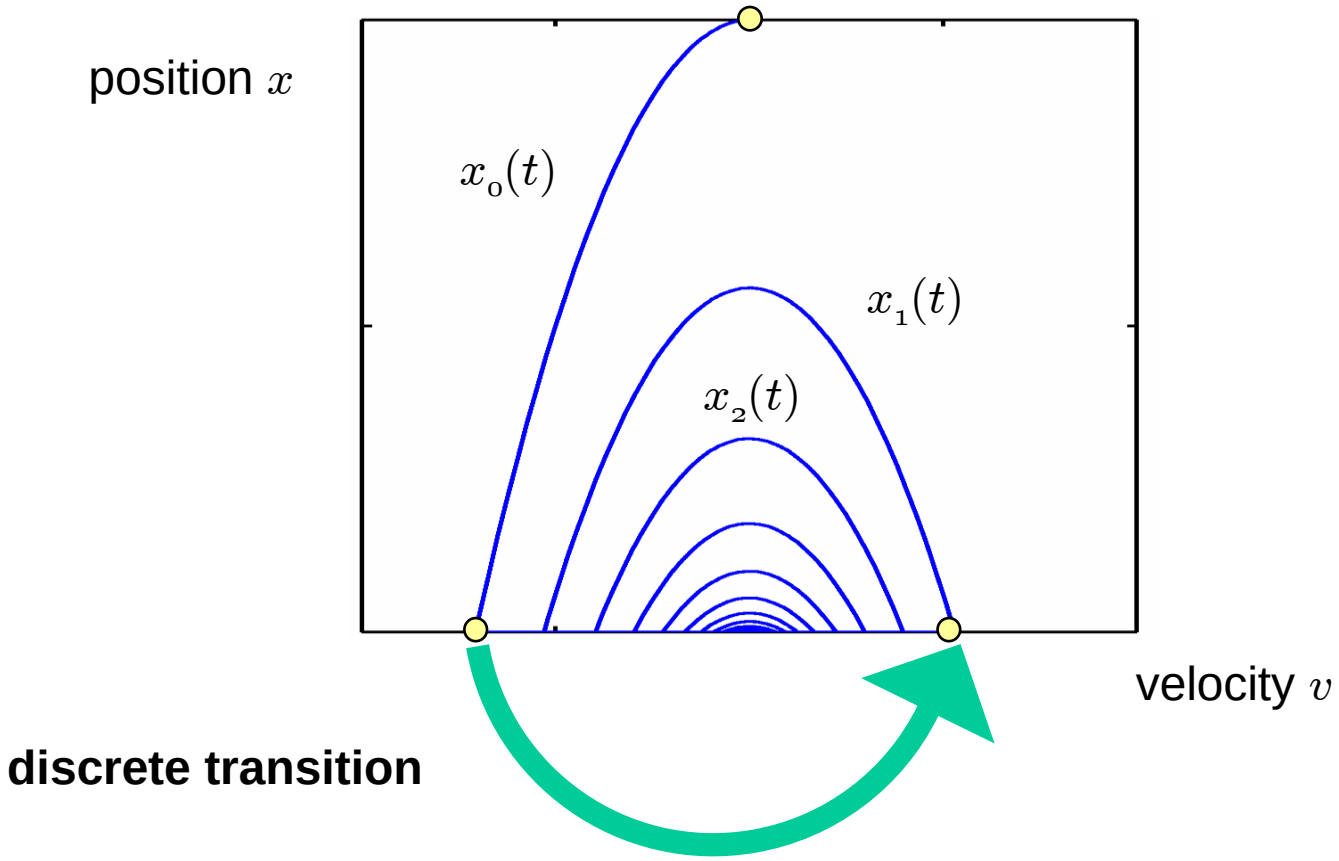


Execution of Bouncing Ball

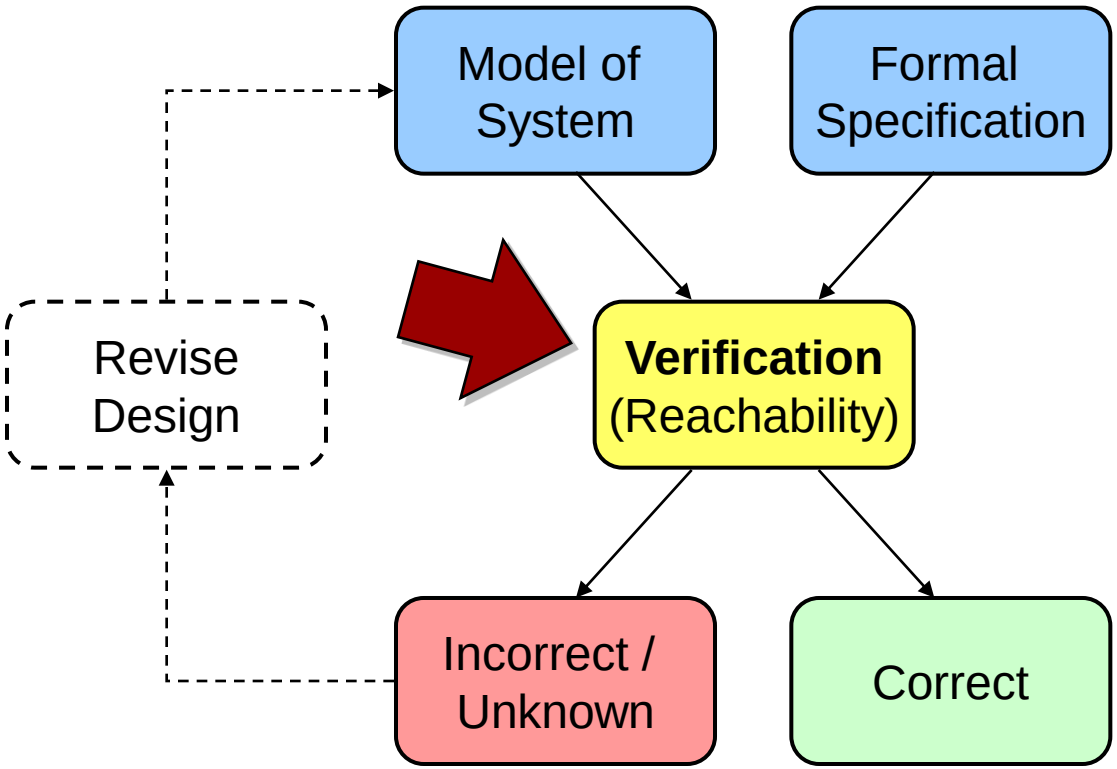


Execution of Bouncing Ball

- **State-Space View (infinite time range)**

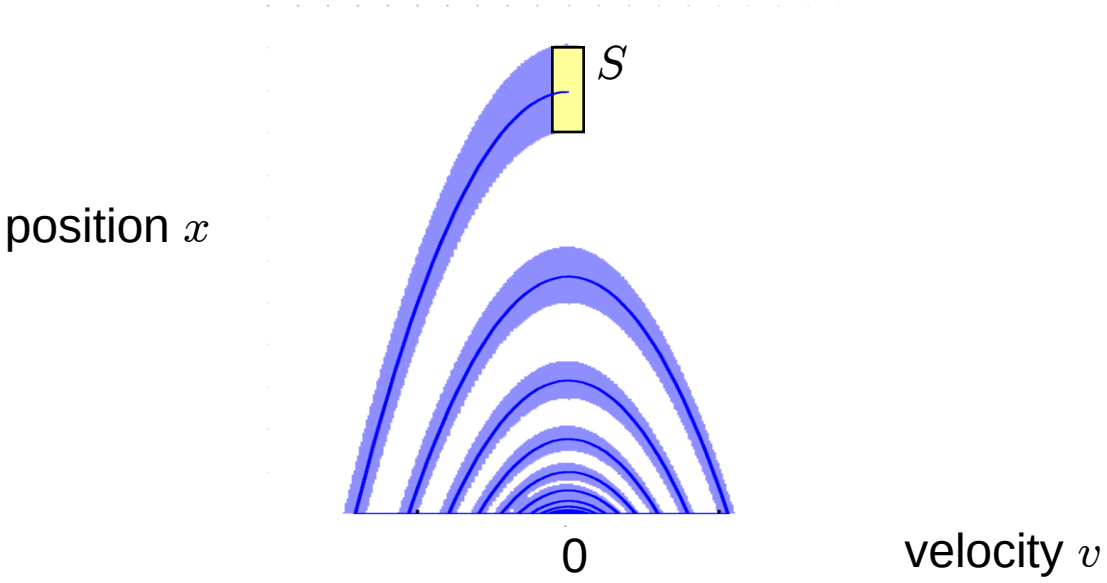


Formal Verification



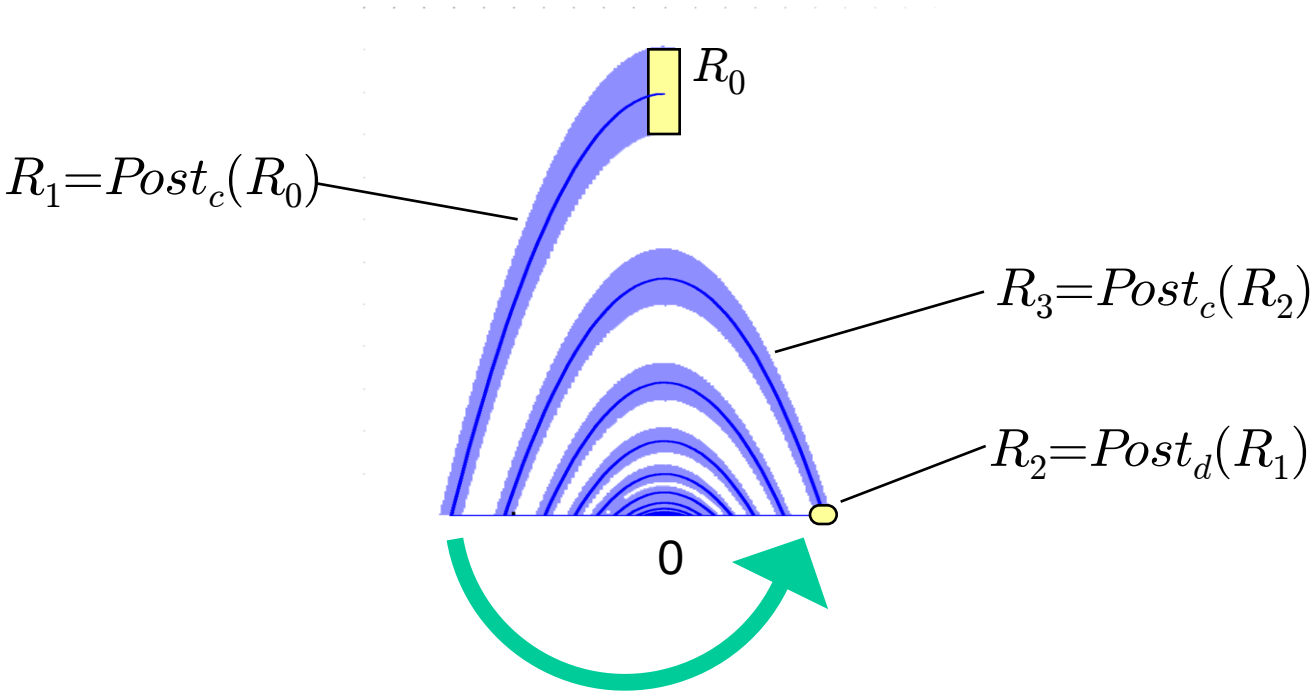
Computing Reachable States

- **Reachable states:** $Reach(S)$
  - any state encountered in a run starting in S



Computing Reachable States

- **Compute successor states**
  - discrete transitions : $Post_d(R)$
  - time elapse : $Post_c(R)$



Computing Reachable States

- **Fixpoint computation**
  - Initialization: $R_0 = Ini$
  - Recurrence: $R_{k+1} = R_k \cup Post_d(R_k) \cup Post_c(R_k)$
  - Termination: $R_{k+1} = R_k \Rightarrow Reach = R_k$.
- **Problems**
  - in general termination not guaranteed
  - time-elapse very hard to compute with sets

Chapter Summary

- **Why should we care?**
  - Reachability Analysis is a set-based computation that can answer many interesting questions about a system (safety, bounded liveness,…)
- **What's the problem?**
  - The hardest part is computing time elapse.
  - Explicit solutions only for very simple dynamics.
- **What's the solution?**
  - First study simple dynamics.
  - Then apply these techniques to complex dynamics.

