Modeling and Analysis of Hybrid Systems

Linear hybrid automata II: Approximation of reachable state sets

Prof. Dr. Erika Ábrahám

Informatik 2 - Theory of Hybrid Systems
RWTH Aachen University

SS 2015

We had a look at state set approximations by

- convex polyhedra,

and at the basic operations

- testing for membership,
- intersection, and
- union

on these.

Thus we can

- approximate state sets and
- compute with them.

How is all this used in the reachability analysis procedure?

General reachability procedure

Input: Set Init of initial states.
Output: Set R of reachable states.

Algorithm:

$$
\begin{array}{l}
R^{\text{new}} := \text{Init}; \\
R := \emptyset; \\
\text{while } (R^{\text{new}} \neq \emptyset)\{ \\
\qquad R \quad := R \cup R^{\text{new}}; \\
\qquad R^{\text{new}} := \boxed{\text{Reach}}(R^{\text{new}}) \backslash R; \\
\}
\end{array}
$$

What is "Reach"?

What is “Reach”?

For hybrid systems, independently of the exact definition of “Reach”, it will involve the following computations:

Given a state set R, compute

- the set of states reachable from R by a flow (i.e., time transisiton), and
- the set of states reachable from R by a jump (i.e., discrete transition).

Computing the jump successors of a set can be done with the operations we already introduced.

The harder part is computing the flow successors. So let’s have a look at that...

Approximating a flow pipe

Consider a dynamical system with state equation

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

Approximating a flow pipe

Consider a dynamical system with state equation

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

We assume f to be Lipschitz continuous.

Approximating a flow pipe

Consider a dynamical system with state equation

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

We assume f to be Lipschitz continuous.

Wikipedia: "Intuitively, a Lipschitz continuous function is limited in how fast it can change: for every pair of points on the graph of this function, the absolute value of the slope of the line connecting them is no greater than a definite real number [...]".

Approximating a flow pipe

Consider a dynamical system with state equation

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

We assume f to be Lipschitz continuous.

Wikipedia: "Intuitively, a Lipschitz continuous function is limited in how fast it can change: for every pair of points on the graph of this function, the absolute value of the slope of the line connecting them is no greater than a definite real number [...]."

Lipschitz continuity implies the existence and uniqueness of the solution to an initial value problem, i.e., for every initial state x_0 there is a unique solution $x(t, x_0)$ to the state equation.

Approximating a flow pipe

The set of reachable states at time t from a set of initial states X_0 is defined as

$$\mathcal{R}_t(X_0) = \{x_t \mid \exists x_0 \in X_0.\ x_t = x(t, x_0)\}.$$

Approximating a flow pipe

The set of reachable states at time t from a set of initial states X_0 is defined as

$$\mathcal{R}_t(X_0) = \{x_t \mid \exists x_0 \in X_0.\ x_t = x(t, x_0)\}.$$

The set of reachable states, the flow pipe, from X_0 in the time interval $[0, t_f]$ is defined as

$$\mathcal{R}_{[0,t_f]}(X_0) = \cup_{t\in[0,t_f]}\mathcal{R}_t(X_0).$$

Approximating a flow pipe

The set of reachable states at time t from a set of initial states X_0 is defined as

$$\mathcal{R}_t(X_0) = \{x_t \mid \exists x_0 \in X_0.\ x_t = x(t, x_0)\}.$$

The set of reachable states, the flow pipe, from X_0 in the time interval $[0, t_f]$ is defined as

$$\mathcal{R}_{[0,t_f]}(X_0) = \cup_{t\in[0,t_f]}\mathcal{R}_t(X_0).$$

We describe a solution which approximates the flow pipe by a sequence of convex polytopes.

