Hybrid Systems
Modeling, Analysis and Control

Radu Grosu
Vienna University of Technology

Lecture 10
Reachability with Affine Dynamics

HA with Affine Dynamics

Flows of continuous variables: $\dot{x} = \mathbf{A}x + \mathbf{B}u$

Invariants and guards: $\mathbf{A}x \leq \mathbf{c}$

Actions: $x := \mathbf{A}x$

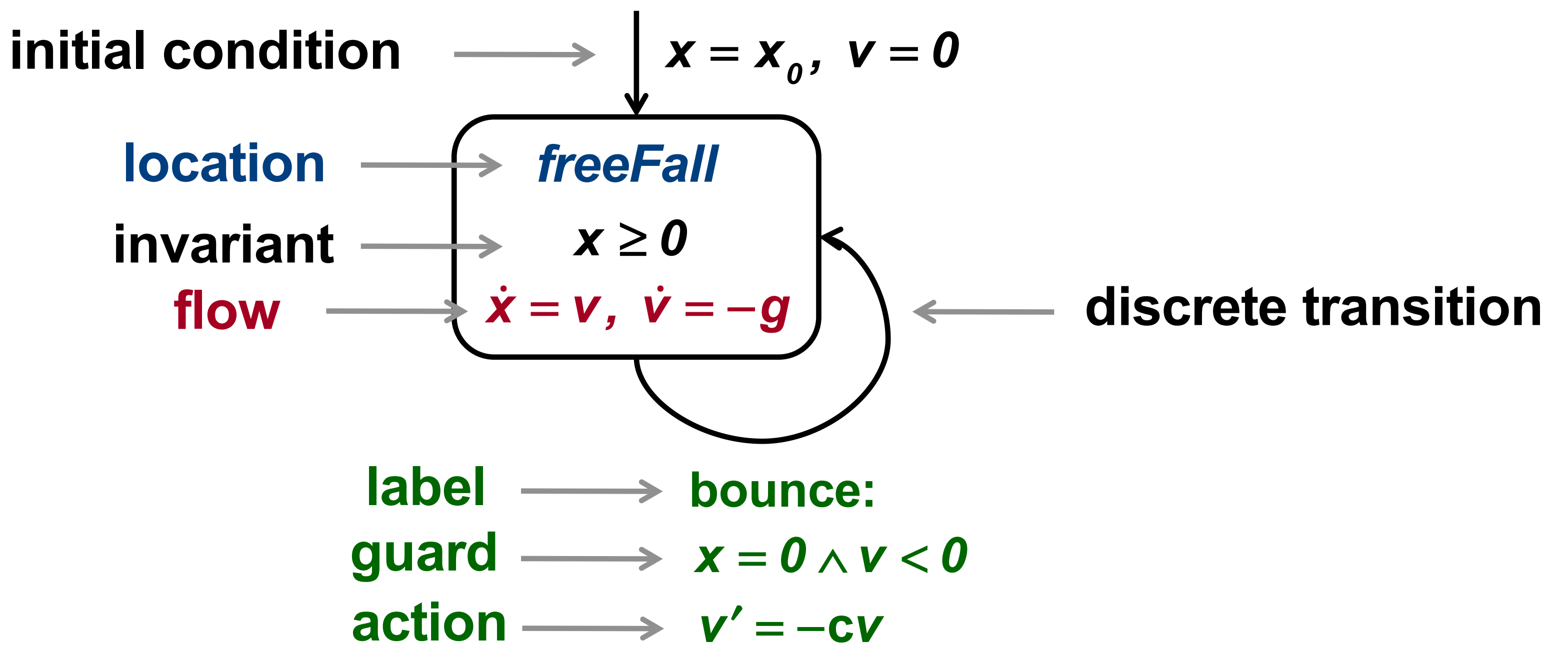
Symbolic representation: convex sets (e.g. polytopes)

Reachability: A semi algorithm

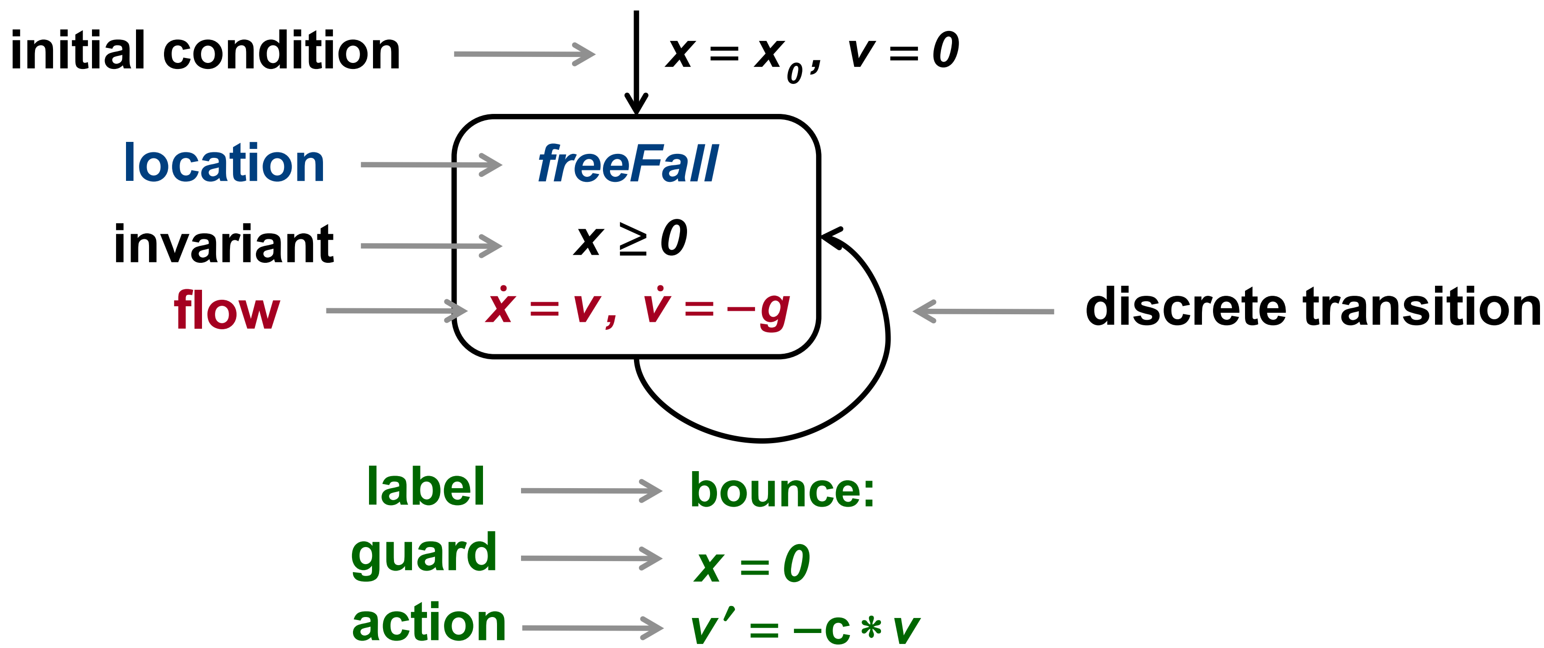
Methodolgy: Exact time elapse only at discrete time

Tools: SpaceEx

Bouncing Ball: HA with Affine Dynamics



Bouncing Ball: Associated Program



float $x = x_0,\ v = 0;\ d = d_0;$ // initial condition

while true { // main loop

  while ($x \geq 0$) { // free fall

    $x = x + v * d;$ // $\dot{x} = v$

    $v = v - g * d;$ } // $\dot{v} = -g$

  $v = -c * v;\ x = 0;$ } // bounce

Linear Dynamics

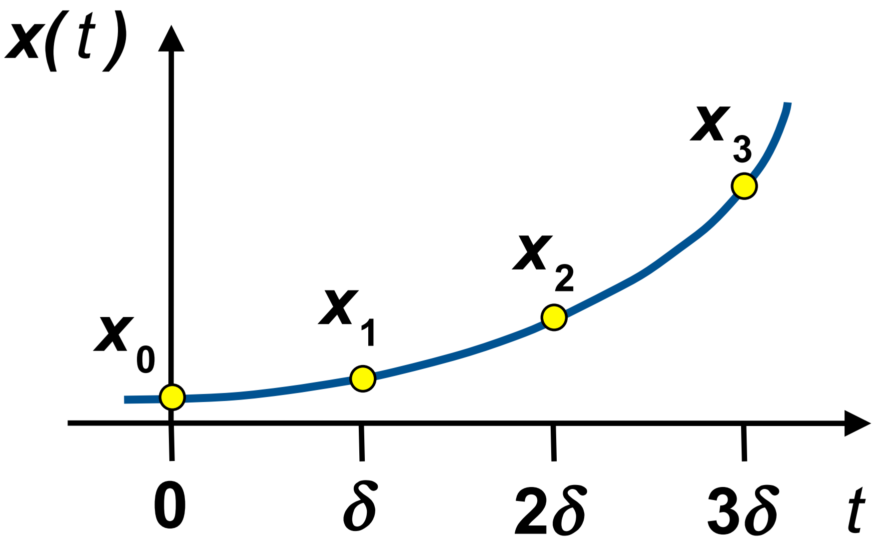
Autonomous part of dynamics: $\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x}, \quad \boldsymbol{x} \in \mathbb{R}^n$