Outline

I. Hybrid Automata and Reachability

II. Reachability for Simple Dynamics

a) Linear Hybrid Automata

b) Piecewise Affine Hybrid Systems

III. Application to Complex Dynamics via Hybridization

In this Chapter…

- **A very simple class of hybrid systems**
- **Exact computation of discrete transitions and time elapse**
  - Note: Reachability (and pretty much everything else) is nonetheless **undecidable**.
- **A case study**

Linear Hybrid Automata

- **Continuous Dynamics**
  - piecewise constant: $\dot{x} = 1$
  - intervals: $\dot{x} \in [1, 2]$
  - conservation laws: $\dot{x}_1 + \dot{x}_2 = 0$
  - general form: conjunctions of linear constraints

$$a \cdot \dot{x} \bowtie b, \qquad a \in \mathbb{Z}^n, b \in \mathbb{Z}, \bowtie \in \{<, \leq\}.$$

= convex polyhedron over derivatives

Linear Hybrid Automata

- **Discrete Dynamics**
  - affine transform: $x := ax + b$
  - with intervals: $x_2 := x_1 \pm 0.5$
  - general form: conjunctions of linear constraints (new value x')

$$a \cdot x + a' \cdot x' \bowtie b, \qquad a, a' \in \mathbb{Z}^n, b \in \mathbb{Z}, \bowtie \in \{<, \leq\}$$

= convex polyhedron over x and x'

Linear Hybrid Automata

- **Invariants, Initial States**
  - general form: conjunctions of linear constraints

$$a \cdot x \bowtie b, \qquad a \in \mathbb{Z}^n, b \in \mathbb{Z}, \bowtie \in \{<, \leq\},$$

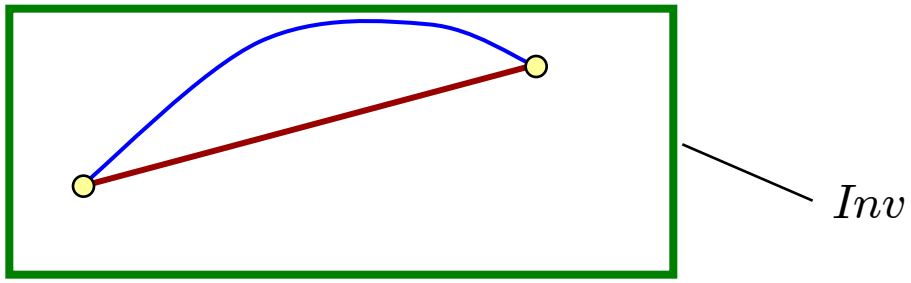
= convex polyhedron over x

Reachability with LHA

- **Compute discrete successor states** $Post_d(S)$
  - all x' for which exists $x \in S$ s.t.
    - $x \in G$
    - $x' \in R(x) \cap Inv$
- **Operations:**
  - existential quantification
  - intersection
  - standard operations on convex polyhedra

Reachability with LHA

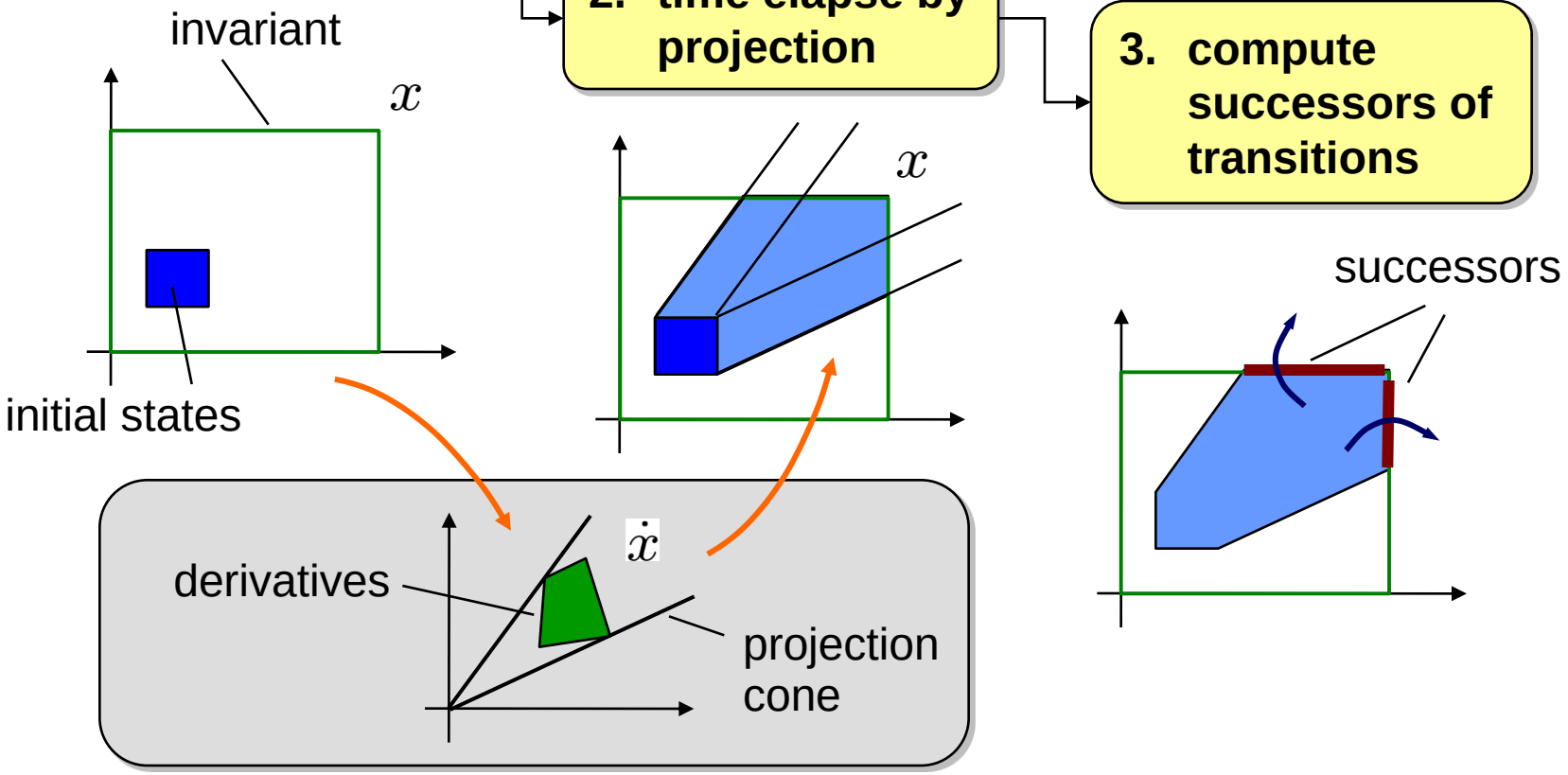
- **Compute time elapse states** $Post_c(S)$
- **Theorem** [Alur et al.]
  - Time elapse along arbitrary trajectory iff time elapse along straight line (convex invariant).



  - time elapse along straight line can be computed as projection along cone [Halbwachs et al.]

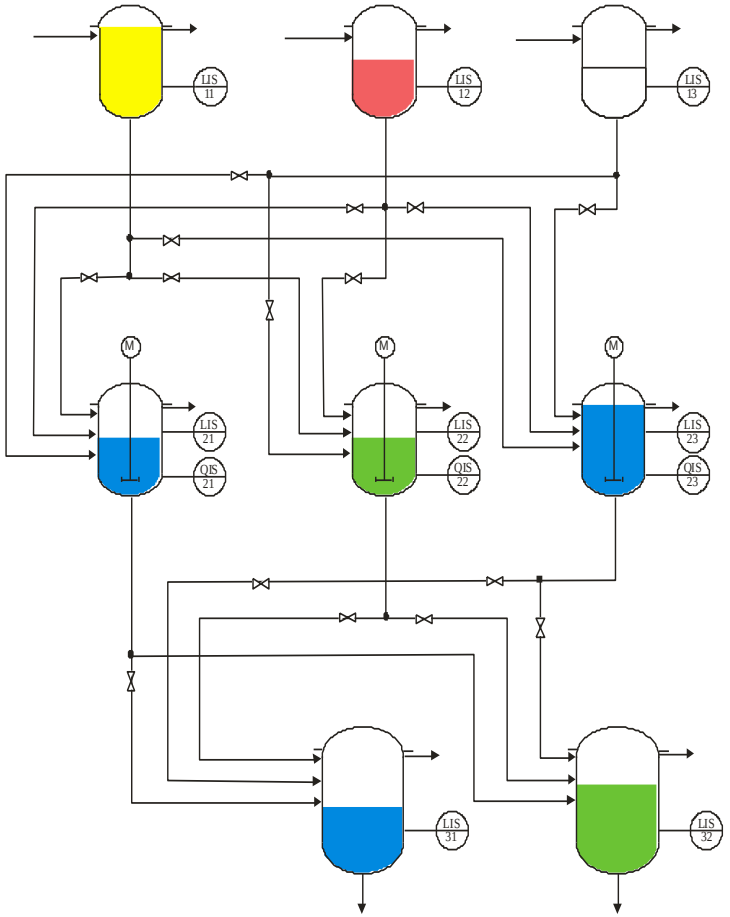
Reachability with LHA [Halbwachs, Henzinger, 93-97]

1. get projection cone
2. time elapse by projection
3. compute successors of transitions



Multi-Product Batch Plant

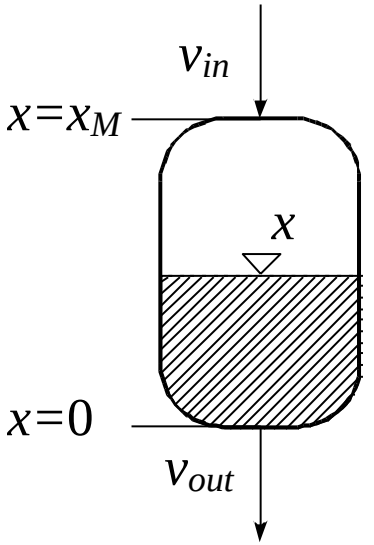
Multi-Product Batch Plant



- **Cascade mixing process**
  - 3 educts via 3 reactors
    $\Rightarrow$ 2 products
- **Verification Goals**
  - Invariants
    - overflow
    - product tanks never empty
  - Filling sequence
- **Design of verified controller**

Switched Buffer Network

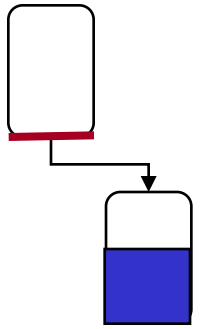
- **Buffers** $s_1,\ldots,s_n$
  - store material → continuous level $x_1,\ldots,x_n$
- **Channels**
  - transport material from buffer to buffer → continuous throughput $v(s,s')$, nondeterministic inside interval
- **Switching**
  - activate/deactivate channels discretely



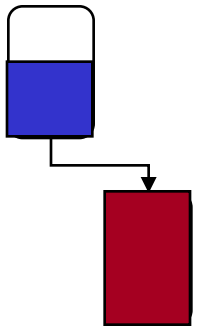
Buffer

Continuous Dynamics

- **Stationary throughput**
  - $v \in [a,b]$



- **Source buffer empty**
  - throughput may seize, $v \in [0,b]$
  - **inflow of source = outflow of source**



- **Target buffer full**
  - throughput may seize, $v \in [0,b]$
  - **inflow of target = outflow of target**

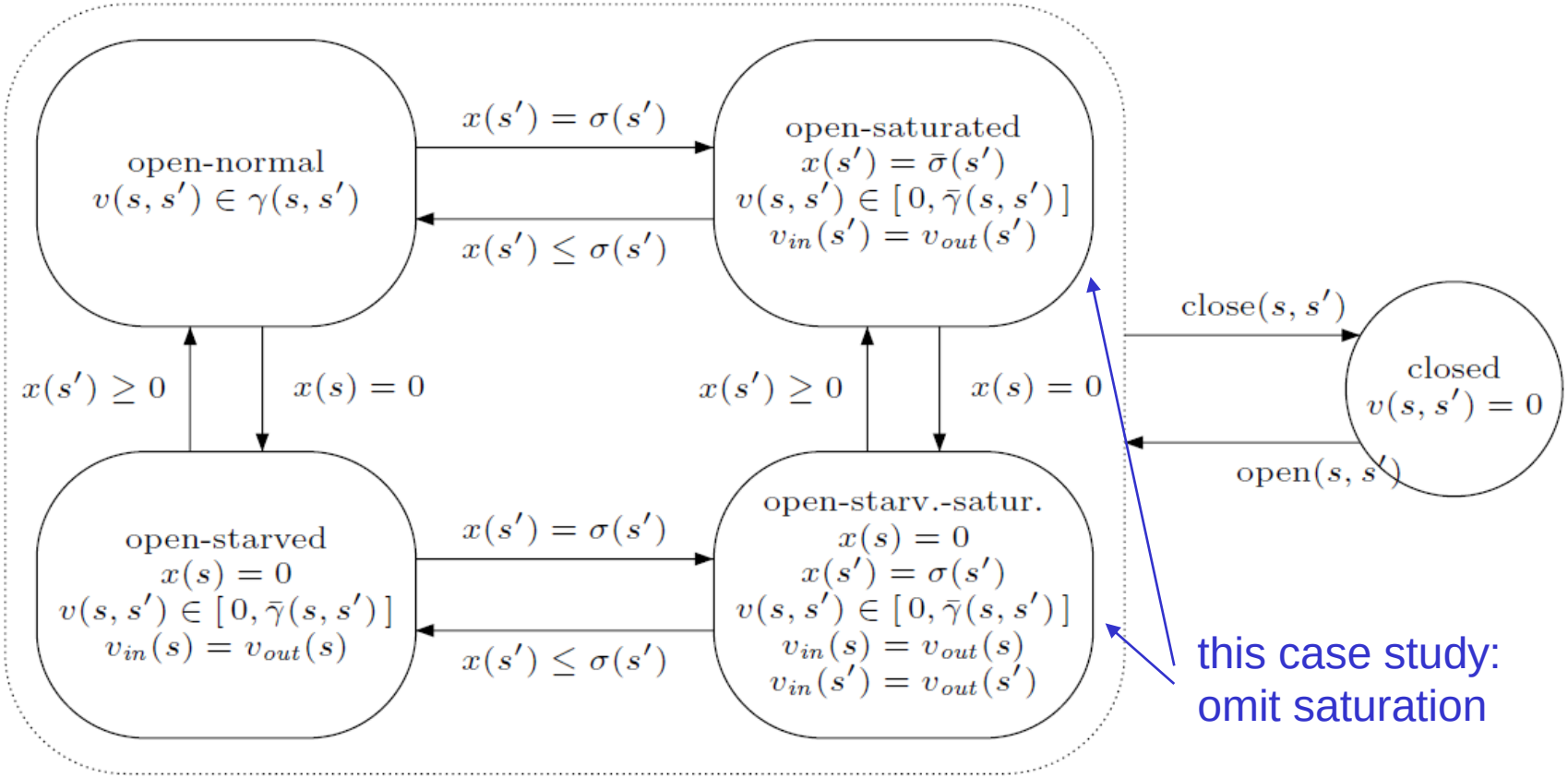
Buffer Automaton Model

- tank levels = cont. variables x_i
- incoming flow $v_{in}(s)=\sum_{s'} v(s',s)$
- outgoing flow $v_{out}(s)=\sum_{s'} v(s,s')$

$$0 \leq x(s) \leq \sigma(s)$$
$$\dot{x}(s) = v_{in}(s) - v_{out}(s)$$

Channel Automaton Model

– throughput = algebraic variable (will be projected away)