Problem statement for polyhedral approximation of flow pipes

Given

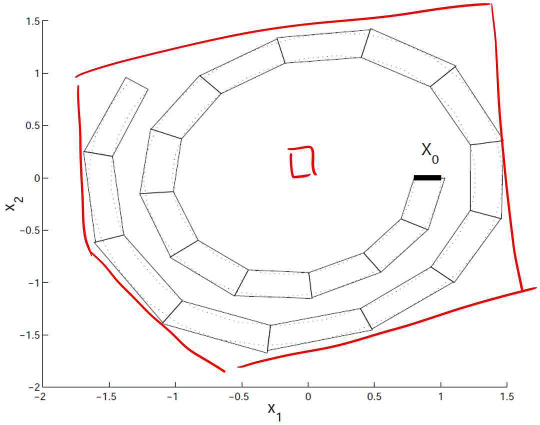
- a set X_0 of initial states which is a polytope, and
- a final time t_f,

compute a polyhedral approximation $\hat{\mathcal{R}}_{[0,t_f]}(X_0)$ to the flow pipe $\mathcal{R}_{[0,t_f]}(X_0)$ such that

$$\mathcal{R}_{[0,t_f]}(X_0) \subseteq \hat{\mathcal{R}}_{[0,t_f]}(X_0).$$

Flow pipe segmentation

Since a single convex polyhedron would strongly overapproximate the flow pipe, we compute a sequence of convex polyhedra, each approximating a flow pipe segment.



Segmented flow pipe approximation

Let the time interval $[0, t_f]$ be divided into $0 < N \in \mathbb{N}$ time segments

$$[0, t_1],\ [t_1, t_2],\ \ldots,\ [t_{N-1}, t_f]$$

with $t_i = i \cdot \frac{t_f}{N}$.

Segmented flow pipe approximation

Let the time interval $[0, t_f]$ be divided into $0 < N \in \mathbb{N}$ time segments

$$[0, t_1],\ [t_1, t_2],\ \ldots,\ [t_{N-1}, t_f]$$

with $t_i = i \cdot \frac{t_f}{N}$.

We generate an approximation $\hat{\mathcal{R}}_{[t_1,t_2]}(X_0)$ for each flow pipe segment:

$$\mathcal{R}_{[t_1,t_2]}(X_0) \subseteq \hat{\mathcal{R}}_{[t_1,t_2]}(X_0).$$

Segmented flow pipe approximation

Let the time interval $[0, t_f]$ be divided into $0 < N \in \mathbb{N}$ time segments

$$[0, t_1],\ [t_1, t_2],\ \ldots,\ [t_{N-1}, t_f]$$

with $t_i = i \cdot \frac{t_f}{N}$.

We generate an approximation $\hat{\mathcal{R}}_{[t_1,t_2]}(X_0)$ for each flow pipe segment:

$$\mathcal{R}_{[t_1,t_2]}(X_0) \subseteq \hat{\mathcal{R}}_{[t_1,t_2]}(X_0).$$

The complete flow pipe approximation is the union of the approximation of all N pipe segments:

$$\mathcal{R}_{[0,t_f]}(X_0) \subseteq \hat{\mathcal{R}}_{[0,t_f]}(X_0) = \bigcup_{k=1,\ldots,N} \hat{\mathcal{R}}_{[t_{k-1},t_k]}(X_0)$$

Approaches

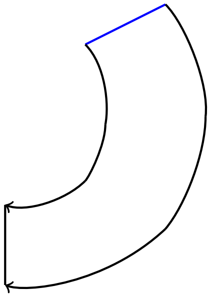
Next we discuss two possible approaches for flow pipe approximation, but there are different other techniques, too.