Analytic solution: $\boldsymbol{x}(t) = \mathrm{e}^{\mathbf{A}t}\, \boldsymbol{x}_0$

Time discretization:

$t = \delta k$

$\boldsymbol{x}(\delta(k+1)) = \mathrm{e}^{\mathbf{A}\delta}\, \boldsymbol{x}(\delta k)$



Recursive DT-Solution:

$\boldsymbol{x}(0) = \boldsymbol{x}_0$

$\boldsymbol{x}(k+1) = \mathrm{e}^{\mathbf{A}\delta}\, \boldsymbol{x}(k)$

↑

Multiplication with constant matrix = linear transformation

$\boldsymbol{x} = \mathbf{M}\boldsymbol{x}$ **As a program instruction**

Linear DT-Dynamics from Initial Set

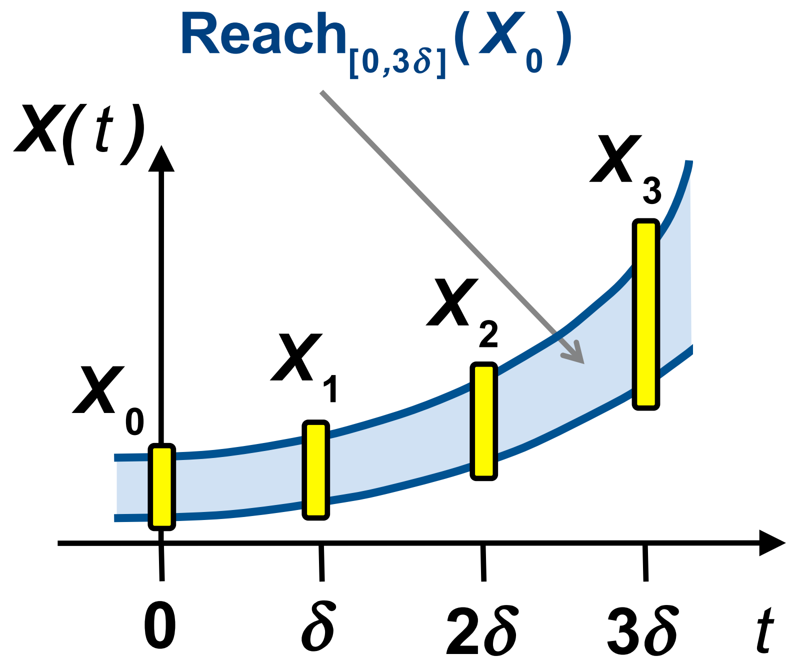
Recursive DT-Solution:

$$X(0) = X_0$$

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

Purely continuous systems:

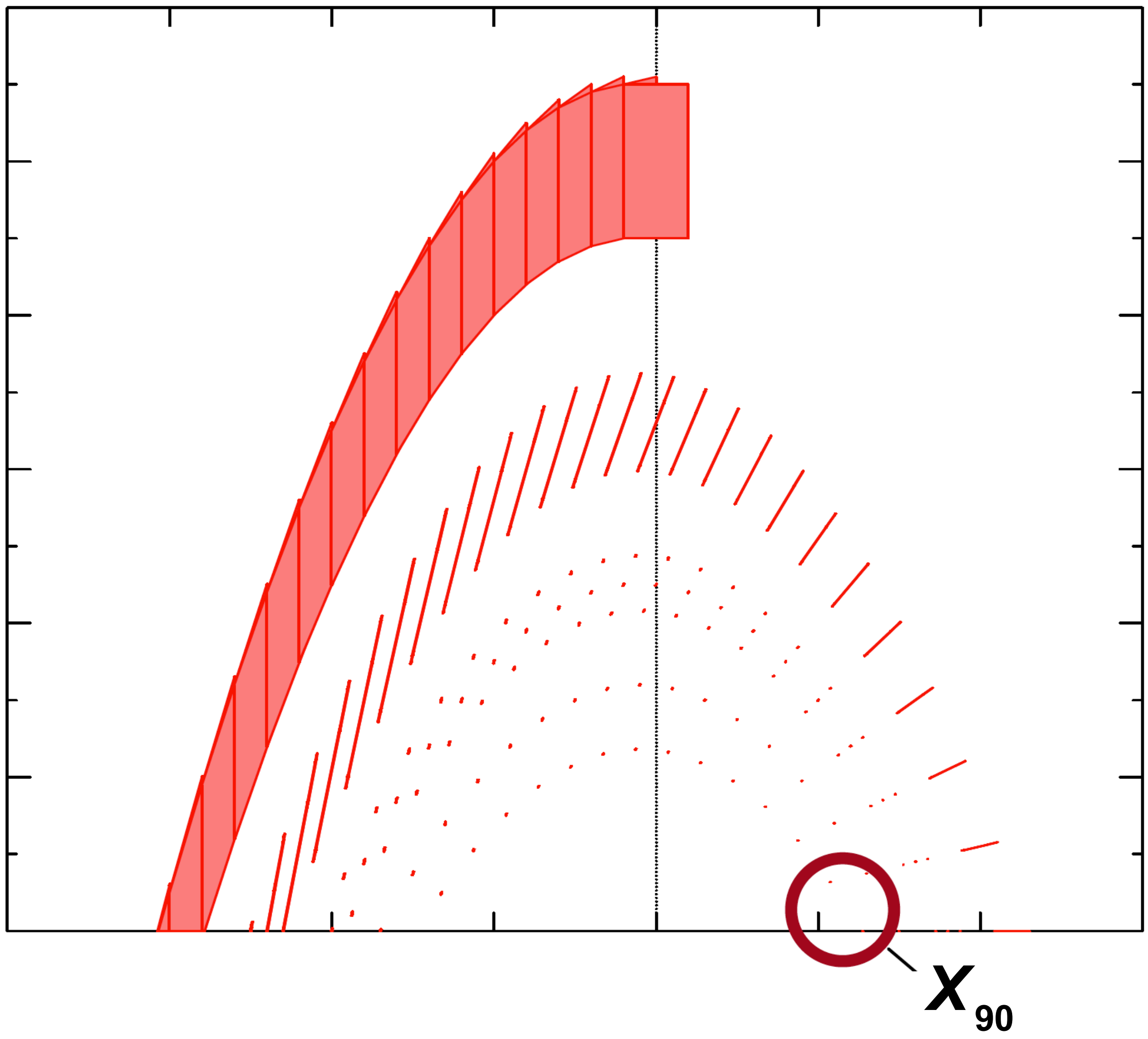
- Discrete solution acceptable
- x (t) is in $\varepsilon(\delta)$-neighbourhood of some X_k in the DT-solution



Not acceptable for hybrid systems:

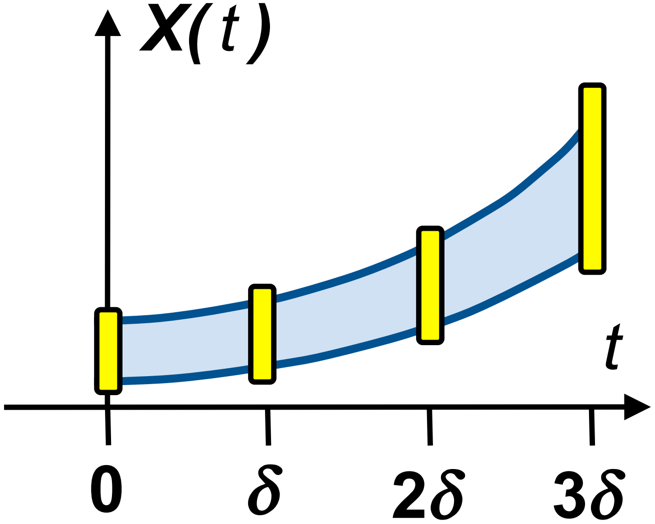
- Discrete transitions may fire between sampling times
- If transitions are missed, x (t) is not in $\varepsilon(\delta)$-neighbourhood

Bouncing Ball: Error in the DT-Solution



Polyhedral Flow-Pipe Approximation

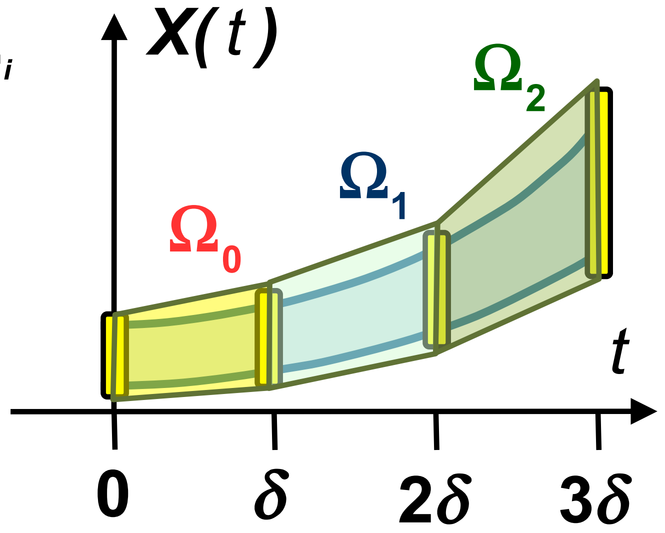
Divide: Reach$_{[0,N\delta]}(X_0)$ into N δ-segments