Production Schedule

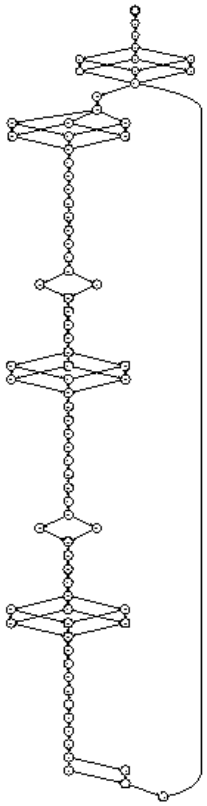
Table 1. Control strategy as sequence of batch transfers (column: from, rows: to)

row	delivery*	B11	B12	B13	R21	R22	R23
1	B11,$B13_2$	○	–	○	–	B32↓	B32↑
2	–	–	R22	$R21_1^*$	○	○	B32↓
3	B12	R23	○	$R22_0$	B31↑	○	○
4	B11,$B13_2$	○	–	○	B31↓	B32↑	–
5	–	R21	–	$R23_1^*$	○	B32↓	○
6	B11	○	R22	$R21_0$	○	○	B31↑
7	B12,$B13_2$	–	○	○	B31↑	–	B31↓
8	–	–	R23	$R22_1^*$	B31↓	○	○
9	B12	R21	○	$R23_0$	○	B32↑	○

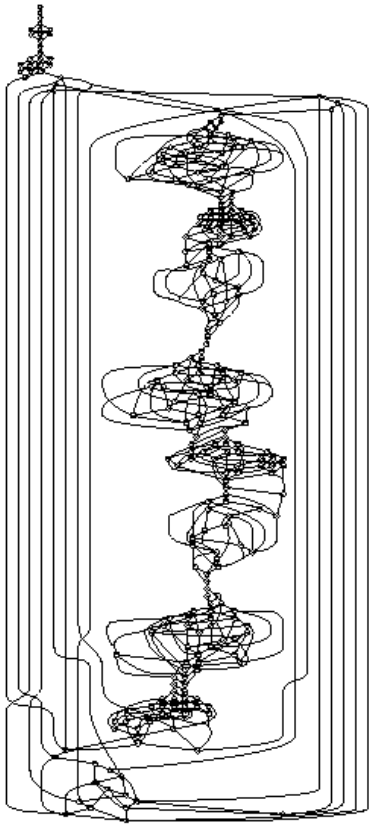
* time critical; $_{2,1,0}$ fill/drain to level $x_{B13} = 1700, 850, 0$

- uses 3 reactors in parallel
- transfers of batches from one tank to another
- formally a control strategy: locations $\times$ cont. variables $\rightarrow$ locations

Verification with PHAVer



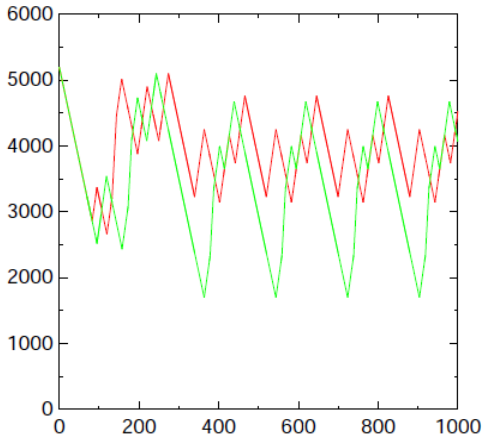
Controller



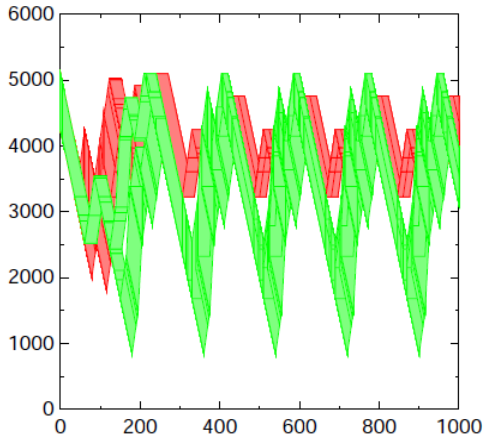
Controlled Plant

- **Controller automaton model**
  - 78 locations
  - ASAP transitions
- **Controller + Plant**
  - 266 locations, 823 transitions (~150 reachable)
- **Reachability over infinite time**
  - 120s—1243s, 260—600MB
  - computation cost increases with nondeterminism (intervals for throughputs, initial states)

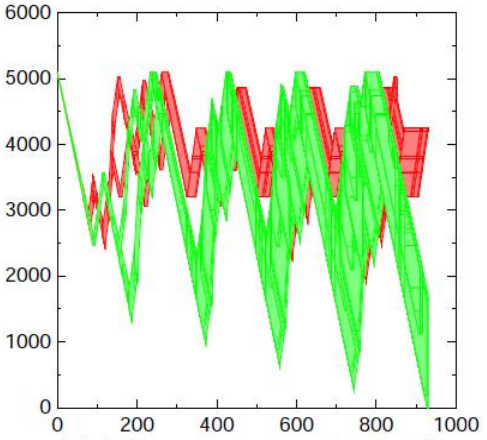
Verification with PHAVer



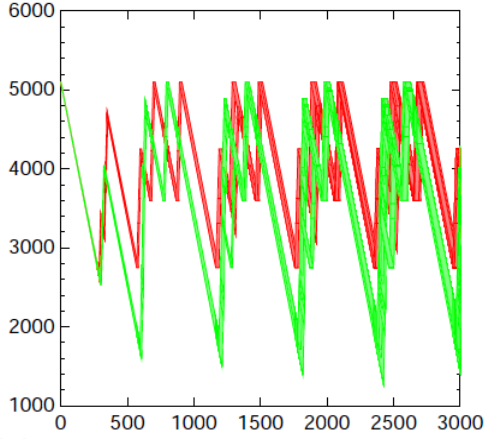
(a) BP8.1: nominal case



(b) BP8.2: varying initial cond.



(c) BP8.3: varying demand



(d) BP8.4: varying but slow demand

					Automaton		Reachable Set	
Instance	Time [s]	Mem. [MB]	Depth[a]	Checks[b]	Loc.	Trans.	Loc.	Poly.
BP8.1	120	267	173	279	266	823	130	279
BP8.2	139	267	173	422	266	823	131	450
BP8.3	845	622	302	2669	266	823	143	2737
BP8.4	1243	622	1071	4727	266	823	147	4772

[*] on Xeon 3.20 GHz, 4GB RAM running Linux; [a] lower bound on depth in breadth-first search; [b] number of applications of post-operator

Outline

I. Hybrid Automata and Reachability

II. Reachability for Simple Dynamics

a) Linear Hybrid Automata

b) Piecewise Affine Hybrid Systems

III. Application to Complex Dynamics via Hybridization

In this Chapter…

- **Another class of (not quite so) simple dynamics**
  - but things are getting serious (no explicit solution for sets)
- **Exact Computation time elapse only at discrete points in time**
  - used to overapproximate continuous time
- **Efficient data structures**

Piecewise Affine Hybrid Systems

- **Affine dynamics**
  - Flow:

$$\dot{x} = Ax + b \text{ (deterministic)}$$

$$\dot{x} \in Ax + B, \text{ with } B \text{ a set (nondeterministic)}$$

  - For time elapse it's enough to look at a single location.

Linear Dynamics

- **Let's begin with "autonomous" part of the dynamics:**

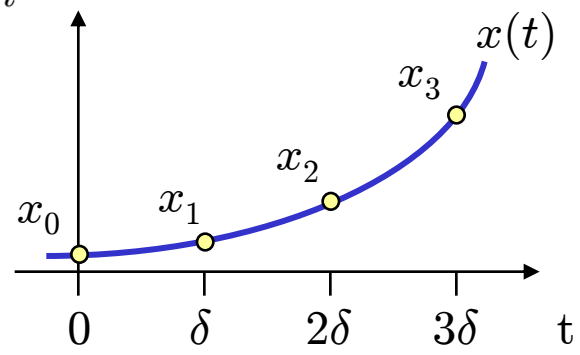
$$\dot{x} = Ax, \quad x \in \mathbb{R}^n$$

- **Known solutions:**
  - analytic solution in continuous time
  - explicit solution at discrete points in time (up to arbitrary accuracy)
- **Approach for Reachability:**
  - Compute reachable states over finite time: $Reach_{[0,\mathrm{T}]}(X_{Ini})$
  - Use time-discretization, but with care!

Time-Discretization for an Initial Point

- **Analytic solution:** $x(t) = e^{At} x_{Ini}$
  - with $t = \delta k$:

$$x(\delta(k+1)) = e^{A\delta} x(\delta k)$$



- **Explicit solution in discretized time (recursive):**

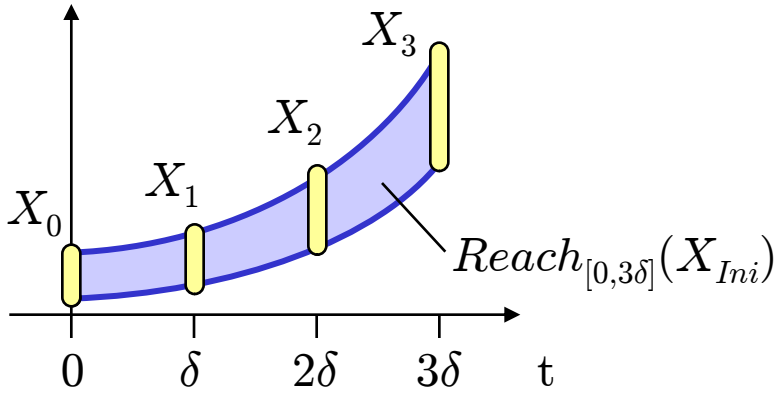
$$x_0 = x_{Ini}$$
$$x_{k+1} = e^{A\delta} x_k$$

multiplication with const. matrix $e^{A\delta}$
= linear transform

Time-Discretization for an Initial Set

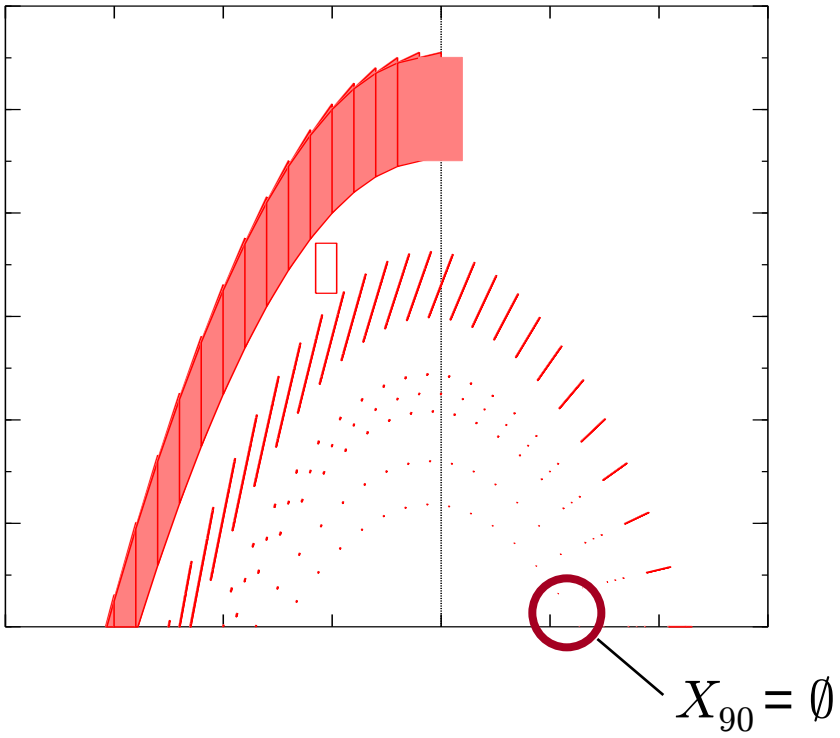
- **Explicit solution in discretized time**

$$\begin{aligned} X_0 &= X_{Ini} \\ X_{k+1} &= e^{A\delta} X_k \end{aligned}$$



- **Acceptable solution for purely continuous systems**
  - $x(t)$ is in $\epsilon(\delta)$-neighborhood of some X_k
- **Unacceptable for hybrid systems**
  - discrete transitions might “fire” between sampling times
  - if transitions are “missed,” $x(t)$ not in $\epsilon(\delta)$-neighborhood

Bouncing Ball



– In other examples this error might not be as obvious…

Reachability by Time-Discretization

- **Goal:**
  - Compute sequence Ω_k over bounded time $[0, N\delta]$ such that:

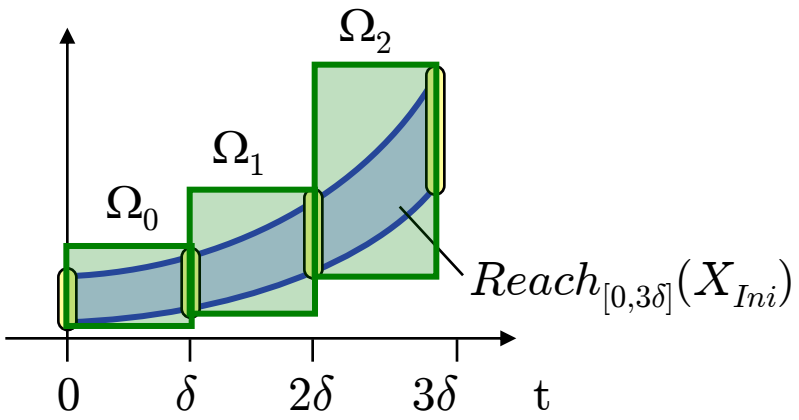
$$\mathrm{Reach}_{[0,N\delta]}(X_{Ini}) \subseteq \Omega_0 \cup \Omega_1 \cup \ldots \cup \Omega_N$$

- **Approach:**
  - Refine Ω_k by recurrence:

$$\Omega_{k+1} \quad = \quad e^{A\delta}\Omega_k$$

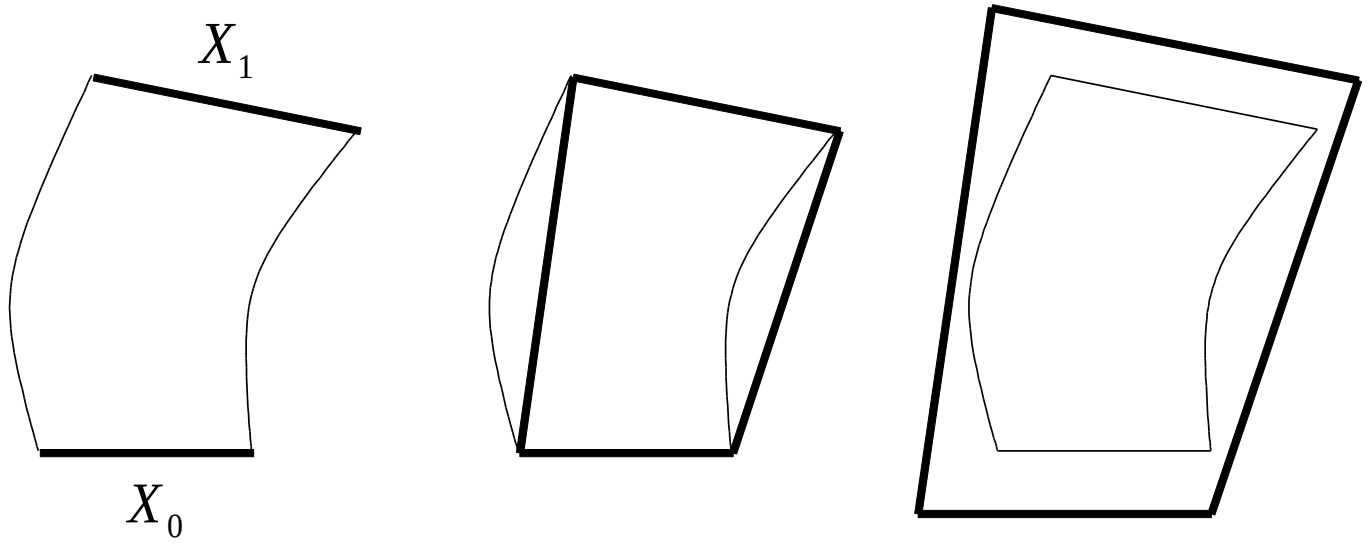
  - Condition for Ω_0:

$$\mathrm{Reach}_{[0,\delta]}(X_{Ini}) \subseteq \Omega_0$$



Time-Discretization with Convex Hull

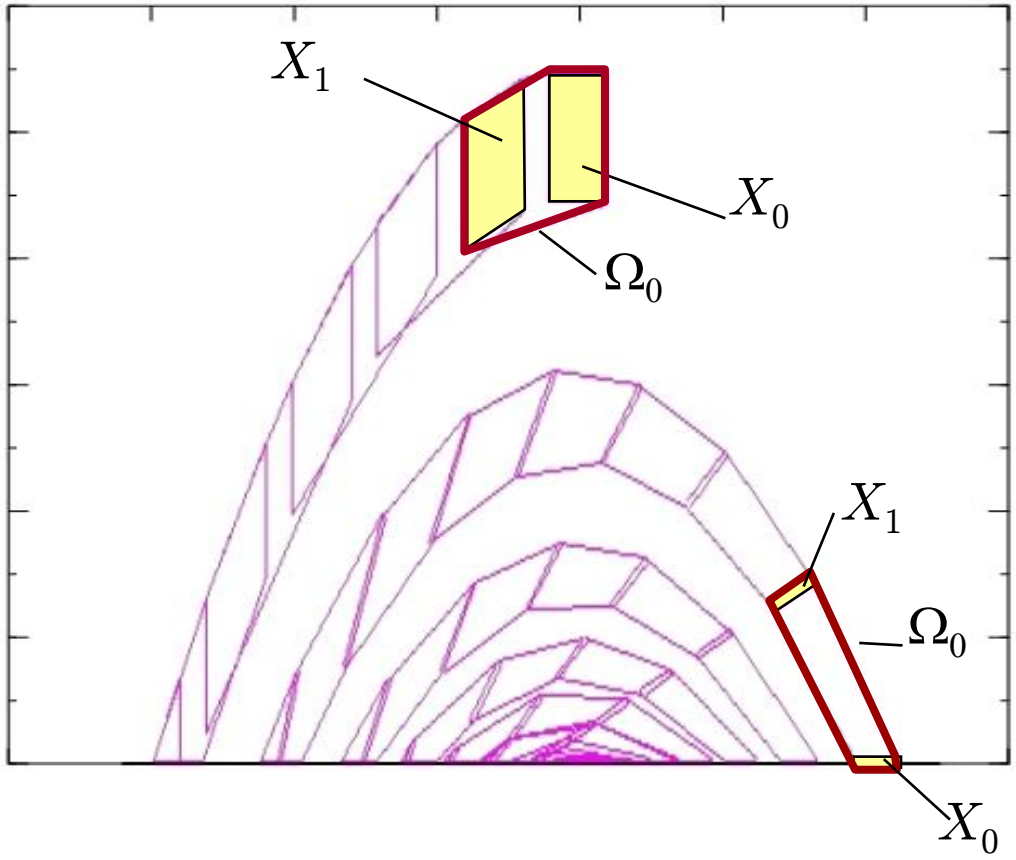
- **Overapproximating** $Reach_{[0,\delta]}$**:**



$Reach_{[0,\delta]}(X_0)$ $Conv(X_0, X_1)$ $Bloat(Conv(X_0, X_1))$

Time-Discretization with Convex Hull

- **Bouncing Ball:**



Nondeterministic Affine Dynamics

- **Let's include the effect of inputs:**

$$\dot{x} = Ax + Bu, \quad x \in \mathbb{R}^n, u \in U \subseteq \mathbb{R}^p$$

  - variables $x_1,\ldots,x_n$, inputs $u_1,\ldots,u_p$

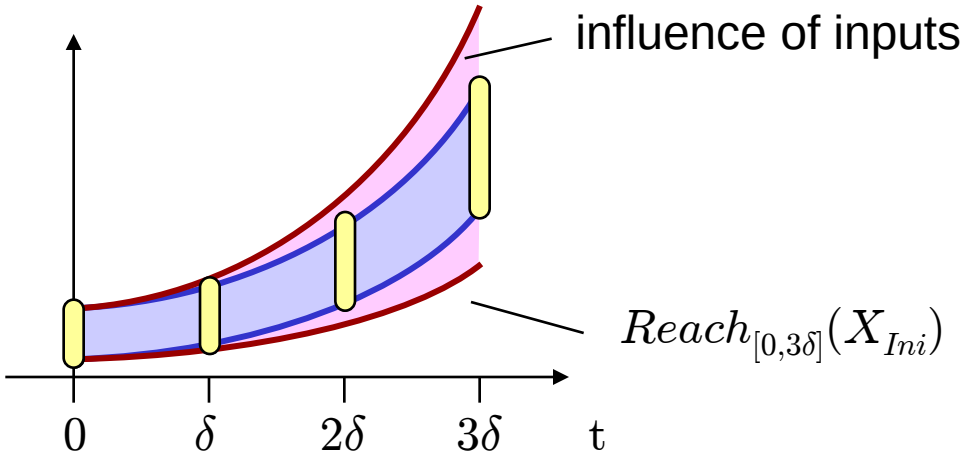
- **Input u models nondeterminism**

$$\dot{x} \in Ax + BU$$

Nondeterministic Affine Dynamics

- **Analytic Solution**

$$x(t) = \underbrace{e^{A\delta}x(0)}_{\text{autonomous dynamics}} + \underbrace{\int_0^{\tau} e^{A(\delta-\tau)}Bu(\tau)d\tau}_{\text{influence of inputs}}$$

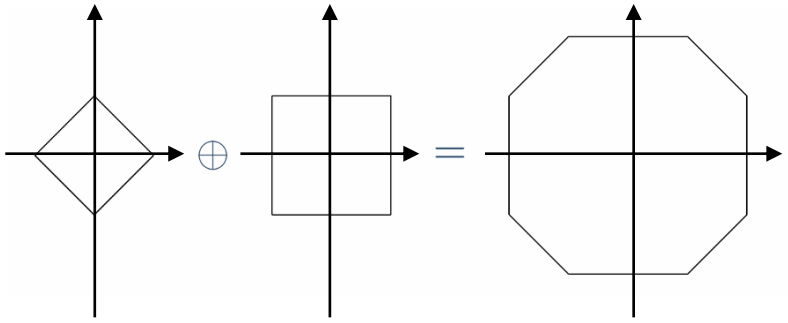


Nondeterministic Affine Dynamics

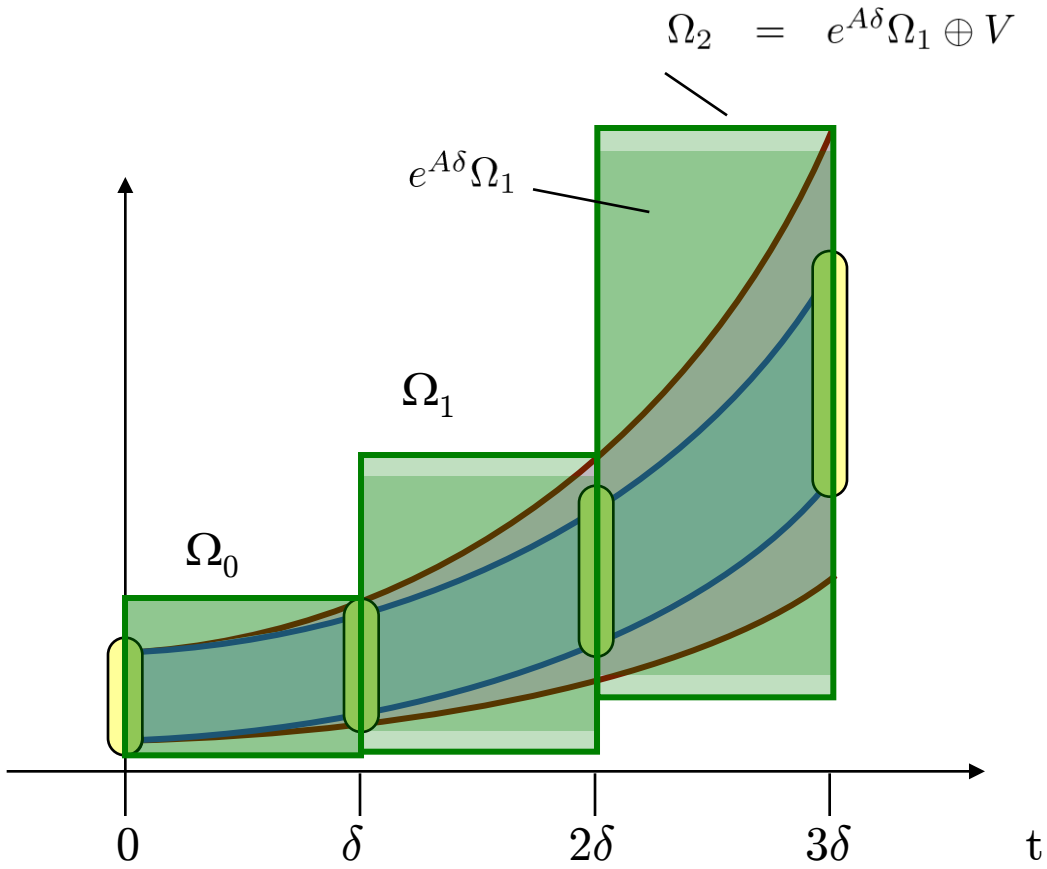
- **How far can the input "push" the system in δ time?**
  - $V =$ box with radius $\frac{e^{||A||\delta}-1}{||A||}\sup_{u\in U}||Bu||$

$$\begin{aligned}\Omega_0 &= Bloat(Conv(X_{Ini}, e^{A\delta}X_{Ini})) \oplus V \\ \Omega_{k+1} &= e^{A\delta}\Omega_k \oplus V\end{aligned}$$