The first approach

Linear hybrid automata II: Time evolution

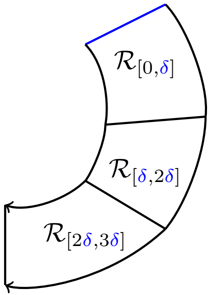
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$

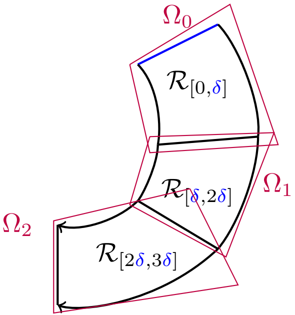


Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$



- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$



Linear hybrid automata II: Time evolution

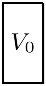
- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:

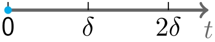
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:

Linear hybrid automata II: Time evolution

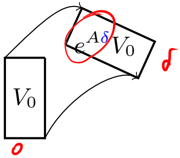
- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:





Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:

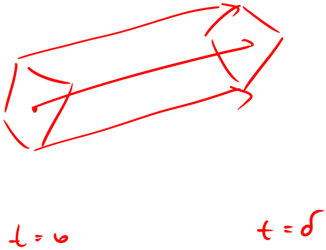


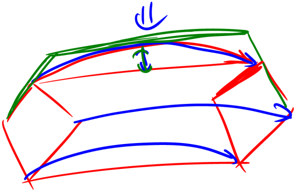
Ann

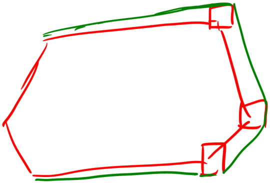
$\dot{x} = const$

$\dot{x} = Ax + Bu$

$u \in Box$

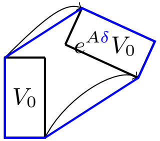


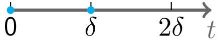




Linear hybrid automata II: Time evolution

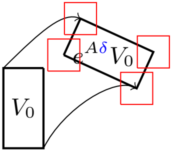
- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:





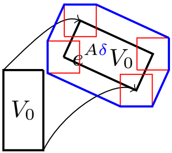
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:



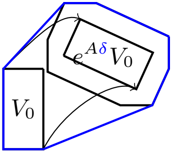
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:



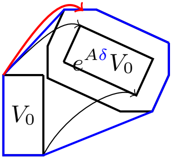
Linear hybrid automata II: Time evolution

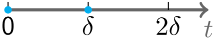
- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:



Linear hybrid automata II: Time evolution

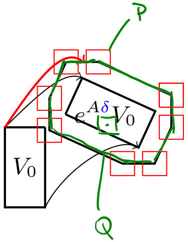
- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:





Linear hybrid automata II: Time evolution

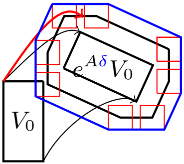
- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:



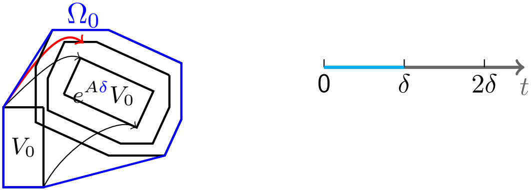
$Q \oplus P$

Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:

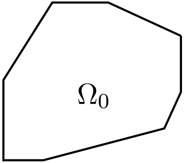


- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:



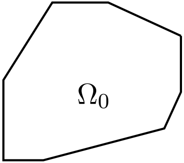
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:



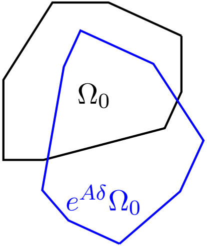
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:
- The remaining ones:



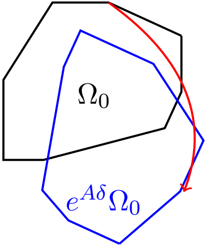
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:
- The remaining ones:



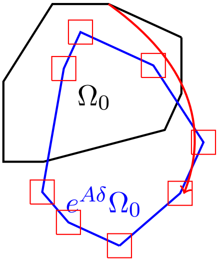
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:
- The remaining ones:



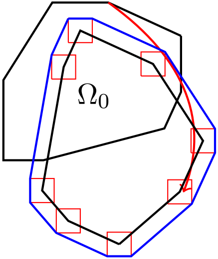
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:
- The remaining ones:



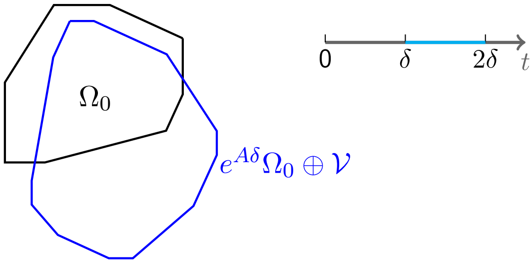
Linear hybrid automata II: Time evolution

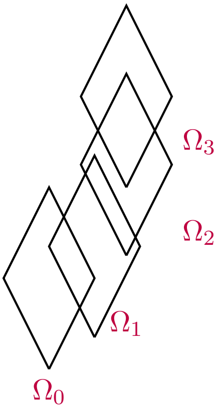
- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:
- The remaining ones:



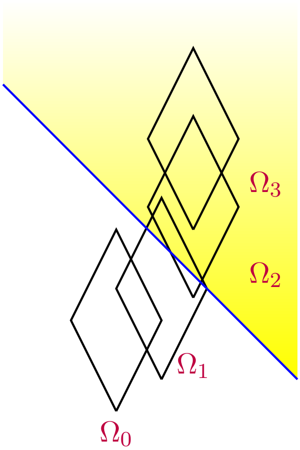
Linear hybrid automata II: Time evolution

- Assume $\dot{x} = Ax + Bu$
- Compute $\Omega_0, \Omega_1, \ldots$ such that $\mathcal{R}_{[i\delta,(i+1)\delta]} \subseteq \Omega_i$
- The first flowpipe segment:
- The remaining ones:

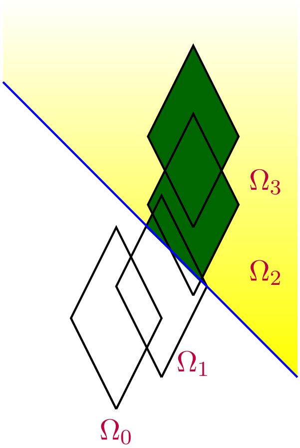



Ω_3
Ω_2
Ω_1
Ω_0

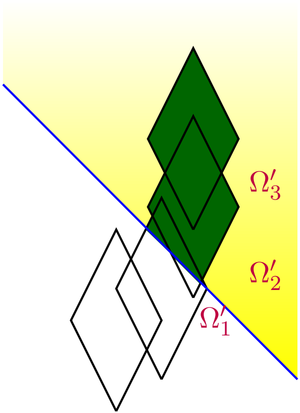
Linear hybrid automata II: Discrete steps (jumps)

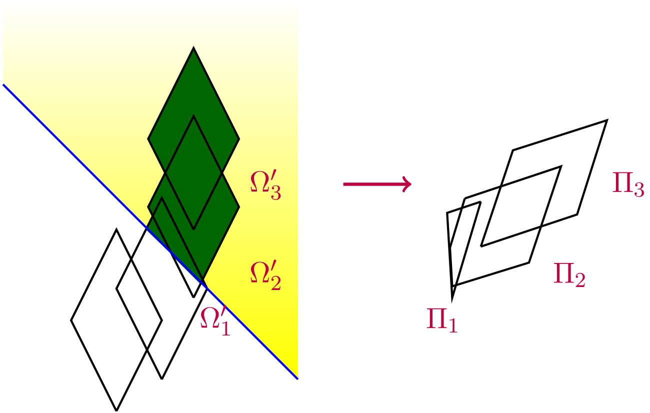


Linear hybrid automata II: Discrete steps (jumps)

