

# Flowpipe Approximation and Clustering in Space-Time

Goran Frehse, Rajat Kateja  
Université Grenoble 1 Joseph Fourier - Verimag  
Centre Equation, 2 av. de Vignate  
38610 Gières, France  
goran.frehse@imag.fr

Colas Le Guernic  
DGA-Maîtrise de l'Information  
La Roche Marguerite route de Laillé BP 7  
35998 Rennes, France  
colas.le.guernic@gmail.com

## ABSTRACT

In this paper, we present an approximation of the set of reachable states, called flowpipe, for a continuous system with affine dynamics. Our approach is based on a representation we call flowpipe sampling, which consists of a set of continuous, interval-valued functions over time. A flowpipe sampling attributes to each time point a polyhedral enclosure of the set of states reachable at that time point, and is capable of representing a nonconvex enclosure of a nonconvex flowpipe. The use of flowpipe samplings allows us to represent and approximate the nonconvex flowpipe efficiently. In particular, we can measure the error incurred by the initial approximation and by further processing such as simplification and convexification. A flowpipe sampling can be efficiently translated into a set of convex polyhedra in a way that minimizes the number of convex sets for a given error bound. When applying flowpipe approximation for the reachability of hybrid systems, a reduction in the number of convex sets spawned by each image computation can lead to drastic performance improvements.

## Categories and Subject Descriptors

G.1.7 [Numerical Analysis]: Ordinary Differential Equations—*Initial value problems*

## Keywords

Hybrid systems, verification, reachability, tools

## 1. INTRODUCTION

A widely used strategy for computing the reachable states of a continuous or hybrid system is to cover the flowpipes (bundles of trajectories in the state space) with a finite but frequently large number of convex sets, which can be represented, e.g., as polyhedra, zonotopes, ellipsoids and support functions [1, 2, 7, 8, 9, 4]. Each of the convex sets covers the flowpipe on a certain time interval, and the approximation error usually increases rapidly with the size of this interval.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HSCC'13, April 8–11, 2013, Philadelphia, Pennsylvania, USA.  
Copyright 2013 ACM 978-1-4503-1567-8/13/04 ...\$15.00.

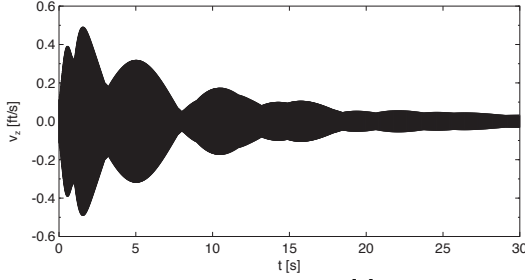
Often, the time step has to be made very small to achieve a desired accuracy. This in turn may lead to a very large number of convex sets that, depending on the processing or further image computation to be performed, can quickly become prohibitive. For reachability analysis of hybrid systems in particular, we have often observed a fatal explosion in the number of sets when more than a few of the convex sets can take a discrete transition, as each will spawn a new flowpipe in the next state, and so on.

The goal of this paper is to address this fundamental problem from two angles: Firstly, we use a representation, which we call flowpipe sampling, that consists of a set of continuous, interval-valued functions over time. A flowpipe sampling attributes to each time point a polyhedral enclosure of the set of states reachable at that time point, thus capable of representing a nonconvex enclosure of a nonconvex flowpipe. This representation helps to decouple, as far as possible, the accuracy from the number of convex sets created in the end. Secondly, we propose a clustering procedure that aims to minimize the number of convex sets produced for a desired accuracy and does so using accuracy bounds established by the flowpipe construction. A-posteriori error measurements help evaluate the distance of the approximation to the actual flowpipe.

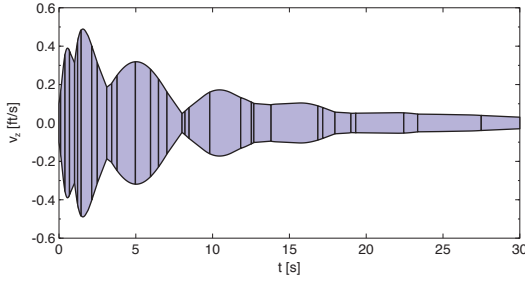
The following examples shall illustrate different aspects of the flowpipe approximation problem, as well as showcase the performance of our proposed solution.