- **Minkowski Sum:** $A \oplus B = \{a + b \mid a \in A,\ b \in B\}$



Nondeterministic Affine Dynamics



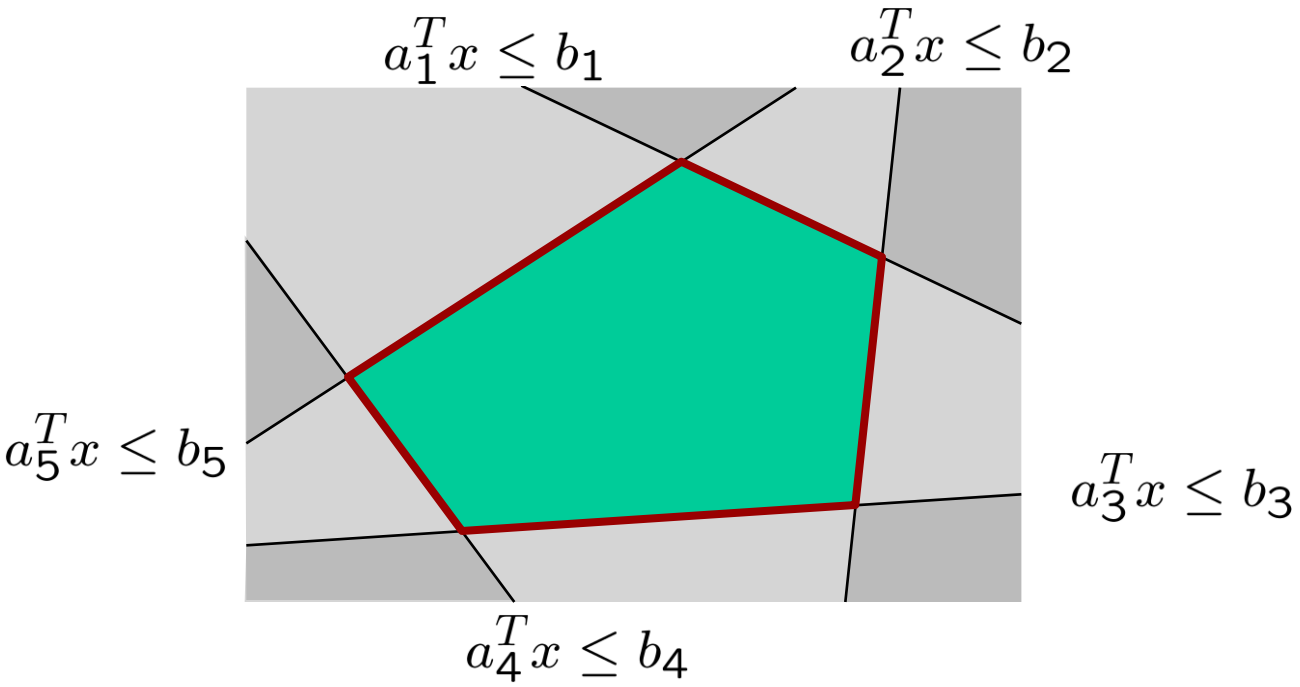
Implementing Reachability

- **Find representation for continuous sets with**
  - linear transformation ($\Omega_{\kappa+1} = \Phi\ \Omega_{\kappa}$)
  - Minkowski Sum
  - intersection (with guards)

Polyhedra

- **Finite conjunction of linear constraints**

$$P = \{x \mid Ax \leq b\}\,.$$



Operations on Polyhedra

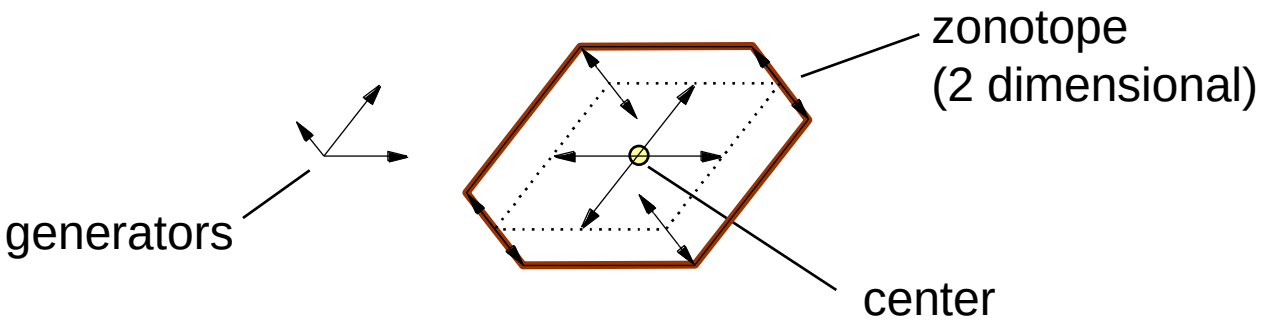
- **Linear Transformation**
  - transform matrix
  - $O(n^3)$
- **Minkowski Sum**
  - need to compute vertices
  - O(exp(n))
- **Intersection**
  - join lists of constraints
  - O(1)

Zonotopes

- **Central symmetric polyhedron**

$$Z = (c, \langle v_1, \ldots, v_m \rangle) = \left\{ c + \sum_{i=1}^{m} \alpha_i v_i \mid \alpha_i \in [-1, 1] \right\}.$$

center generators



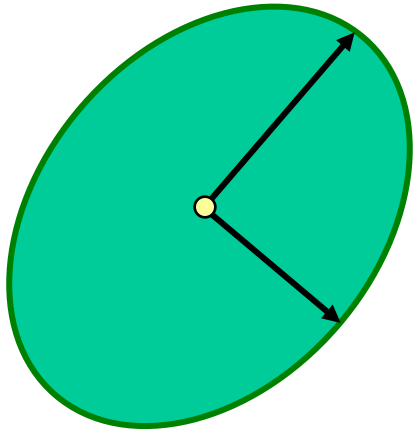
Operations on Zonotopes

- **Linear Transformation**
  - transform generators $\Phi Z = (\Phi c, \langle \Phi v_1, \ldots, \Phi v_m \rangle)$
  - O(mn^2)
- **Minkowski Sum**
  - join lists of generators $Z \oplus Z' = (c + c', \langle v_1, \ldots, v_m, v'_1, \ldots, v'_{m'} \rangle)$
  - O(n)
- **Intersection**
  - Problem: intersection of zonotopes is not a zonotope
  - overapproximate

Ellipsoids

- **Quadratic form**
  - matrix or generator representation

$$E = \left\{ x \mid x^T Q x + A x \leq b \right\}.$$



Operations on Ellipsoids

- **Linear Transformation**
  - transform generators
  - $O(n^2)$
- **Minkowski Sum**
  - Problem: result is not an ellipsoid
  - overapproximate
- **Intersection**
  - Problem: intersection of ellipsoids is not an ellipsoid
  - overapproximate

Implementing Reachability

- **Complexity of 1 Step of Time Elapse:**
  - Polyhedra: O(exp(n))
  - Zonotopes: $O(mn^2)$ ✓

- **Problem: With each iteration, Ω_i get more complex**

$$\Omega_{k+1} = e^{A\delta}\Omega_k \oplus V$$

  - Minkowski sum increases number of
    - Polyhedra: constraints
    - Zonotopes: generators

Wrapping Effect

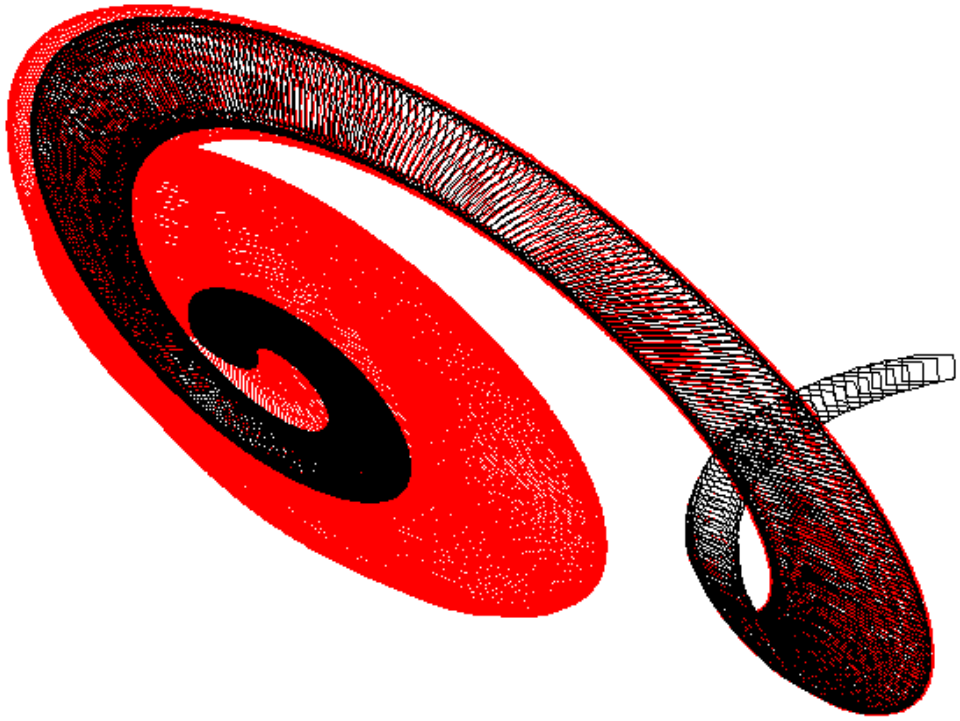
- **Fight complexity by overapproximation**
- **Overapproximated Sequence**

$$\hat{\Omega}_{k+1} \quad = \quad Approx(e^{A\delta}\hat{\Omega}_k \oplus V)$$

  - accumulation of approximations $\rightarrow$ Wrapping Effect
  - exponential increase in approximation error!

Wrapping Effect

- **Exact vs. overapproximation**
  - dimension 5 for 600 time steps
  - overapproximation with 100 generators



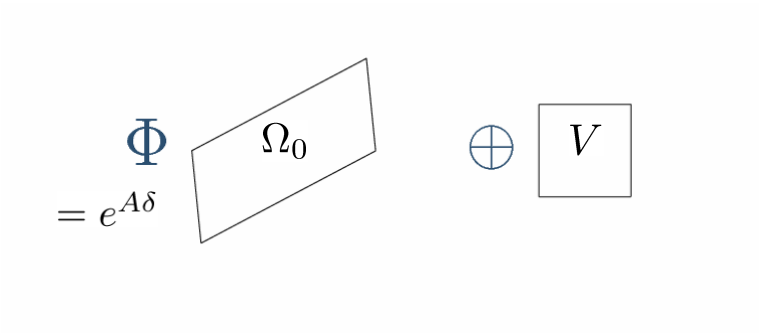
Wrapping Effect

$$\hat{\Omega}_{k+1} \quad = \quad Approx(e^{A\delta}\hat{\Omega}_k \oplus V)$$

- **How does error accumulate?**
  - linear transformation (scaling error up $\rightarrow$ exp)
  - V is added (adding some more error)