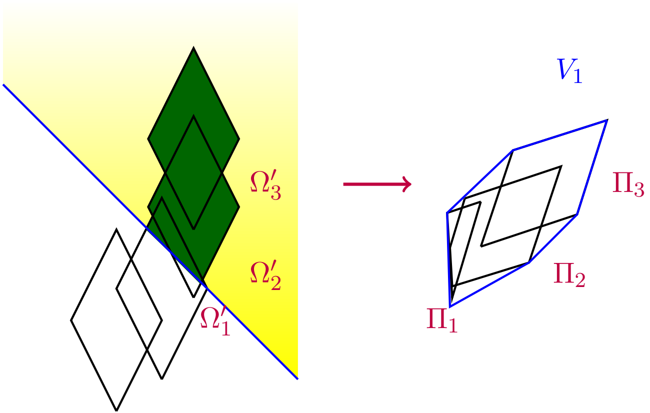


Linear hybrid automata II: Discrete steps (jumps)




Ω'_3
Ω'_2
Ω'_1
Π_3
Π_2
Π_1

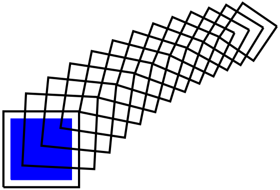
Linear hybrid automata II: Discrete steps (jumps)



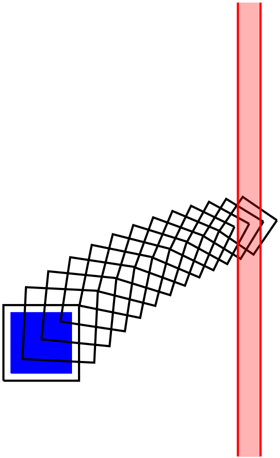
Linear hybrid automata II: The global picture

Linear hybrid automata II: The global picture

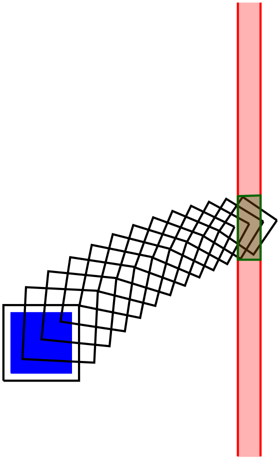
Linear hybrid automata II: The global picture



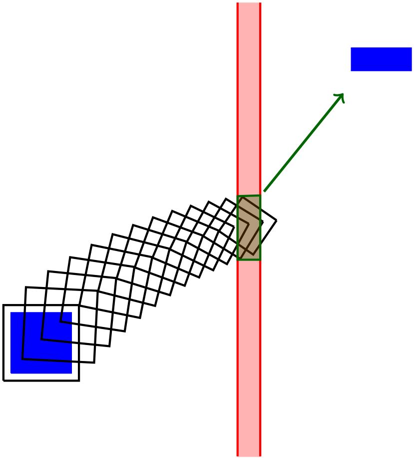
Linear hybrid automata II: The global picture



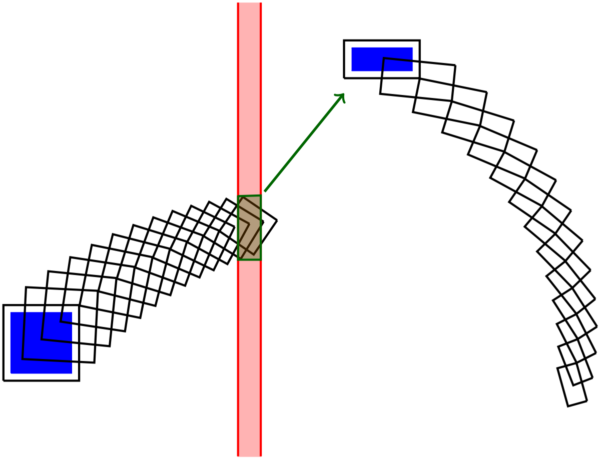
Linear hybrid automata II: The global picture



Linear hybrid automata II: The global picture



Linear hybrid automata II: The global picture



The second approach

Literatur

Alongkrit Chutinan and Bruce H. Krogh:
Computing Polyhedral Approximations to Flow Pipes for Dynamic Systems
In Proceedings of the 37rd IEEE Conference on Decision and Control, 1998

Olaf Stursberg and Bruce H. Krogh:
Efficient Representation and Computation of Reachable Sets for Hybrid Systems
Hybrid Systems: Computation and Control, LNCS 2623, pp. 482-497, 2003