Polyhedral Flow-Pipe Approximation

Divide: Reach$_{[0,N\delta]}(X_0)$ into N δ-segments

Enclose: segments with convex polytopes Ω_i



Polyhedral Flow-Pipe Approximation

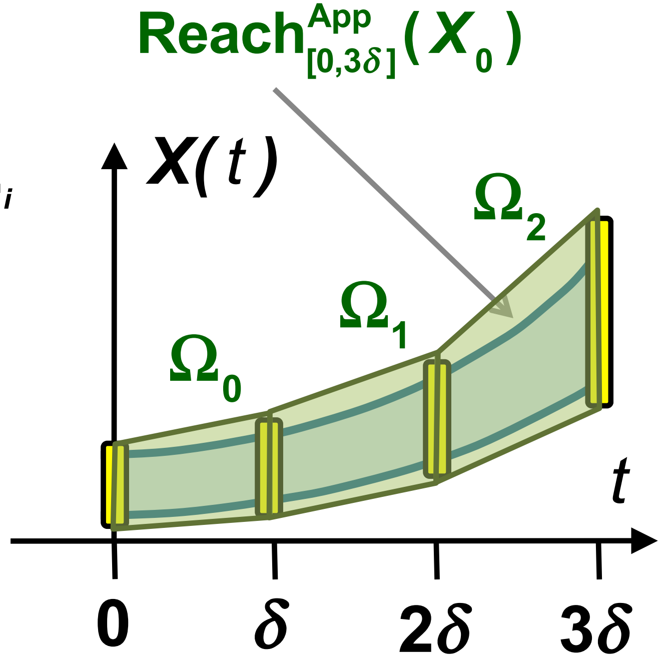
Divide: $\text{Reach}_{[0,N\delta]}(X_0)$ into N δ-segments

Enclose: segments with convex polytopes Ω_i

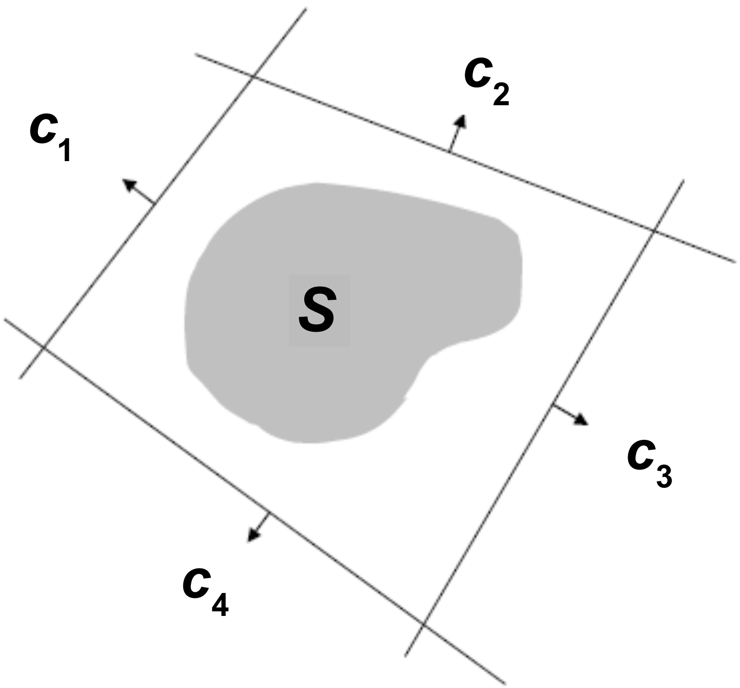
Union: $\text{Reach}^{App}_{[0,N\delta]}(X_0) = \cup_{i=0}^{N-1}\Omega_i$

Approx: $\text{Reach}_{[0,N\delta]}(X_0) \subseteq \text{Reach}^{App}_{[0,N\delta]}(X_0)$

Works for general nonlinear systems!



Wrapping Hyper-Planes Around a Set

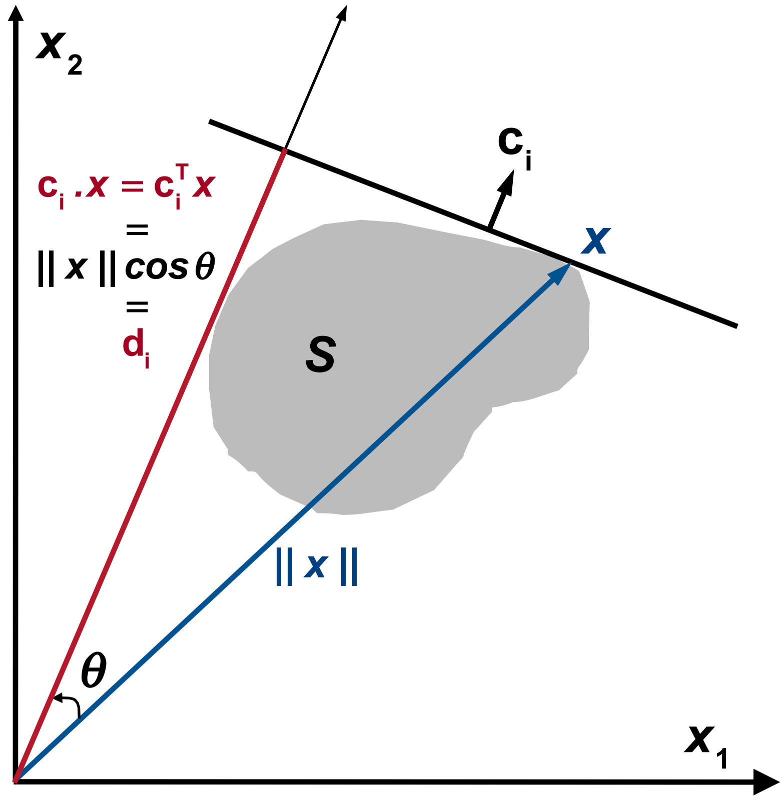


Step$_1$:

– Choose: directions $c_1 \ldots c_n$

– Construct: $C^T = [c_1 \ldots c_n]$

Wrapping Hyper-Planes Around a Set

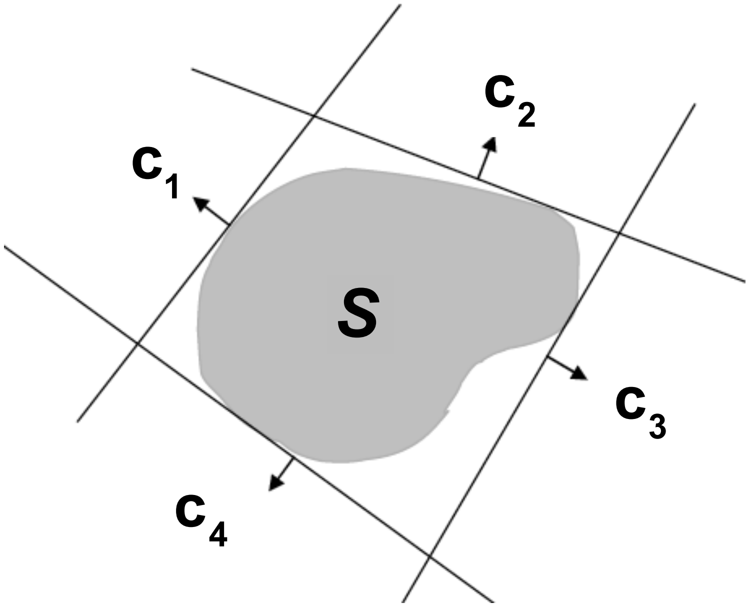


Step$_2$:

- **Compute optimal d:** in $C^T x \le d$ where $C^T = [c_1 ... c_4]$ and $x \in S$
- **Stepwise:** For each direction i

$$d_i = \max_{x \in S} c_i .x$$

Wrapping Hyper-Planes Around a Set



Step$_2$:

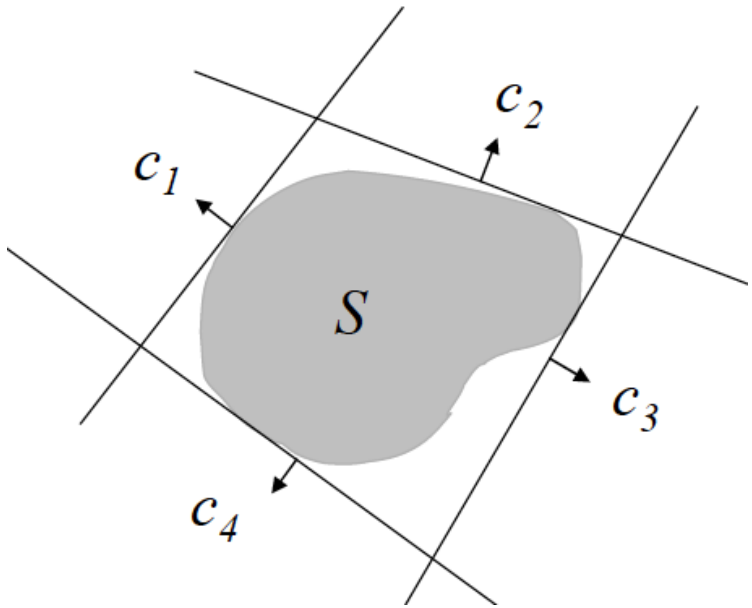
– **Compute optimal d:** in $C^T x \leq d$ where $C^T = [c_1 \ldots c_4]$ and $x \in S$