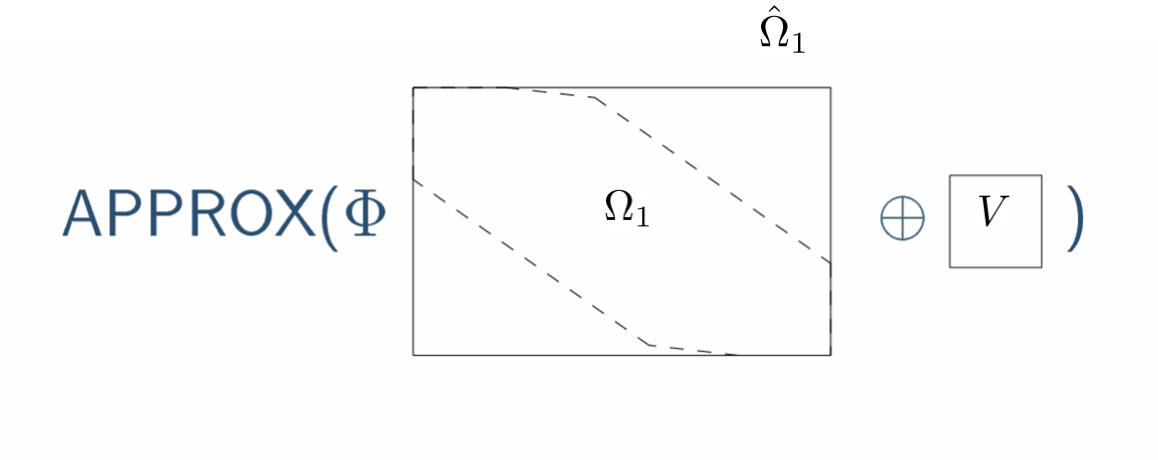
Wrapping Effect

$$\hat{\Omega}_{k+1} \quad = \quad Approx(e^{A\delta}\hat{\Omega}_k \oplus V)$$



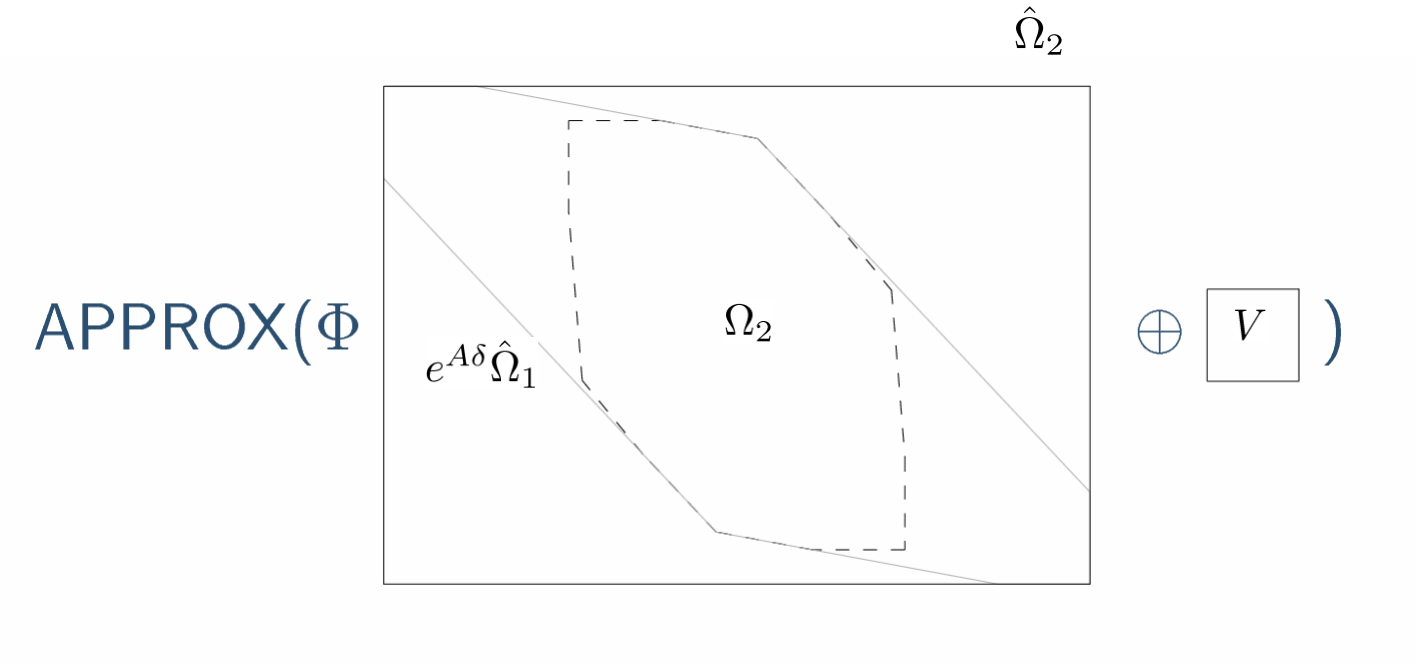
Wrapping Effect

$$\hat{\Omega}_{k+1} \quad = \quad Approx(e^{A\delta}\hat{\Omega}_k \oplus V)$$



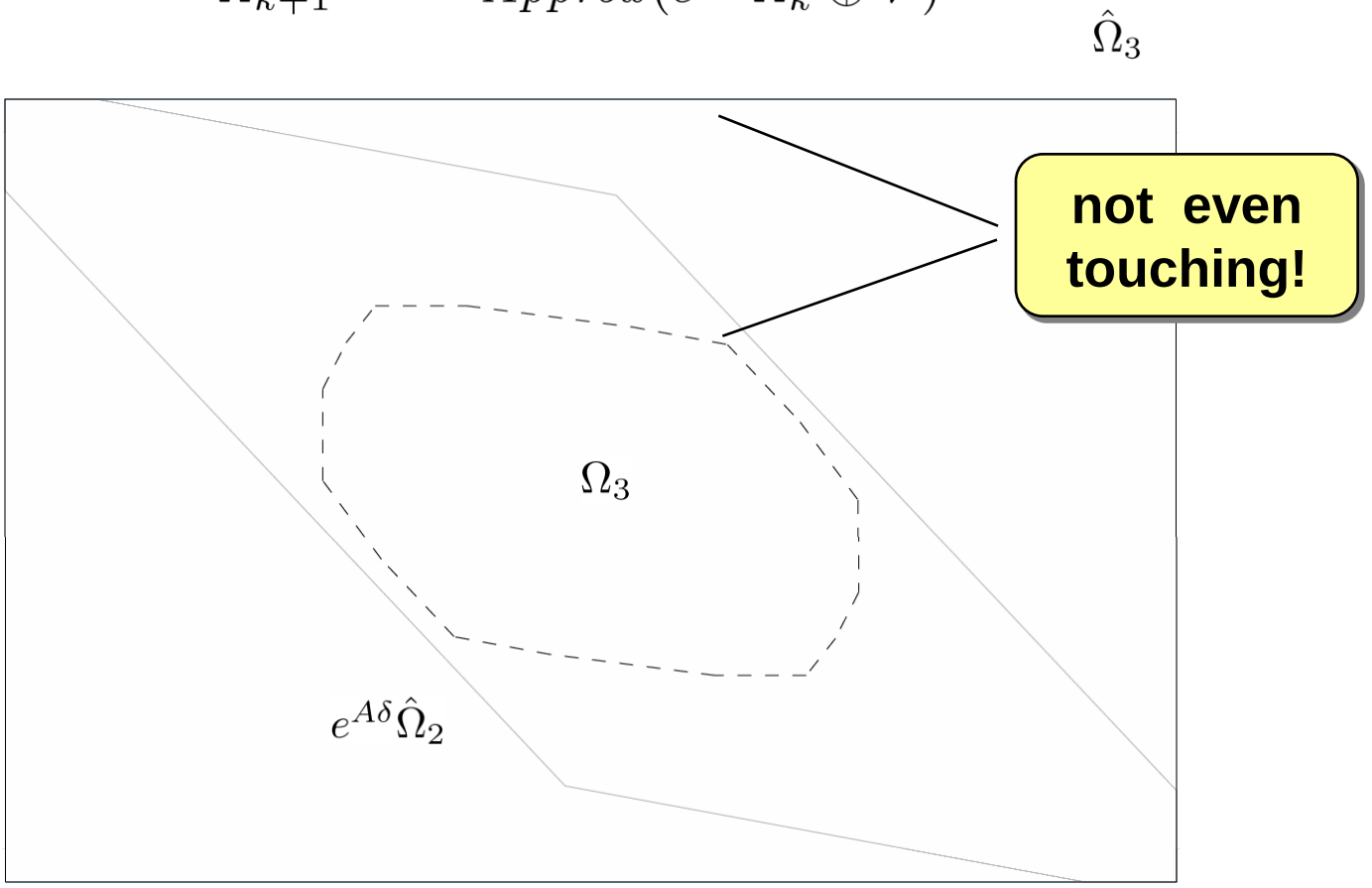
Wrapping Effect

$$\hat{\Omega}_{k+1} \quad = \quad Approx(e^{A\delta}\hat{\Omega}_k \oplus V)$$



Wrapping Effect

$$\hat{\Omega}_{k+1} \quad = \quad Approx(e^{A\delta}\hat{\Omega}_k \oplus V)$$



Fighting the Wrapping Effect

- **Separate transformations and Minkowski sums:**

$$\Omega_{k+1} = \underbrace{e^{(k+1)\delta A}\Omega_0}_{R_{i+1}} \oplus \underbrace{\underbrace{e^{k\delta A}V}_{V_i} \oplus \underbrace{\left(e^{(k-1)\delta A}V \oplus \cdots \oplus V\right)}_{S_i}}_{S_{i+1}}.$$

- **4 Sequences:**

$$\begin{aligned} R_{i+1} &= e^{\delta A}R_i, \\ V_{i+1} &= e^{\delta A}V_i, \\ S_{i+1} &= S_i \oplus V_i, \\ \Omega_{i+1} &= R_{i+1} \oplus S_{i+1} \end{aligned} \qquad R_0 = \Omega_0,\ V_0 = V,\ S_0 = \{0\}$$

4-Sequence Algorithm

$$
\begin{aligned}
R_{k+1} &= e^{\delta A} R_k, \\
V_{k+1} &= e^{\delta A} V_k, \\
S_{k+1} &= S_k \oplus V_k, \\
\Omega_{k+1} &= R_{k+1} \oplus S_{k+1}
\end{aligned}
$$

- **Only transformations in R_k and V_k**
  - complexity independent of k
  - no overapproximation necessary
- **Only Minkowski sum in S_k and Ω_k**
  - growing number of generators, but no longer transformed
  - $O(Nn^3)$ instead of $O(N^2n^3)$

4-Sequence Algorithm

$$
\begin{aligned}
R_{k+1} &= e^{\delta A} R_k, \\
V_{k+1} &= e^{\delta A} V_k, \\
\hat{S}_{k+1} &= \hat{S}_k \oplus Approx(V_k), \\
\hat{\Omega}_{k+1} &= R_{k+1} \oplus \hat{S}_{k+1}
\end{aligned}
$$

- **Use overapproximation with**

$$Approx(X) \oplus Approx(Y) = Approx(X \oplus Y)$$

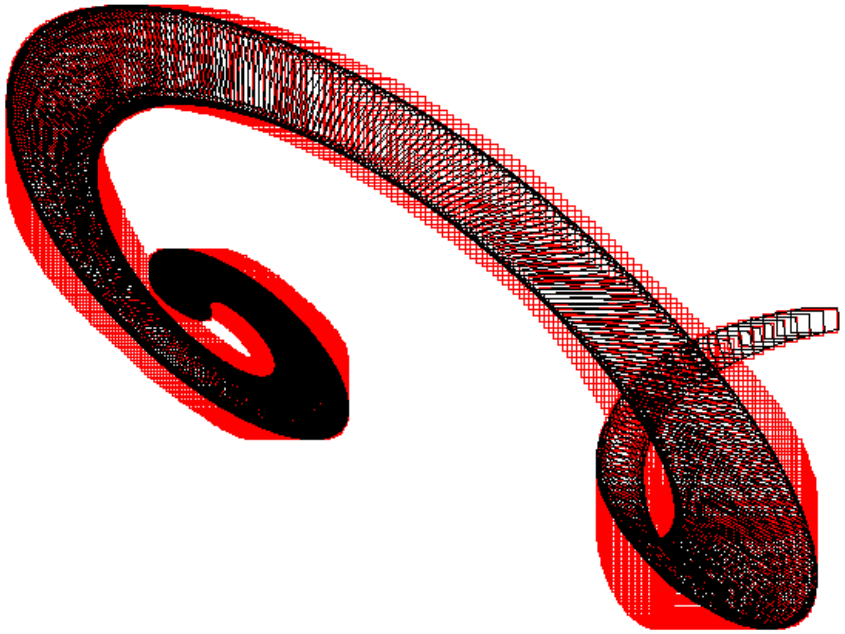
  - bounding box, octagonal, etc.

- **No accumulation of error:**

$$
\begin{aligned}
\hat{S}_k &= Approx(S_k) \\
\hat{\Omega}_k &\subseteq Approx(\Omega_k)
\end{aligned}
$$

Fighting the Wrapping Effect

- **Exact vs. overapproximation**
  - dimension 5 for 600 time steps
  - overapproximation with bounding box



Experimental Results

- **Time and memory for 100 steps**

	5	10	20	50	100	150	200
4-Sequence Zonotopes	0.0s	0.02s	0.11s	1.11s	8.43s	35.9s	136s
4-Sequence Box	0.0s	0.01s	0.07s	0.91s	8.08s	28.8s	131s
Zonotope, 20 Gen.	0.16s	0.61s	3.32s	22.6s	152s		

	5	10	20	50	100	150	200
4-Sequence Zonotopes	246KB	492KB	1.72MB	8.85MB	33.7MB	75.2MB	133MB
4-Sequence Box	246KB	246KB	246KB	492KB	983KB	2.21MB	3.69MB
Zonotope, 20 Gen.	737KB	2.46MB	8.36MB	44.5MB	177MB		

Outline

I. Hybrid Automata and Reachability

II. Reachability for Simple Dynamics

   a) Linear Hybrid Automata

   b) Piecewise Affine Hybrid Systems

III. Application to Complex Dynamics via Hybridization

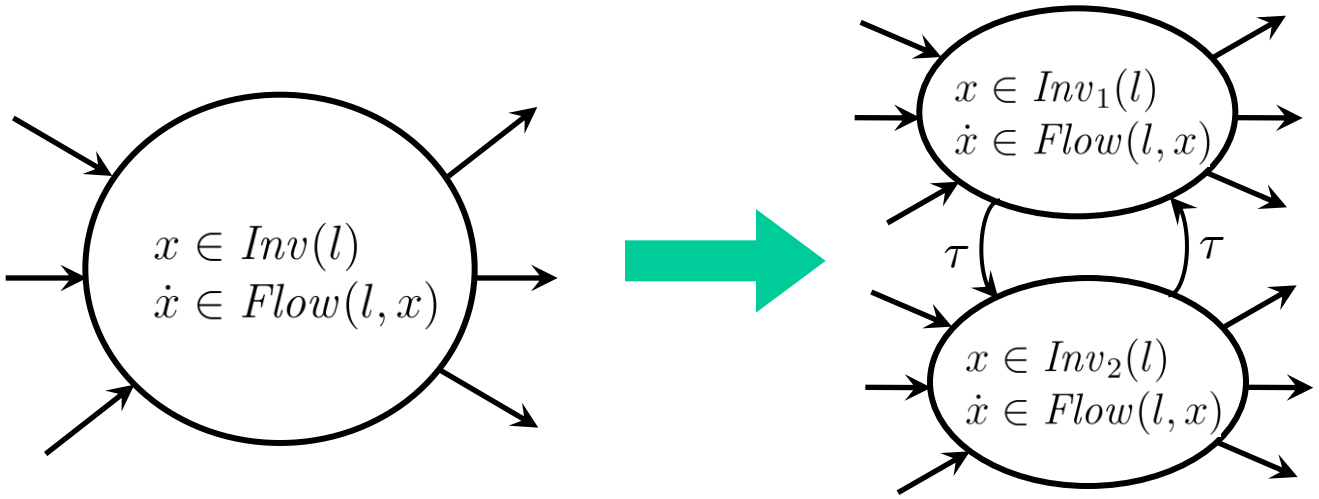
In this Chapter…

- **Complex nonlinear dynamics**
  - and how to overapproximate them with simpler dynamics