**EXAMPLE 1.1 (HELICOPTER).** *Figure 1(a) show a flowpipe approximation for an affine helicopter model with 28 state variables plus a clock variable. It was obtained using the approach in [4], which constructs for each time-step a convex polyhedron in the 29-dimensional state space. The facet normals of the polyhedra, also called template directions, are given by the user. In this case, the axis directions are used so that the polyhedra are boxes. The complex dynamics of the system require using a very small time step throughout the time horizon of 30 s. Note that only the projection on two of the 29 variables is shown (the vertical speed and the clock), while the approximation takes the variation of all variables into account. As a result of the small time step, 1440 convex sets are constructed for a given directional error estimate of  $\epsilon = 0.025$ . The construction itself is computationally cheap at 5.9 s CPU time, but the sheer number of sets makes further processing and image computation impractical.*

*The approach proposed in this paper combines an enhanced flowpipe approximation with adaptive clustering that guarantees a conservative error bound on the directional distance to*



(a) The flowpipe approximation from [4] constructs one convex polyhedra per time-step, in total 1440 polyhedra



(b) The flowpipe approximation with clustering proposed in this paper constructs 32 convex polyhedra

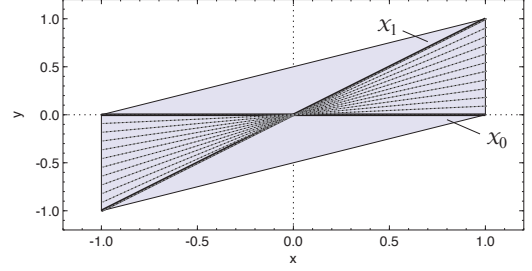
**Figure 1: Flowpipe approximations for the 28-dimensional affine helicopter model from Ex. 1.1, plus a clock. The shown sets are projections of 29-dimensional polyhedra onto two variables, the vertical speed and the clock**

the actual reachable set at each point in time. The time step is adapted separately for each of the template directions and can therefore be considerably larger. In the directions corresponding to the axis of the clock the system evolves linearly, so the time step spans the entire time horizon. The clustering step produces the 32 polyhedra shown in Fig. 1(b) for a given directional error bound of  $\epsilon = 0.025$ . The construction of the flowpipe sampling takes 9.4 s and the clustering and outer polyhedral approximation 4.8 s.

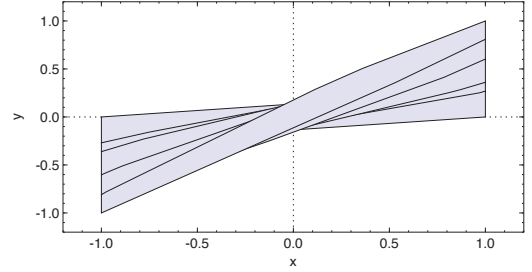
As the following example illustrates, the convexification error is by no means restricted to complex dynamics.

**EXAMPLE 1.2 (HOURGLASS).** Consider the simple linear ODE system  $\dot{x} = 0$ ,  $\dot{y} = x$ , with initial states  $\mathcal{X}_0 = \{-1 \leq x(0) \leq 1, y(0) = 0\}$  as shown in Fig. 2(a). We consider the states reachable up to time  $t = 1$ . The reachable set is pointwise convex in time, but every convex set that covers the reachable states over a nonsingular time interval is forcibly an overapproximation. The time-step adaptation of the LGG-algorithm in [4] decreases the time step until all error terms fall below the desired threshold. In this case, the error terms are zero since the trajectories are linear, i.e.,  $x(t) = x(0)$ ,  $y(t) = t \cdot x(0)$ . The LGG-algorithm therefore covers the whole flowpipe in a single timestep. In the best case (arbitrarily well-chosen template directions), the result is the convex hull of the flowpipe, shown in Fig. 2(a).