Some notations

We will use the following notations:

- Let $POLY(C, d)$ denote the convex polytope defined by the pair $(C, d) \in \mathbb{R}^{m \times n} \times \mathbb{R}^m$ according to

$$POLY(C, d) = \{x \mid Cx \leq d\}.$$

- For a polytope P by $V(P)$ we denote the finite set of its vertices, which are points in P that cannot be written as a strict convex combination of any other two points in P.
- Given a finite set of points Γ, the convex hull $conv(\Gamma)$ of Γ is the smallest convex set that contains Γ.

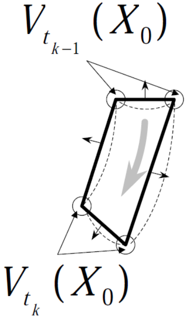
Approximation of a flow pipe segment

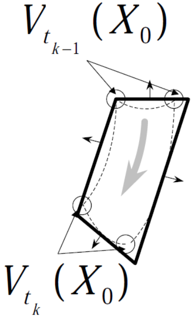
The approximation of the flow pipe for the time segment $[t_{k-1}, t_k]$ ($k \in \{1, \ldots, N\}$) consists of the following steps:

Approximation of a flow pipe segment

The approximation of the flow pipe for the time segment $[t_{k-1}, t_k]$ $(k \in \{1, \ldots, N\})$ consists of the following steps:

- Evolve vertices: Compute the set of points reachable from the vertices of X_0 in time t_{i-1} and in time t_i.

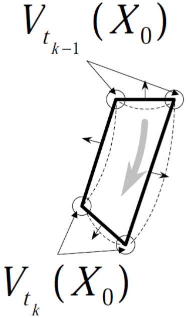


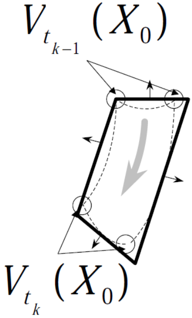


Approximation of a flow pipe segment

The approximation of the flow pipe for the time segment $[t_{k-1}, t_k]$ $(k \in \{1, \ldots, N\})$ consists of the following steps:

- Evolve vertices: Compute the set of points reachable from the vertices of X_0 in time t_{i-1} and in time t_i.
- Determine hull: Compute the convex hull of those points.

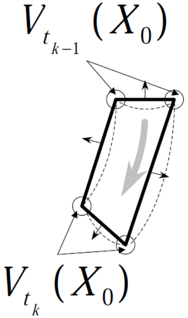


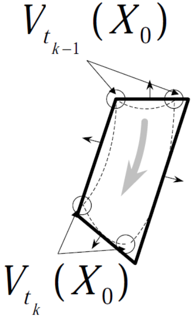


Approximation of a flow pipe segment

The approximation of the flow pipe for the time segment $[t_{k-1}, t_k]$ $(k \in \{1, \ldots, N\})$ consists of the following steps:

- Evolve vertices: Compute the set of points reachable from the vertices of X_0 in time t_{i-1} and in time t_i.
- Determine hull: Compute the convex hull of those points.
- Bloat hull: Enlarge the hull until it contains all points of the flow pipe segment.





1. Evolve vertices

To gain some geometrical information about the flow pipe segment, we begin with taking sample points at times t_{k-1} and t_k from the trajectories emanating from the vertices of X_0.

1. Evolve vertices

To gain some geometrical information about the flow pipe segment, we begin with taking sample points at times t_{k-1} and t_k from the trajectories emanating from the vertices of X_0.

In particular, we compute the sets $V_{t_{k-1}}(X_0)$ and $V_{t_k}(X_0)$ where

$$V_t(X_0) = \{x(t, v) \mid v \in V(X_0)\}.$$

1. Evolve vertices

To gain some geometrical information about the flow pipe segment, we begin with taking sample points at times t_{k-1} and t_k from the trajectories emanating from the vertices of X_0.