Hybridization

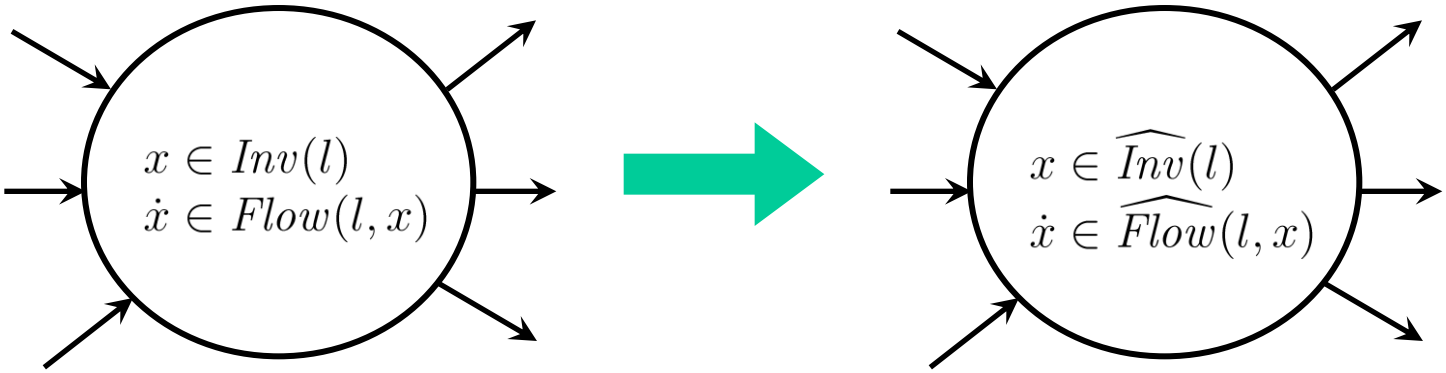
- **Goal: Overapproximation of H with**
  - simpler dynamics
  - approximation error $\leq \epsilon$
- **Observation:**
  - approximation error depends on size of invariant in each location
- **Approach:**
  - split locations until all invariants small enough
  - overapproximate dynamics in each location

Splitting Locations



- **same behavior as before if**
  - τ-transitions don't change variables and are unobservable
  - $Inv_1 \cup Inv_2 = Inv$ (and some details)

Overapproximating Dynamics



- **same or more behavior as before if**

$$\begin{aligned} Inv(l) &\subseteq \widehat{Inv}(l) \\ Flow(l, x) &\subseteq \widehat{Flow}(l, x) \end{aligned}$$

Some Approximation Results

- **Reachable set of the hybridization overapproximates the reachable set of H**
- **On bounded time interval $[0,T]$ the approximation error is in $O(\epsilon\ exp(T))$**
  - approximation diverges on an unbounded time interval…
- **Unless the system has a global attractor**
  - accumulation of approximation error is compensated by contraction of the dynamics.
  - reachable set on unbounded time interval can be approximated arbitrarily close

From Affine to LHA-Dynamics

$$\dot{x} \in Ax + B, \quad B \subseteq \mathbb{R}^n \quad \Rightarrow \quad \dot{x} \in C, \quad C \subseteq \mathbb{R}^n$$

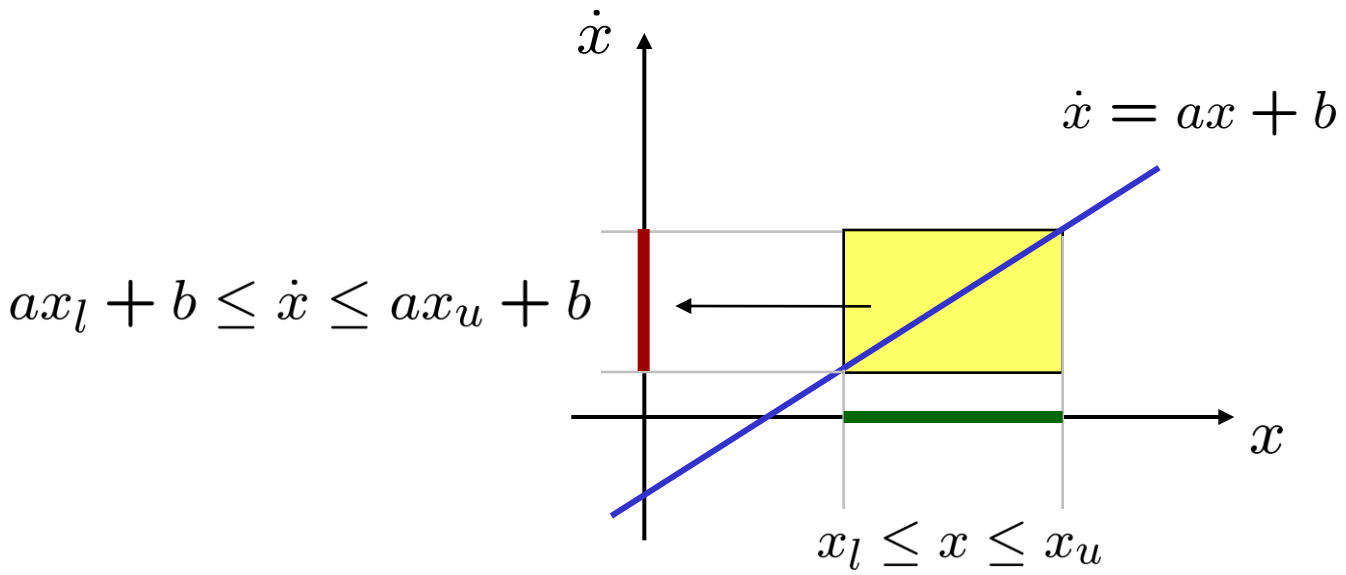
- **By definition** $x \in Inv(l)$**:**
  - overapproximation

$$C = \{x' \mid \exists x \in Inv(l) : x' \in Ax + B\}$$

- **If** B,Inv **polyhedra**
  - C polyhedron
  - $O(exp(n))$

From Affine to LHA-Dynamics

$$\dot{x} \in Ax + B, \quad B \subseteq \mathbb{R}^n \quad \Rightarrow \quad \dot{x} \in C, \quad C \subseteq \mathbb{R}^n$$



Hybridization with LHA

- **Bouncing Ball Dynamics**

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

  - dynamics of x are affine (depend on v).
- **Invariant:** $x \geq 0$
  - no restriction on $v \Rightarrow \dot{x} \in \mathbb{R}$
  - entire invariant reachable

Hybridization with LHA

- **Bouncing Ball Dynamics**

$$\begin{aligned} \dot{x} &= v \\ \dot{v} &= -g \end{aligned}$$

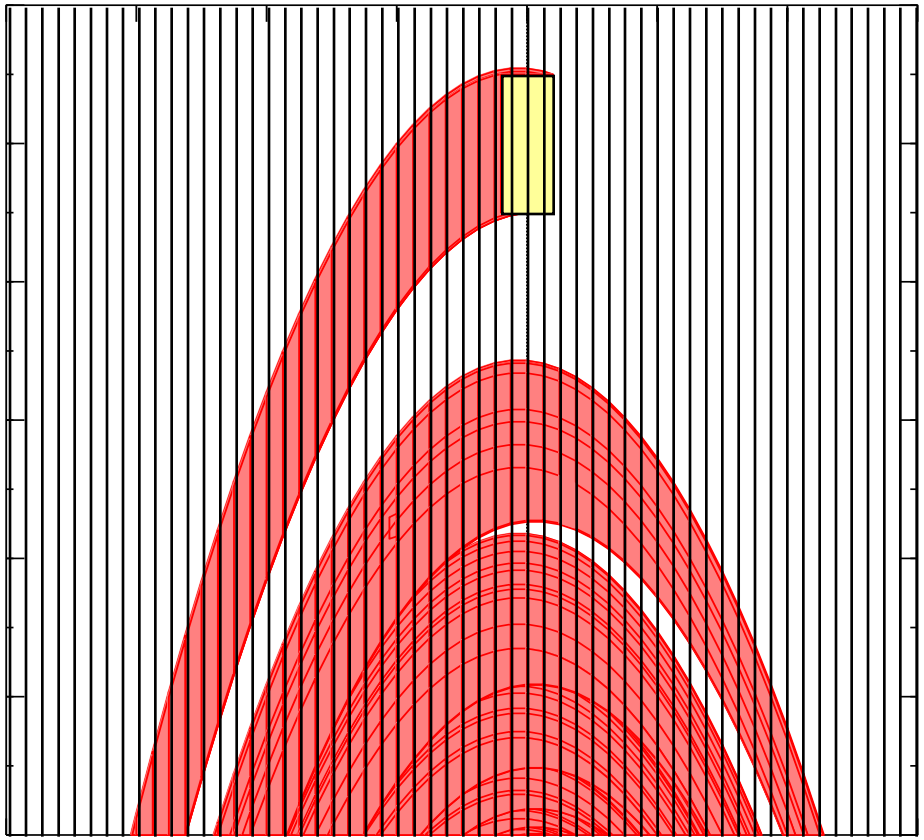
- **Split v–axis in K parts**
  - on bounded subset $v \in [-2,2]$

- **Arbitrary accuracy for small enough K**

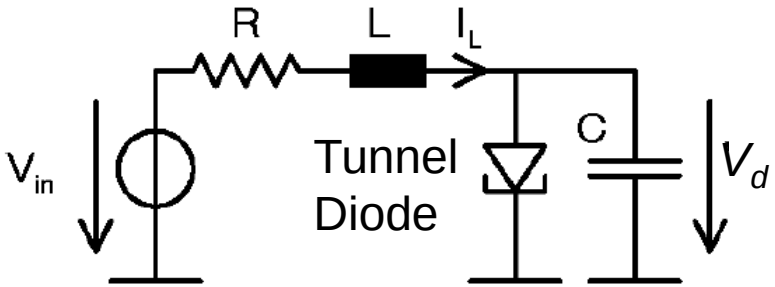
$$\dot{x} \in \{v \pm 4/K\}$$
$$K \to \infty \quad \Rightarrow \quad \dot{x} \to v$$

Hybridization with LHA

- **Bouncing Ball – Reachable states for $K=64$:**



Tunnel Diode Oscillator



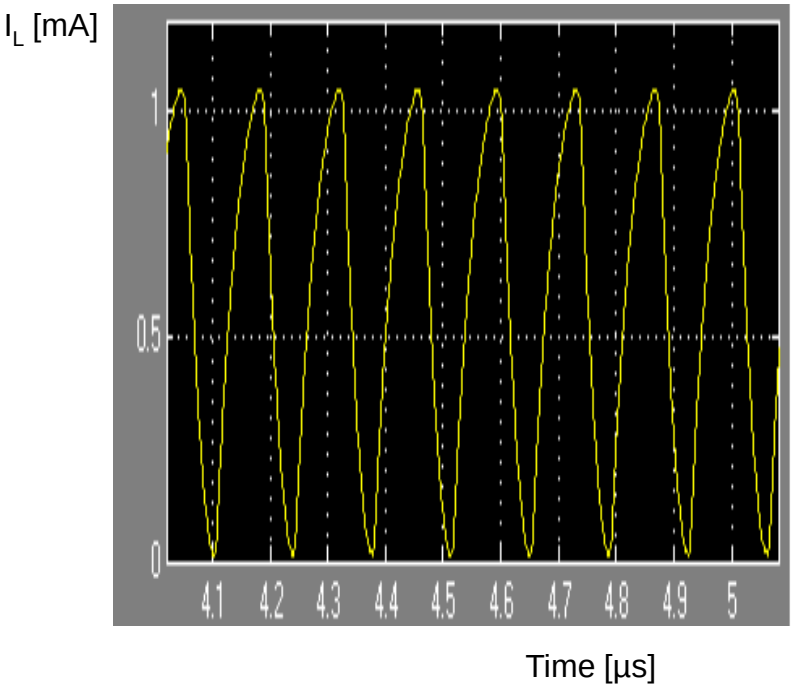
$$\dot{V}_C = \frac{1}{C}\left(-I_d(V_C) + I_L\right)$$

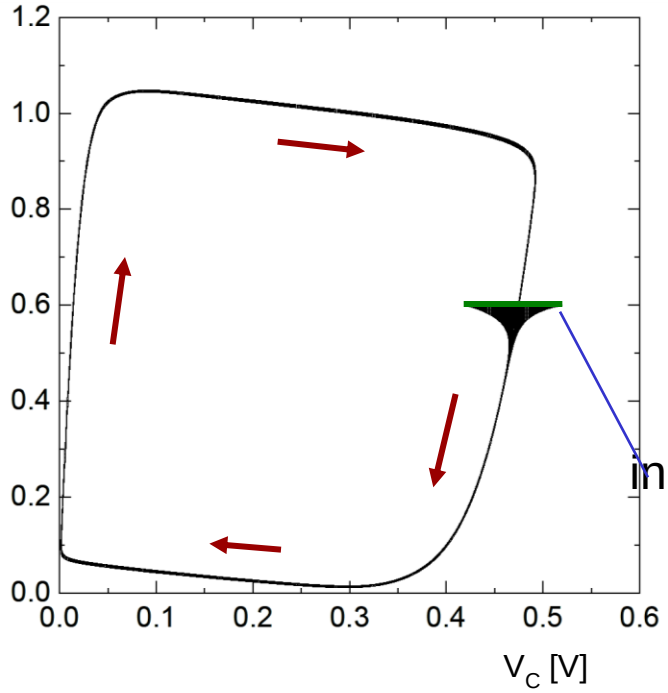
$$\dot{I}_L = \frac{1}{L}\left(-V_C - RI_L + V_{in}\right)$$

- **What are good parameters?**
  - startup conditions
  - parameter variations
  - disturbances