– **Stepwise:** For each direction i

$$d_i = \max_{x \in S} c_i . x$$

Wrapping a Reached Set

Given directions $c_1 \ldots c_n$, **wrap** $\text{Reach}_{[t_k, t_{k+1}]}(X_0)$ within a **polytope** by solving for each i the **optimisation problem**



$$d_i = \max_{x_0, t} c_i^T x(t, x_0)$$

$$x_0 \in X_0$$

$$t \in [t_k, t_{k+1}]$$

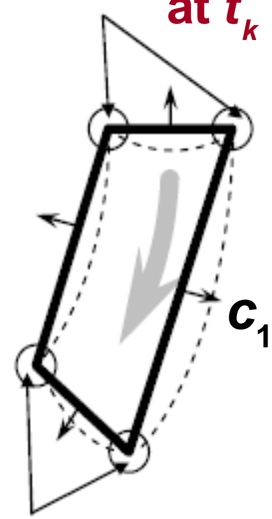
Optimization problem is solved by embedding simulation into objective function computation

Flow Pipe Segment Approximation

Step 1:

a. Simulate trajectories from each vertex of X_0

b. Take convex hull and identify outward normal vectors c_i



Step 2:

Solve optimization problem for each d_i

Flow-pipe segment approximated by

$$\{x \mid c_i^T x \le d_i, \forall i\}$$

Improvements for Linear Systems

Step 1:

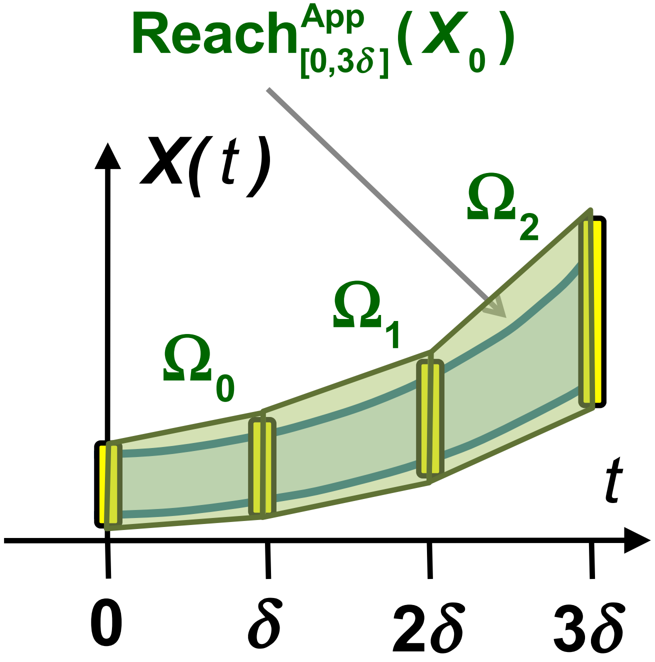
- Compute Ω_0 as discussed before, but

  $X(t) = e^{At} X_0$

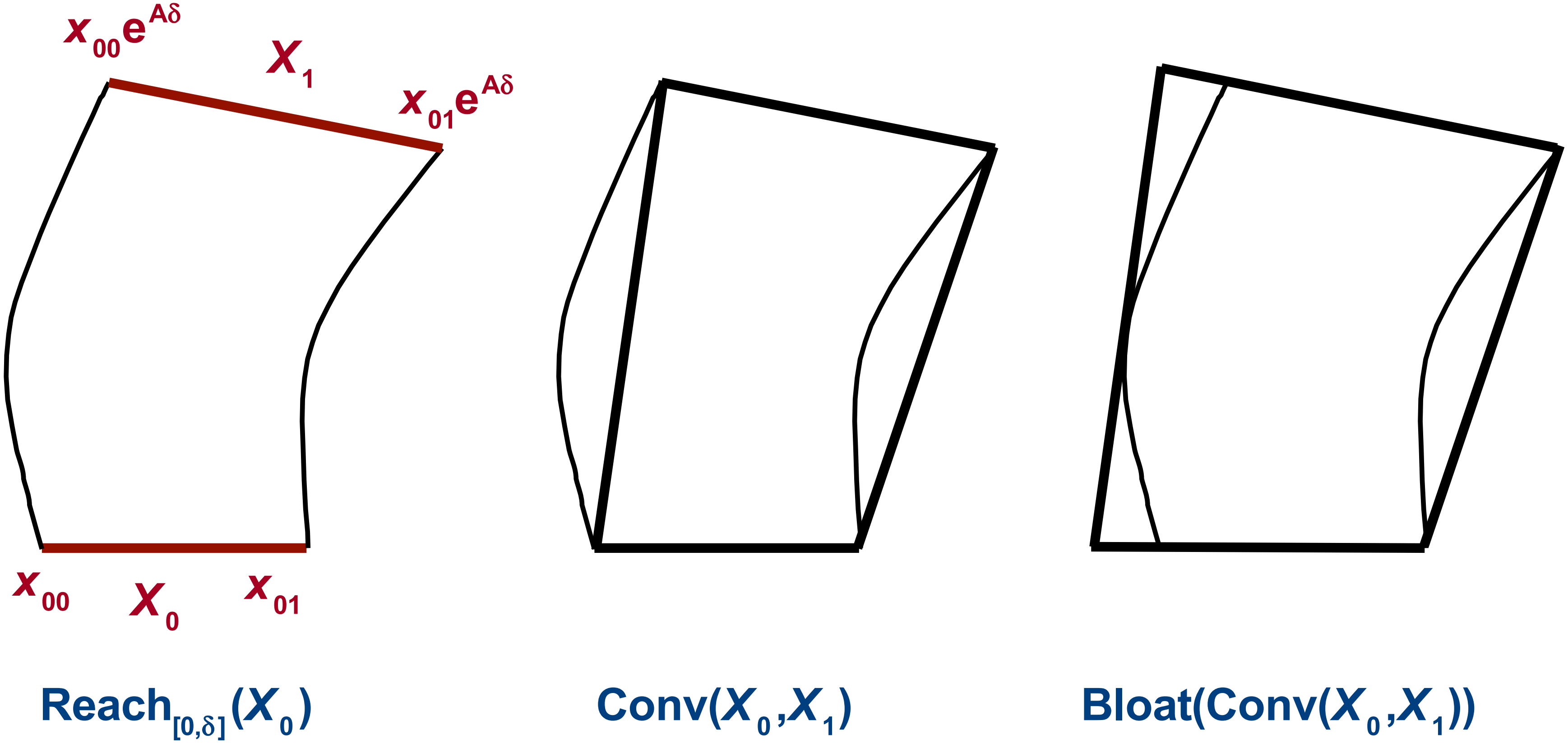
Step 2:

- Compute Ω_{k+1} by using the recurrence:

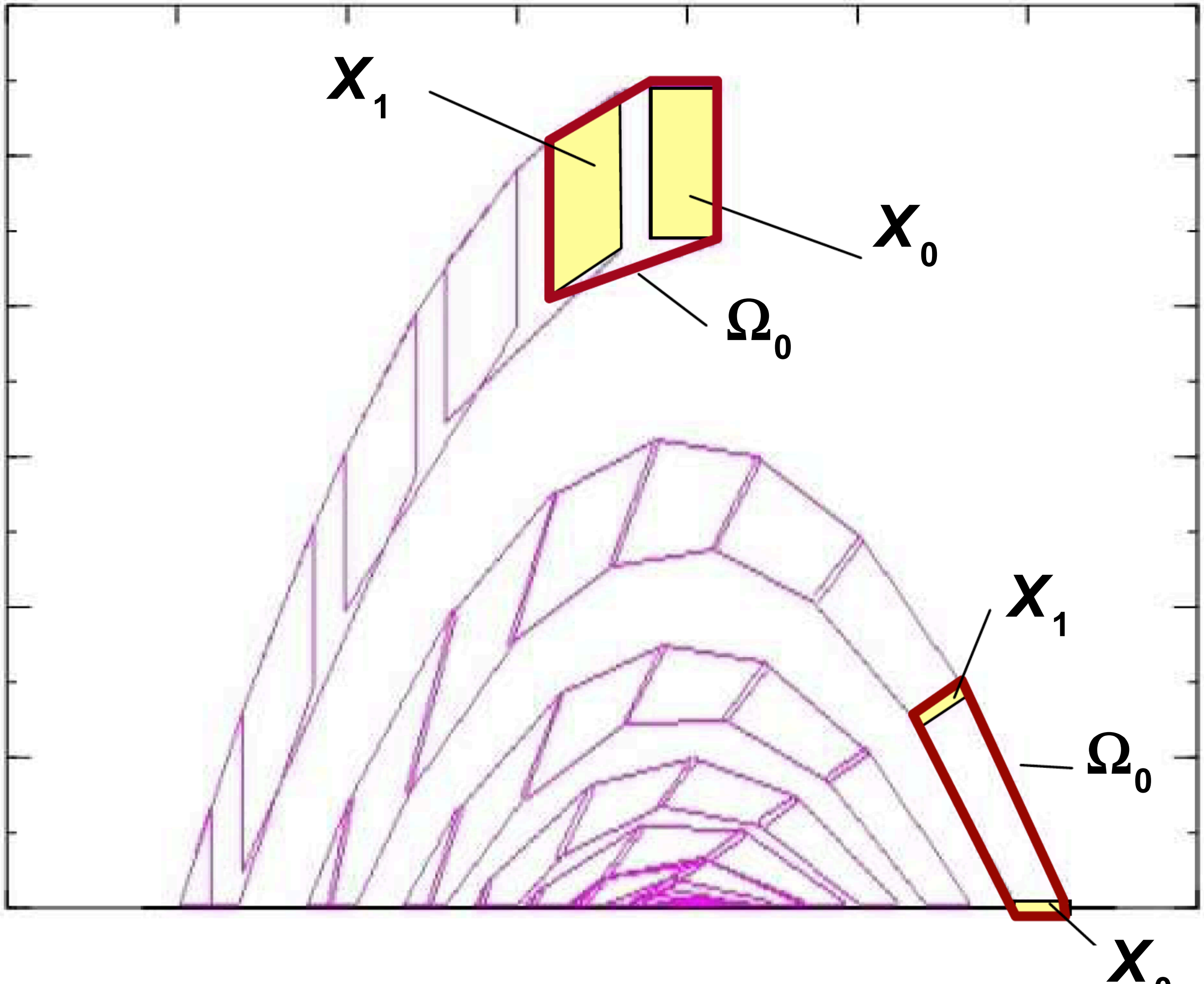
  $\Omega_{k+1} = e^{A\delta}\Omega_k$ (proof in Girard's Thesis)