In particular, we compute the sets $V_{t_{k-1}}(X_0)$ and $V_{t_k}(X_0)$ where

$$V_t(X_0) = \{x(t, v) \mid v \in V(X_0)\}.$$

Each point in the above sets can be obtained

- by analytic solution of the state equation and computing the value, or
- by simulation.

2. Determine hull

We use the evolved vertices in $V_{t_{k-1}}(X_0)$ and $V_{t_k}(X_0)$ to form a convex hull which serves as an initial approximation to the flow pipe segment $\mathcal{R}_{[t_{k-1},t_k]}(X_0)$, denoted by

$$\Phi_{[t_{k-1},t_k]}(X_0) = conv(V_{t_{k-1}}(X_0) \cup V_{t_k}(X_0)).$$

2. Determine hull

We use the evolved vertices in $V_{t_{k-1}}(X_0)$ and $V_{t_k}(X_0)$ to form a convex hull which serves as an initial approximation to the flow pipe segment $\mathcal{R}_{[t_{k-1},t_k]}(X_0)$, denoted by

$$\Phi_{[t_{k-1},t_k]}(X_0) = conv(V_{t_{k-1}}(X_0) \cup V_{t_k}(X_0)).$$

Note that $\Phi_{[t_{k-1},t_k]}(X_0)$ may not contain the whole flow pipe segment $\mathcal{R}_{[t_{k-1},t_k]}(X_0)$.

2. Determine hull

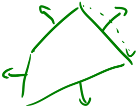
We use the evolved vertices in $V_{t_{k-1}}(X_0)$ and $V_{t_k}(X_0)$ to form a convex hull which serves as an initial approximation to the flow pipe segment $\mathcal{R}_{[t_{k-1},t_k]}(X_0)$, denoted by

$$\Phi_{[t_{k-1},t_k]}(X_0) = conv(V_{t_{k-1}}(X_0) \cup V_{t_k}(X_0)).$$

Note that $\Phi_{[t_{k-1},t_k]}(X_0)$ may not contain the whole flow pipe segment $\mathcal{R}_{[t_{k-1},t_k]}(X_0)$.

Let (C_Φ, d_Φ) be the matrix-vector pair defining the convex hull, i.e.,

$$\Phi_{[t_{k-1},t_k]}(X_0) = POLY(C_\Phi, d_\Phi).$$



3. Bloat hull

- The normal vector on each face of the polytope points outward.

3. Bloat hull

- The normal vector on each face of the polytope points outward.
- We use the normal vectors to the faces of this convex hull as a set of direction vectors to bloat the convex set until it contains the whole flow pipe segment.

3. Bloat hull

- The normal vector on each face of the polytope points outward.
- We use the normal vectors to the faces of this convex hull as a set of direction vectors to bloat the convex set until it contains the whole flow pipe segment.
- Given: $POLY(C_\Phi, d_\Phi)$.

3. Bloat hull

- The normal vector on each face of the polytope points outward.
- We use the normal vectors to the faces of this convex hull as a set of direction vectors to bloat the convex set until it contains the whole flow pipe segment.
- Given: $POLY(C_\Phi, d_\Phi)$.
- We want: $\mathcal{R}_{[t_{k-1},t_k]}(X_0) \subseteq POLY(C_\Phi, \boxed{\text{d}})$.

3. Bloat hull

- We compute d as the solution to the following optimization problem:

$$\begin{aligned} \min_{d} \quad & volume[POLY(C_\Phi, d)] \\ s.t. \quad & \mathcal{R}_{[t_{k-1}, t_k]}(X_0) \subseteq POLY(C_\Phi, d). \end{aligned} \tag{1}$$

3. Bloat hull

- We compute d as the solution to the following optimization problem:

$$\begin{aligned} \min_{d} \quad & volume[POLY(C_\Phi, d)] \\ s.t. \quad & \mathcal{R}_{[t_{k-1},t_k]}(X_0) \subseteq POLY(C_\Phi, d). \end{aligned} \tag{1}$$