Tunnel Diode Oscillator

R=0.20Ω ⇒ Oscillation

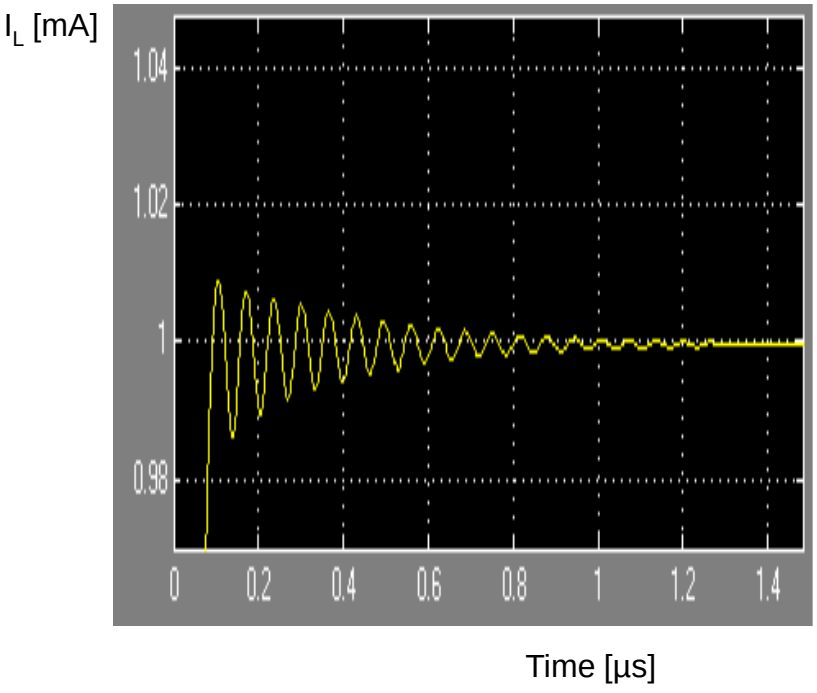


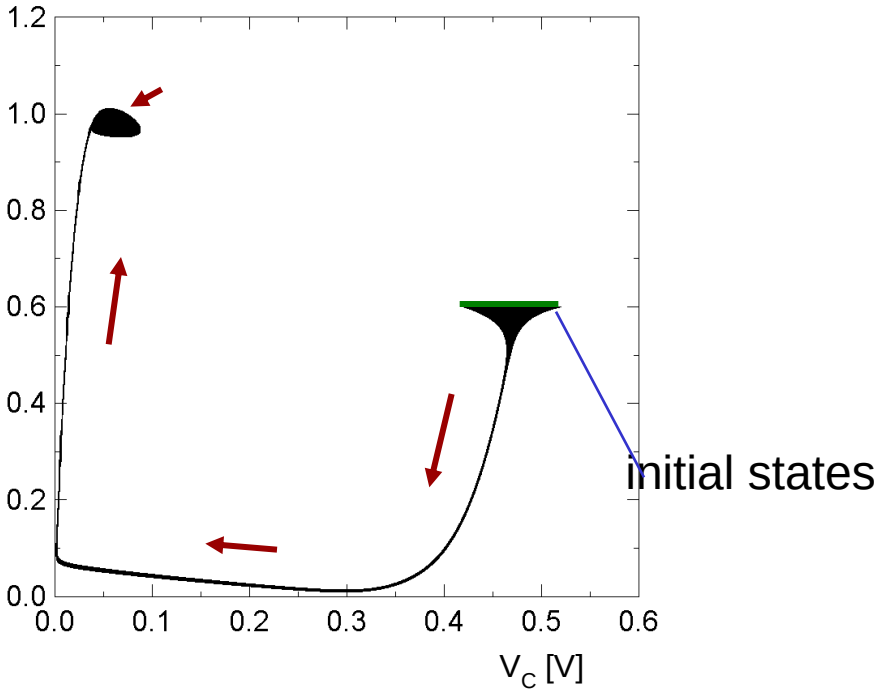


initial states

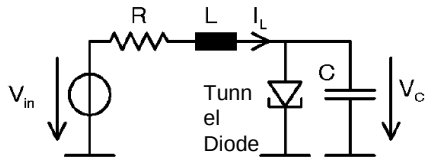
Tunnel Diode Oscillator

R=0.24Ω ⇒ Stable equilibrium

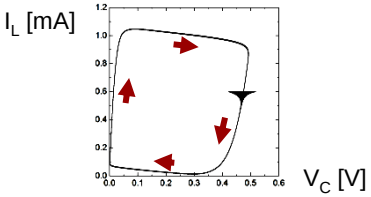




Tunnel Diode Oscillator



$$\dot{V}_C = \tfrac{1}{C}\left(-I_d(V_C) + I_L\right)$$
$$\dot{I}_L = \tfrac{1}{L}\left(-V_C - RI_L + V_{in}\right)$$



- **Oscillation**
- **Jitter**
- **…**

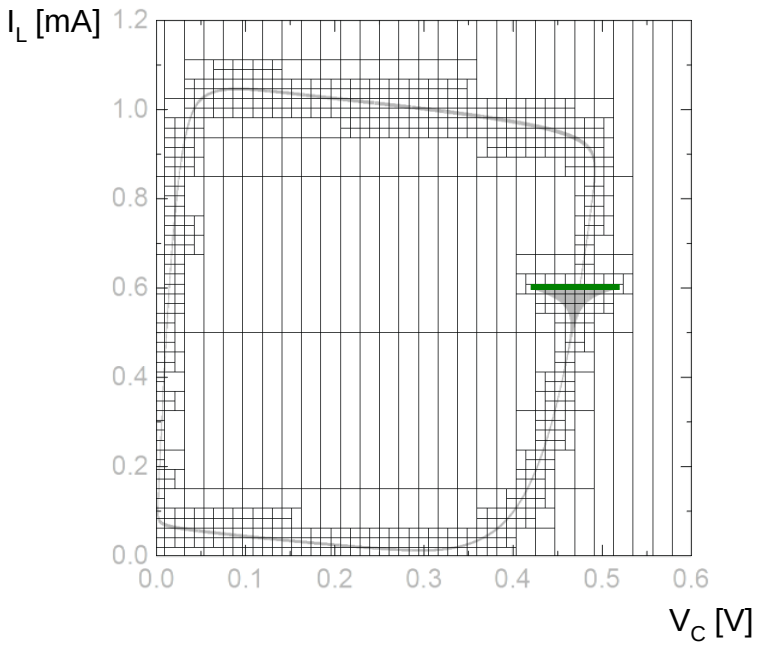
Analog/Mixed Signal Circuit

Formal Model

Reachability Analysis

Guaranteed Safety Property

Reachability Analysis

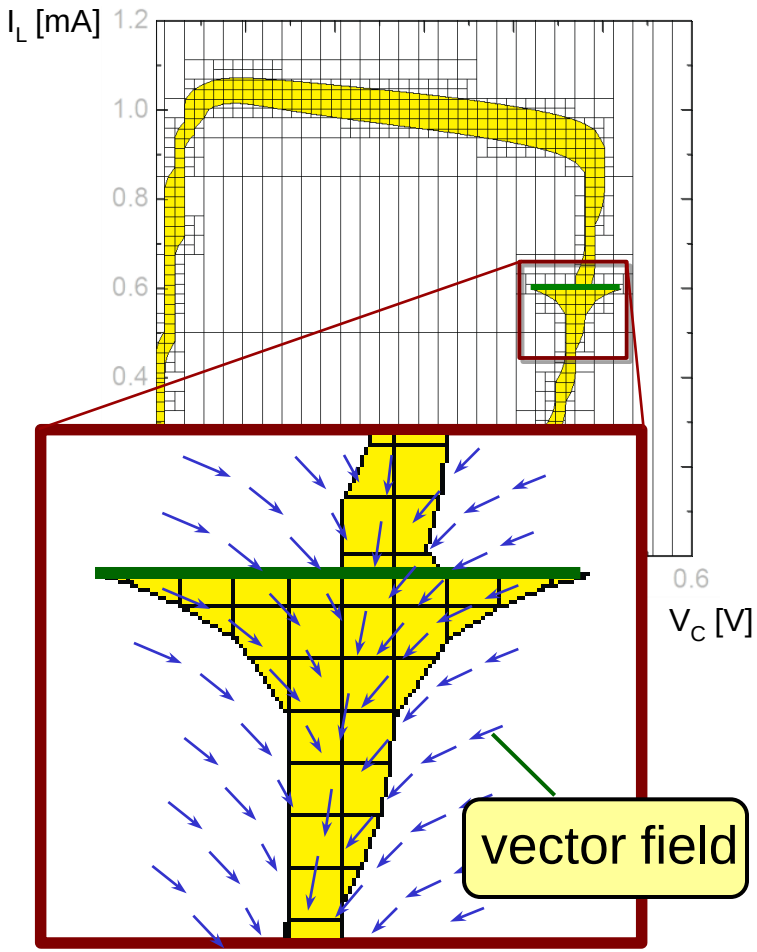


1. Hybridization

- Partition State Space (on the fly)
- Switching between

$\Rightarrow$ Hybrid System

Reachability Analysis



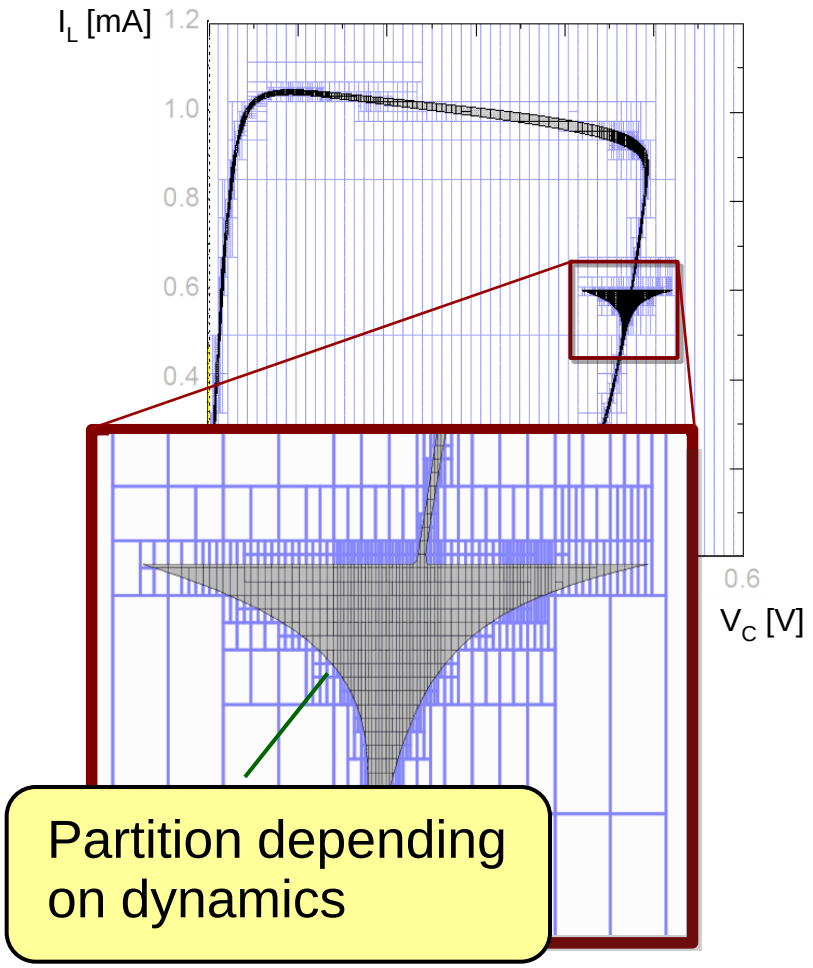
1. **Hybridization**
   - Partition State Space (on the fly)
   - Switching between

   ⇒ Hybrid System

2. **Overapproximation**
   - Linear Hybrid Automata

⇒ **Polyhedral enclosure of actual trajectories**

Reachability Analysis



- **Efficiency through**
  - adapting partitions to dynamics
  - overapproximation of complex polyhedra with simplified polyhedra
- **Good performance**
  - Reachability with high accuracy in 72s, 127MB

Bibliography

- **Hybrid Systems Theory**
  - Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. Theoretical Computer Science 138:3-34, 1995
  - Thomas A. Henzinger. The theory of hybrid automata. Proceedings of the 11th Annual Symposium on Logic in Computer Science (LICS), IEEE Computer Society Press, 1996, pp. 278-292
- **Linear Hybrid Automata**
  - Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi, HyTech: The next generation. RTSS'95
  - Goran Frehse. PHAVer: Algorithmic Verification of Hybrid Systems past HyTech. HSCC'05

Bibliography

- **Affine Dynamics**
  - E. Asarin, O. Bournez, T. Dang, and O. Maler. Approximate Reachability Analysis of Piecewise-Linear Dynamical Systems. HSCC'00
  - A. Girard, C. Le Guernic, and O. Maler. Efficient computation of reachable sets of linear time-invariant systems with inputs. HSCC'06
- **Hybridization and Nonlinear Dynamics**
  - Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. IEEE Transactions on Automatic Control 43:540-554, 1998
  - E. Asarin, T. Dang, and A. Girard. Hybridization methods for the analysis of nonlinear systems. Acta Informatica, 43(7):451-476, 2007

Verification Tools for Hybrid Systems

- **HyTech: LHA**
  - http://embedded.eecs.berkeley.edu/research/hytech/
- **PHAVer: LHA + affine dynamics**
  - http://www-verimag.imag.fr/~frehse/
- **d/dt: affine dynamics + controller synthesis**
  - http://www-verimag.imag.fr/~tdang/Tool-ddt/ddt.html
- **Matisse Toolbox: zonotopes**
  - http://www.seas.upenn.edu/~agirard/Software/MATISSE/
- **HSOLVER: nonlinear systems**
  - http://hsolver.sourceforge.net/