Computation of Ω_0



Bouncing Ball: DT with Convex Hull



Adding Input

Flows of continuous state variables:

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}\boldsymbol{u}, \quad \boldsymbol{x} \in \mathbb{R}^{\mathrm{n}}, \boldsymbol{u} \in \mathbf{U} \subseteq \mathbb{R}^{\mathrm{p}}$$

where *x* is the state and *u* is the input

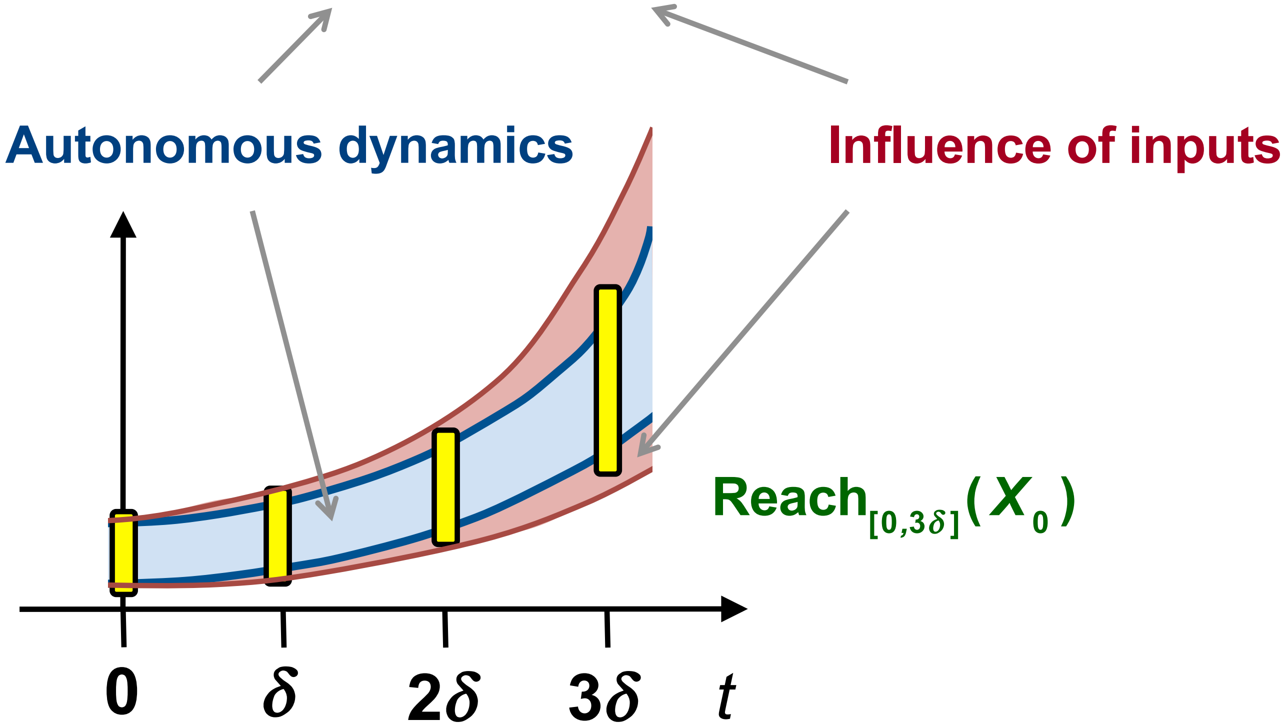
Input is treated as nondeterministic:

$$\dot{\boldsymbol{x}} = \mathbf{A}\boldsymbol{x} + \mathbf{B}\mathbf{U}, \quad \boldsymbol{x} \in \mathbb{R}^{\mathrm{n}}, \mathbf{U} \subseteq \mathbb{R}^{\mathrm{p}}$$

used later for overapproximating nonlinear dynamics

Analytic Solution

$$x(t) = e^{A\delta} x_0 + \int_0^{\delta} e^{A(\delta-\tau)} B u(\tau)\, d\tau$$



Nondeterminism Overapproximation

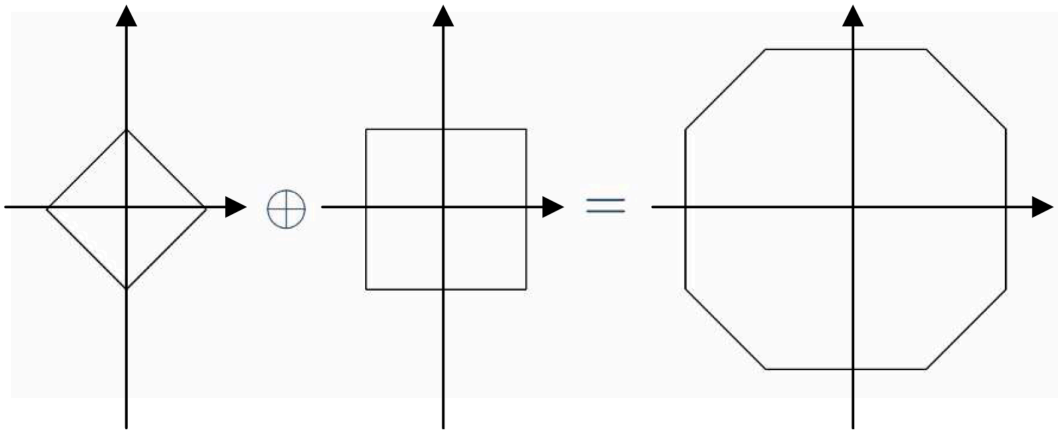
How far can the input push the system within δ?

$$V = \text{box with radius } \frac{e^{\|A\|\delta} - 1}{\|A\|} \sup_{u \in U} \|Bu\|$$

$$\Omega_0 = \text{Bloat}(\text{Conv}(X_0, e^{A\delta} X_0)) \oplus V$$

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

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



Implementing Reachability

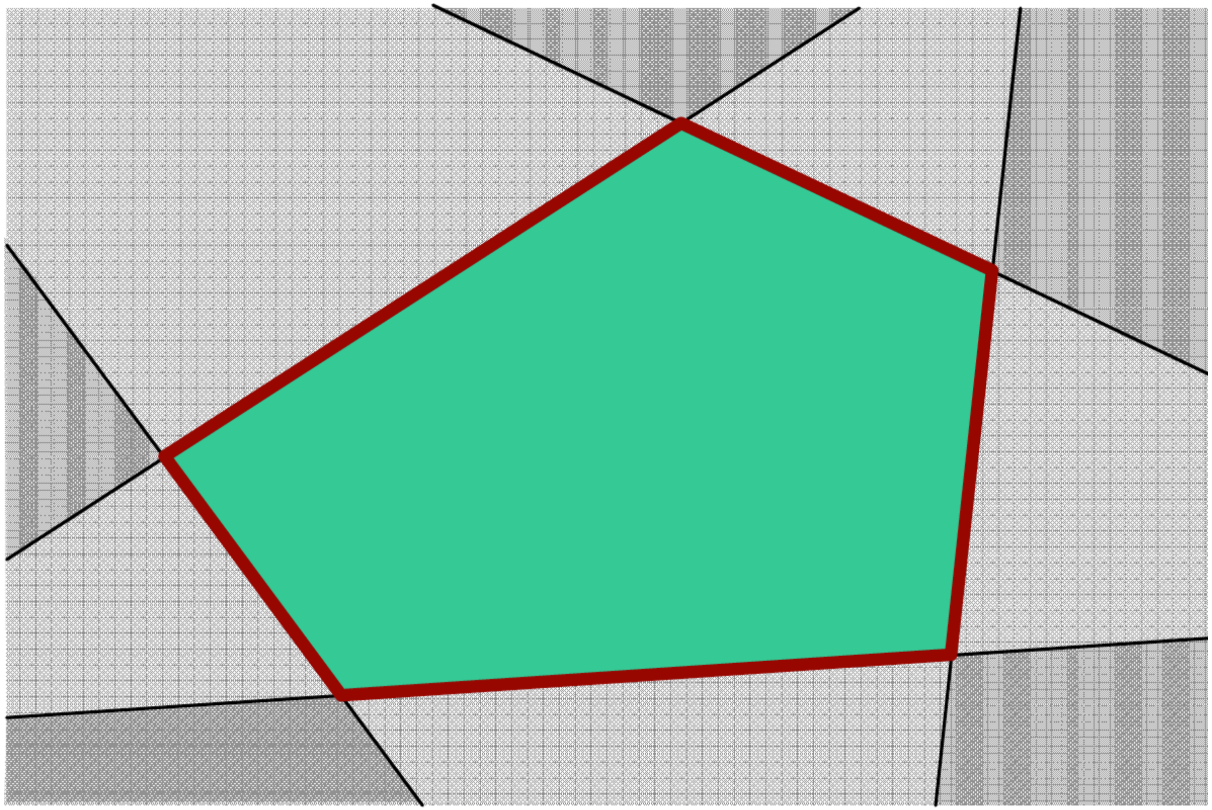
Find representation for continuous sets with:

- **Linear transformation:** $\Omega_{k+1} = \Phi\Omega_k$
- **Minkowski sum:** $\Omega_{k+1} = \Phi\Omega_k \oplus V$
- **Intersection with guards:** $\mathbf{A}x \leq \mathbf{c}$

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:

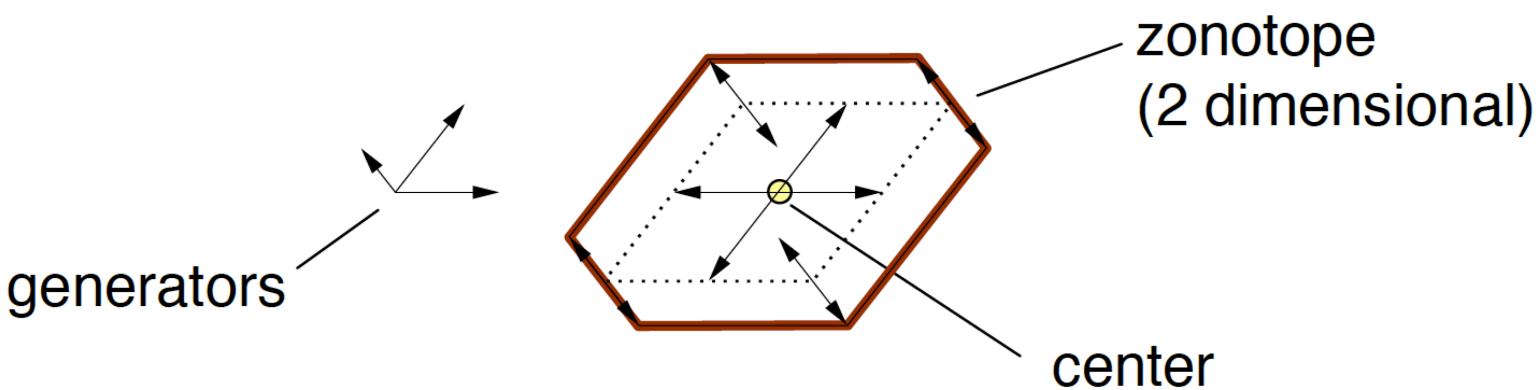
- Join lists of constraints
- O(1)

Zonotopes

Central-symmetric polyhedron:

$$Z = (c, < v_1, \ldots, v_m >) = \left\{ c + \sum_{i=1}^{m} \alpha_i v_i \mid -1 \le \alpha_i \le 1 \right\}$$

center generators



Operations on Zonotopes

Linear transformation:

- Transform generators: $\Phi Z = (\Phi c, < \Phi v_1, \ldots, \Phi v_m >)$
- $O(n^2)$

Minkowski sum:

- Join generator lists: $Z + Z' = (c + c', < v_1, \ldots, v_m, v'_1, \ldots, v'_{m'} >)$
- O(1)

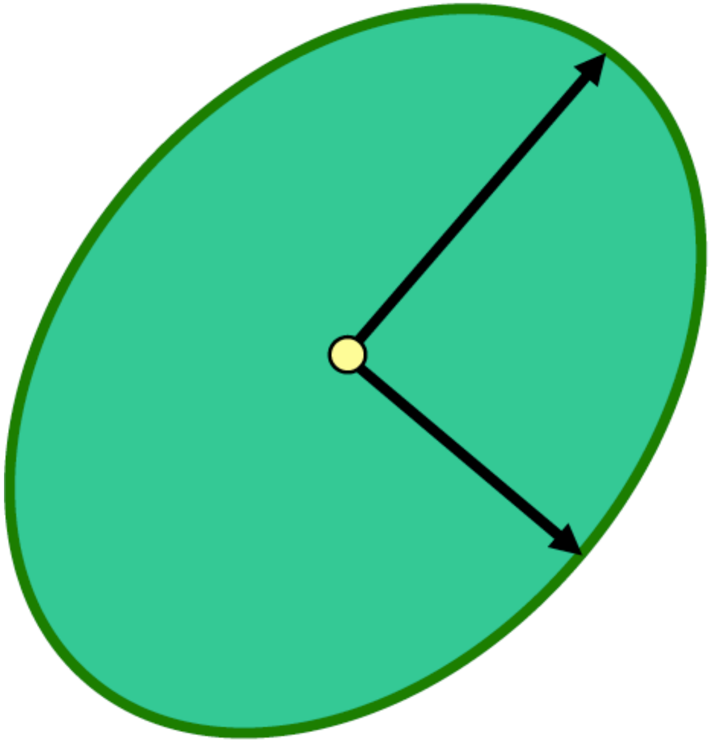
Intersection:

- Problem: intersection of zonotopes is not a zonotope
- Overapproximate

Ellipsoids

Quadratic form:

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



Operations on Ellipsoids

Linear transformation:

- Transform generators
- $O(n^2)$

Minkowski sum:

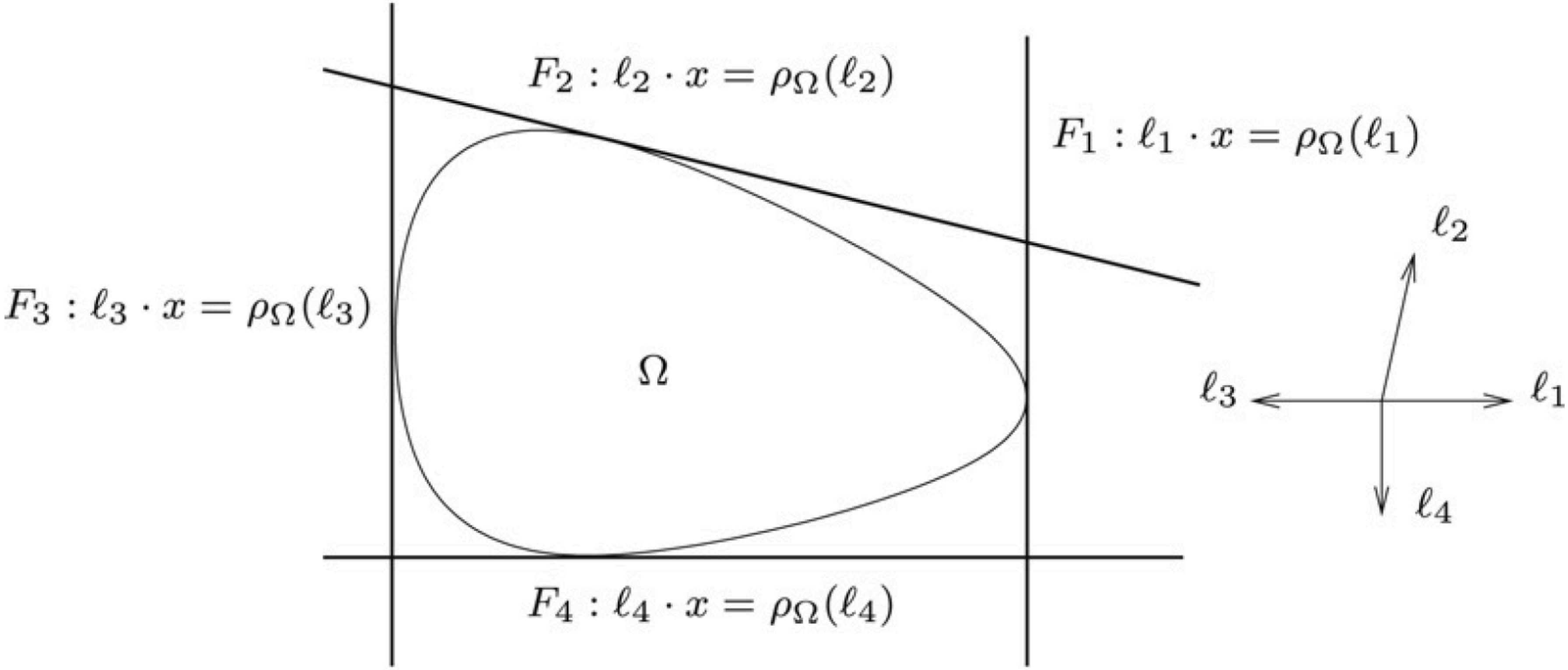
- Problem: result is not an ellipsoid
- Overapproximate

Intersection:

- Problem: intersection of ellipsoids is not an ellipsoid
- Overapproximate

Support Functions

Functional form (lazy evaluation): $\rho_{\Omega}(l) = \max_{x \in \Omega} l.x$



$$\Omega = \bigcap_{l \in \mathbb{R}^d} \{x \in \mathbb{R}^d \mid l.x \leq \rho_{\Omega}(l)\}$$

Support Functions

Unit ball for 2-norm $B_2 = \{x \in \mathbb{R}^d \mid ||x||_2 \leq 1\}$:

$$\rho_\Omega(l) = ||l||_2$$

Ellipsoid $\Omega = \{x \in \mathbb{R}^d \mid x^T Q x \leq 1\}$, Q **is positive definite:**

$$\rho_\Omega(l) = \sqrt{l^T Q l}$$

Hyperrectangle $\Omega = [-h_1, h_1] \times \cdots \times [-h_d, h_d]$, $h_i \in \mathbb{R}^d_+$:

$$\rho_\Omega(l) = \sum_{j=1}^{d} | h_j l_j |$$

Zonotope $\Omega = \{ \sum_{j=1}^{r} \alpha_i g_i \mid \alpha_i \in [-1,1]\}$, **generators** $g_i \in \mathbb{R}^d$:

$$\rho_\Omega(l) = \sum_{j=1}^{r} | g_j . l |$$

Polytope $\Omega = \{ x \in \mathbb{R}^d \mid C x \leq d\}$:

$$\rho_\Omega(l) = \max_{Cx \leq d} l.x$$

Operations on Support Functions

Linear transformation:

- $\rho_{A\Omega}(l) = \rho_{\Omega}(A^T l)$
- $\rho_{\lambda\Omega}(l) = \rho_{\Omega}(\lambda l) = \lambda\rho_{\Omega}(l)$

Minkowski sum:

- $\rho_{\Omega\oplus\Omega'}(l) = \rho_{\Omega}(l) + \rho_{\Omega'}(l)$

Convex hull:

- $\rho_{CH(\Omega,\Omega')}(l) = \max\{\rho_{\Omega}(l), \rho_{\Omega'}(l)\}$

Intersection:

- **Reduces to an optimization problem**

Implementing Reachability

Complexity of 1-step of time elapse:

- **Polyhedra: O(exp(n))**
- **Zonotopes: $O(n^2)$**

Problem:

- **With each iteration, Ω_k gets more complex**
- $\Omega_{k+1} = e^{A\delta}\Omega_k \oplus V$

Minkowski sum increases the number of:

- **Polyhedra: constraints**
- **Zonotopes: generators**

Wrapping Effect

Fight complexity by overapproximation

Overapproximated sequence:

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

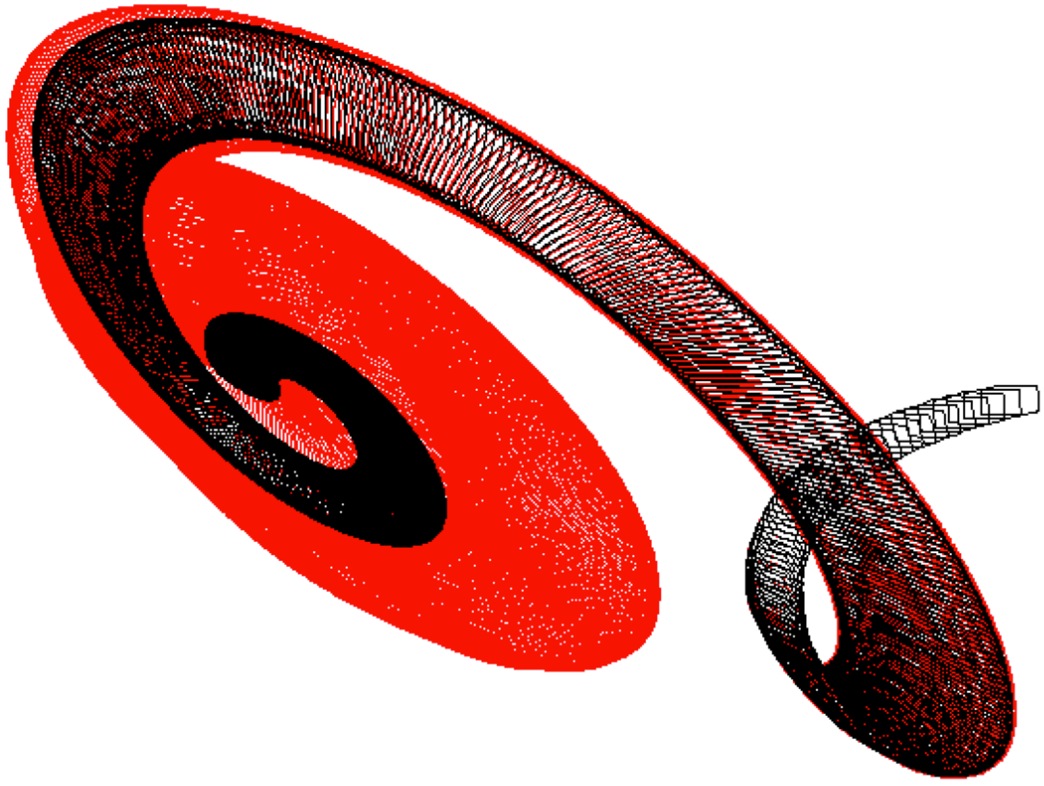
Accumulation of approximations:

- **Wrapping effect**
- **Exponential increase in approximation error!**

Wrapping Effect

Exact versus overapproximation

- Dimension 5 for 600 time steps
- Overapproximation with 100 generators



Wrapping Effect

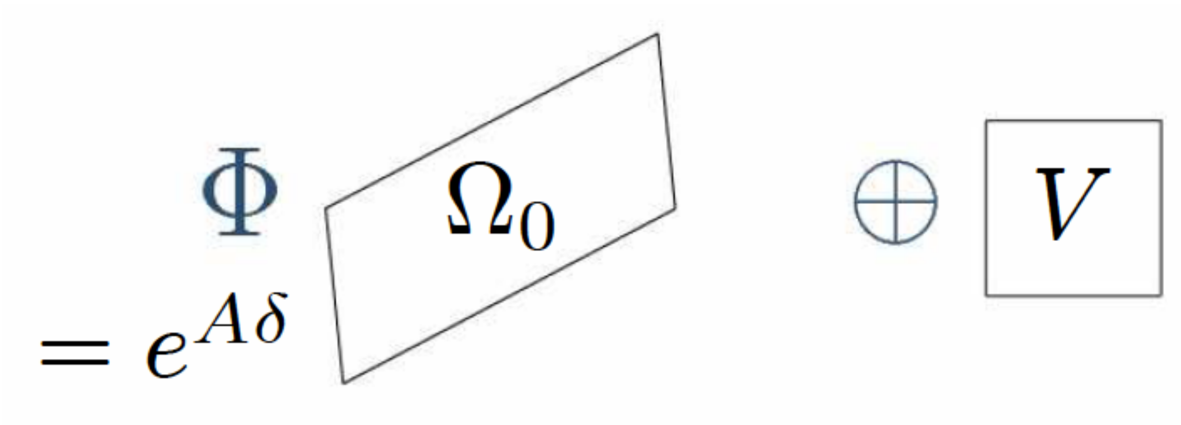
$$\hat{\Omega}_{k+1} = \text{Approx}(e^{A\delta}\hat{\Omega}_k \oplus V)$$

How does error accumulate?

- **Linear transformation:** Scaling error up (exponential)
- **Minkowski sum:** adding V adds more error