- The ith component d_i^* of the optimum d^* can be found by solving

$$\max_{x} \; c_i^T x \qquad s.t. \; x \in \mathcal{R}_{[t_{k-1},t_k]}(X_0). \tag{2}$$

3. Bloat hull

- We compute d as the solution to the following optimization problem:

$$\begin{aligned} \min_{d} \quad & volume[POLY(C_\Phi, d)] \\ s.t. \quad & \mathcal{R}_{[t_{k-1}, t_k]}(X_0) \subseteq POLY(C_\Phi, d). \end{aligned} \tag{1}$$

- The ith component d_i^* of the optimum d^* can be found by solving

$$\max_{x} \; c_i^T x \qquad s.t. \; x \in \mathcal{R}_{[t_{k-1}, t_k]}(X_0). \tag{2}$$

- or, equivalently,

$$\max_{x_0, t} \; c_i^T x(t, x_0) \qquad s.t. \; x_0 \in X_0, \; t \in [t_{k-1}, t_k]. \tag{3}$$

- We compute d as the solution to the following optimization problem:

$$\begin{aligned} \min_{d} \quad & volume[POLY(C_\Phi, d)] \\ s.t. \quad & \mathcal{R}_{[t_{k-1},t_k]}(X_0) \subseteq POLY(C_\Phi, d). \end{aligned} \tag{1}$$

- The ith component d_i^* of the optimum d^* can be found by solving

$$\max_{x} \; c_i^T x \qquad s.t. \; x \in \mathcal{R}_{[t_{k-1},t_k]}(X_0). \tag{2}$$

- or, equivalently,

$$\max_{x_0,t} \; c_i^T x(t, x_0) \qquad s.t. \; x_0 \in X_0, \; t \in [t_{k-1}, t_k]. \tag{3}$$

- Solution (x_0^*, t^*) to 3 $\rightarrow$

  Solution $x(t^*, x_0^*)$ to 2 $\rightarrow$

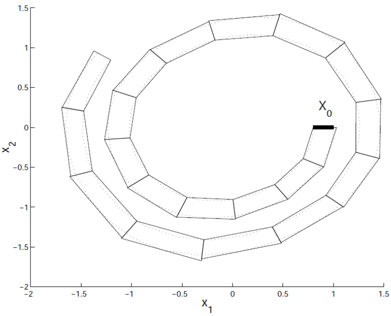
  Solution $d_i^* = c_i^T x(t^*, x_0^*)$ to 1.

Example

- Van der Pol equation:

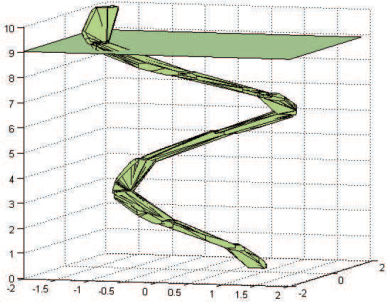
$$\begin{aligned}\dot{x}_1 &= x_2\\ \dot{x}_2 &= -0.2(x_1^2-1)x_2 - x_1.\end{aligned}$$

- Intial set: $X_0 = \{(x_1, x_2) \mid 0.8 \le x_1 \le 1 \wedge x_2 = 0\}$.
- Time: $t_f = 10$.
- Segments: 20

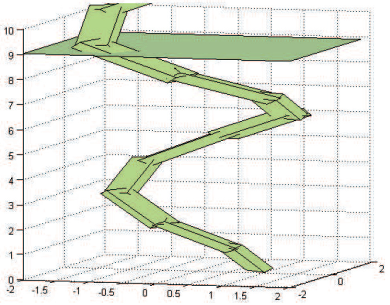


- Van der Pol equation with a third variable being a clock.
- Approximation

with convex polyhedra and



with oriented rectangular hull:



Partitioning the initial set

Var der Pol system with initial set $X_0 = \{(x_1, x_2) \mid 5 \leq x_1 \leq 45 \wedge x_2 = 0\}$.