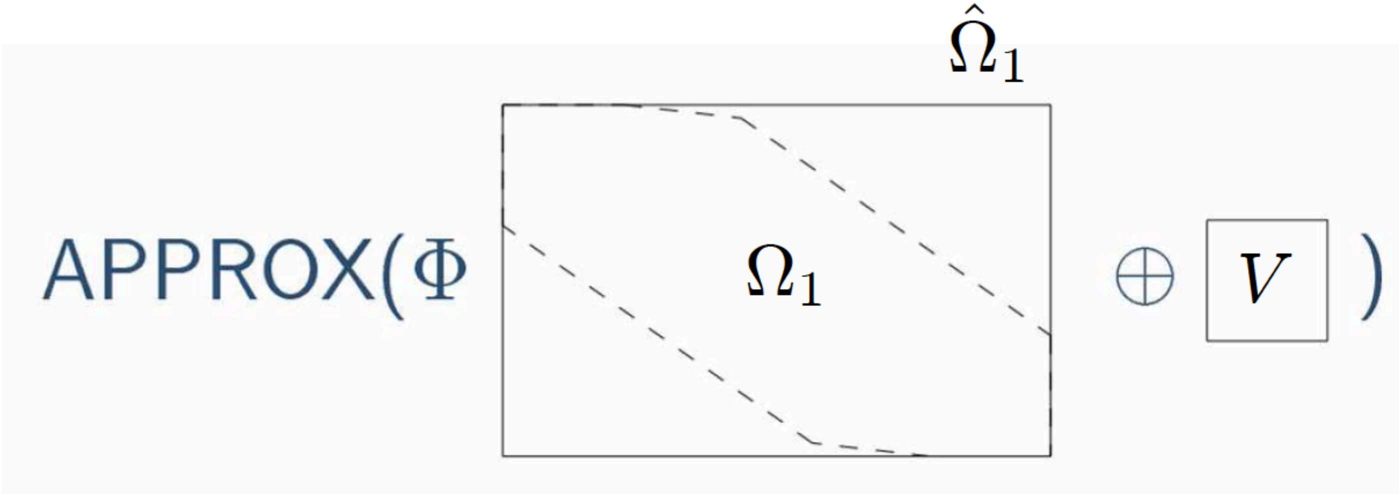
Wrapping Effect

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



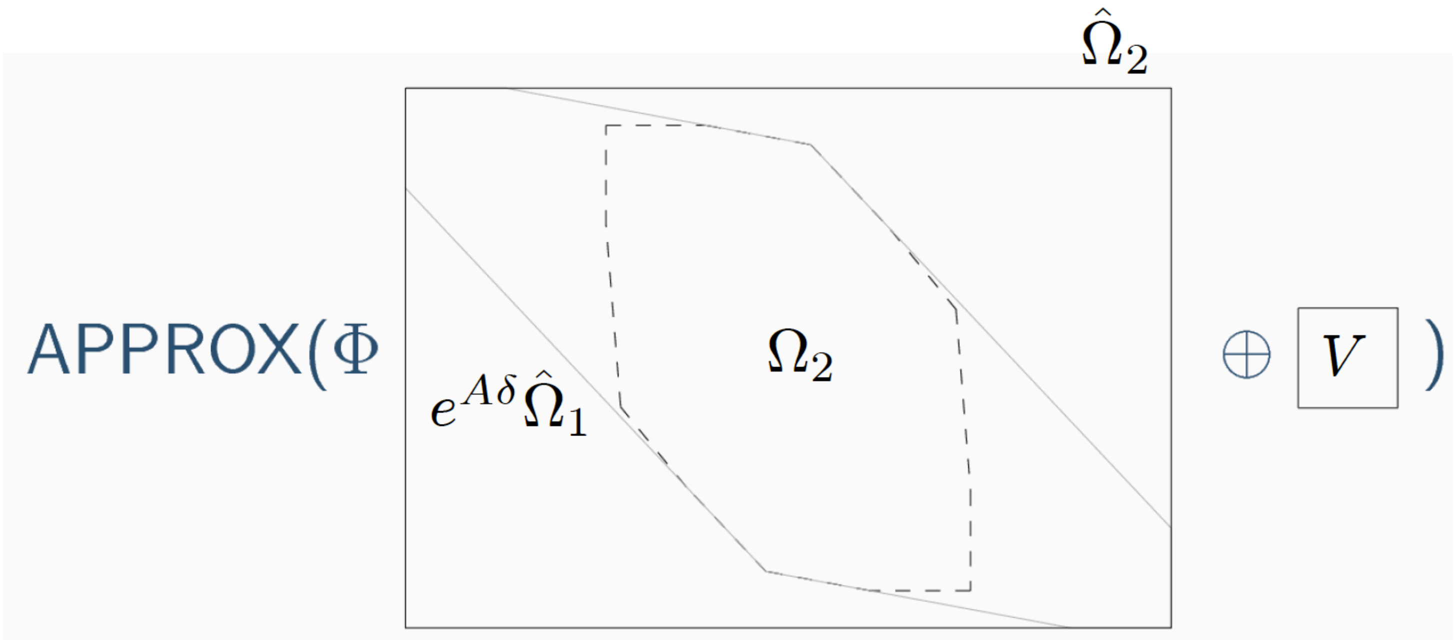
Wrapping Effect

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



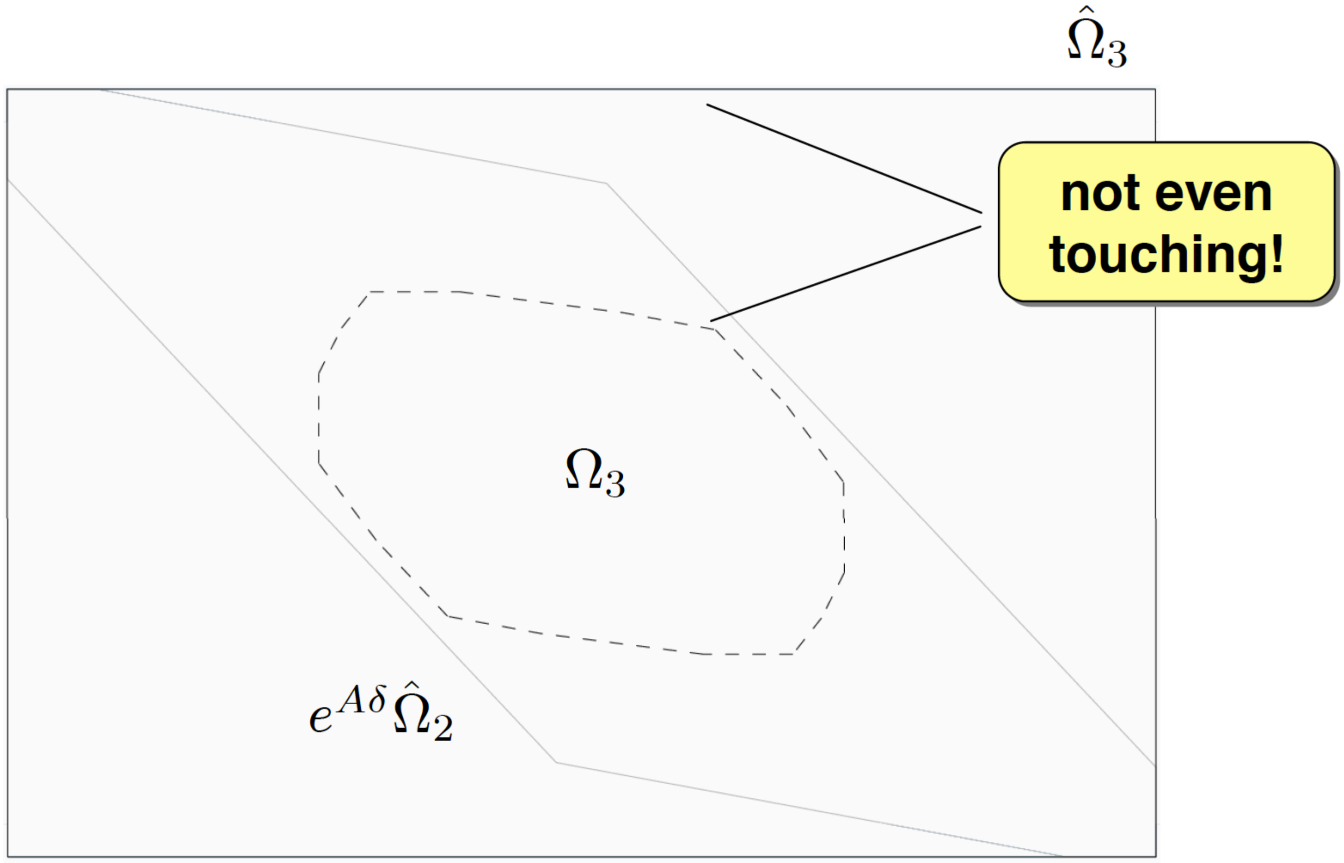
Wrapping Effect

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



Wrapping Effect

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



Fighting the Wrapping Effect

Separate transformations and Minkowski sum:

$$\Omega_{k+1} = \underbrace{e^{A(k+1)\delta}\Omega_0}_{R_{k+1}} \oplus \underbrace{\underbrace{e^{Ak\delta} V}_{V_k} \oplus \underbrace{(e^{A(k-1)\delta} V \oplus \cdots \oplus V)}_{S_k}}_{S_{k+1}}$$

Use four sequences:

$$R_0 = \Omega_0 \qquad V_0 = V \qquad S_0 = \{0\}$$

$$R_{k+1} = e^{A\delta} R_k \qquad V_{k+1} = e^{A\delta} V_k \qquad S_{k+1} = S_k \oplus V_k \qquad \Omega_{k+1} = R_{k+1} \oplus S_{k+1}$$

Four-Sequence Algorithm

$R_{k+1} = e^{A\delta} R_k \quad V_{k+1} = e^{A\delta} V_k \quad S_{k+1} = S_k \oplus V_k \quad \Omega_{k+1} = R_{k+1} \oplus S_{k+1}$

Transformations only in R_k and V_k:

- Complexity independent of k
- No overapproximation necessary

Minkowski sum only in S_k and Ω_k:

- Growing no of generators, but no longer transformed
- $O(Nn^3)$ instead of $O(N^2n^3)$

Four-Sequence Algorithm

$$R_{k+1} = e^{A\delta} R_k \quad V_{k+1} = e^{A\delta} V_k \quad \hat{S}_{k+1} = \hat{S}_k \oplus \mathrm{App}(V_k) \quad \Omega_{k+1} = R_{k+1} \oplus S_{k+1}$$

Use overapproximation with:

- $\mathrm{App}(X) \oplus \mathrm{App}(Y) = \mathrm{App}(X \oplus Y)$
- Bounding box, octogonal, etc.

No accumulation of error:

- $\hat{S}_k = \mathrm{App}(S_k)$
- $\hat{\Omega}_k \subseteq \mathrm{App}(\Omega_k)$

Fighting the Wrapping Effect

Exact versus overapproximation

- Dimension 5 for 600 time steps
- Overapproximation with bounding box