The flowpipe approximation proposed in this paper can produce a result of arbitrary accuracy in terms of a directional error that is measured in the template directions. As more template directions are added, this directional error converges towards the Hausdorff distance between the ac-



(a) The initial set  $\mathcal{X}_0$ , the final set  $\mathcal{X}_1$ , and the smallest convex approximation of the reachable set  $\mathcal{X}_{[0,1]}$  (shaded)



(b) An approximation of  $\mathcal{X}_{[0,t]}$  constructed by the proposed technique for a given error bound 0.1

**Figure 2: Example 1.2 has a nonconvex reachable set in the shape of an hourglass. It is pointwise convex in time, but every convex set covering a nonsingular interval is forcibly an overapproximation. All trajectories are linear, so the approximation error is impossible to detect based on the dynamics alone**

tual reachable states and the overapproximation. Figure 2(b) shows the result for a given error bound 0.1 in 64 uniformly distributed template directions.

The basis of our approach is the representation of a convex set by its support function. Simply put, given the normal vector of a halfspace (direction), the support function tells the position where the halfspace touches the set such that it contains it. If the dynamics are affine and the set of initial states is convex, the states reachable at a given point in time is also convex and can be represented by their support function. The flowpipe can therefore be described by a family of support functions parameterized by time, which has been considered in, e.g., [13, 10]. Our flowpipe sampling builds on this concept, which we refine by accounting for the fact that we can only compute a finite number of values in terms of both direction and time points. There are methods to approximate flowpipes directly with sets that are nonconvex, e.g., with polynomial tubes [12].

The flowpipe construction in this paper builds heavily on previous work in [9, 4]. Notably, this paper provides error measurements that include the convexification error. For more detailed comments, see Sect. 2.4. The clustering approach of Sect. 3 is, to the best of our knowledge, novel. For lack of space, we have omitted several proofs that can be found in [5]. The SpaceEx tool and examples from the paper are available at the SpaceEx website [3].

In the next section, we define flowpipe samplings as a general means to describe and approximate flowpipes and then present our flowpipe approximation algorithm. We also show how a flowpipe sampling can be translated into a set

of convex polyhedra. In Sect. 3, we present our clustering approach, which minimizes the number convex polyhedra that a flowpipe sampling defines. Section 4 presents some experimental results.

## 2. FLOWPIPE APPROXIMATION IN SPACE-TIME

We consider continuous dynamical systems given by differential equations of the form

$$\dot{x}(t) = Ax(t) + u(t), \quad u(t) \in \mathcal{U}, \quad (1)$$

where  $x(t) \in \mathbb{R}^n$  and  $A \in \mathbb{R}^n \times \mathbb{R}^n$ . The closed, bounded, and convex set  $\mathcal{U} \subseteq \mathbb{R}^n$  can represent, e.g., nondeterministic inputs, disturbances or approximation errors. The initial states of the system are given as a convex compact set  $\mathcal{X}_0$ , i.e.,  $x(0) \in \mathcal{X}_0$ . We refer to the states reachable from  $\mathcal{X}_0$  by time elapse as the *flowpipe* of  $\mathcal{X}_0$ . We compute a sequence of closed and bounded convex sets  $\Omega_0, \dots, \Omega_N$  that covers the flowpipe using an extension of the approximation technique in [4]. In our construction, each  $\Omega_i$  is the result of convex hull and Minkowski sum operations on polytopes. So  $\Omega_i$  is itself a polytope, but explicitly computing it would be prohibitively expensive in higher dimensions. Its support function can, however, be computed efficiently for any given direction. In the next section we present how we approximate convex sets by computing their support function, and in the following section we extend the approach to flowpipes.

### 2.1 Approximating Convex Sets with Support Samples

A convex set can be represented by its support function, which attributes to each direction in  $\mathbb{R}^n$  the signed distance of the farthest point of the set to the origin. Computing the value of the support function for a given set of directions, one obtains a polyhedron that overapproximates the set. We call this *sampling the support function*. In this paper, we allow this computation to be approximative, i.e., a lower and an upper bound on the support function is computed. In this section, we recall some basics, define support samples and derive error bounds for approximations from support samples.

We use the following standard notation for operations on sets. Let  $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$  be sets,  $\lambda \in \mathbb{R}$ , and  $M$  be an  $m \times n$  matrix of reals. We denote  $\lambda\mathcal{X} = \{\lambda x \mid x \in \mathcal{X}\}$ ,  $M\mathcal{X} = \{Mx \mid x \in \mathcal{X}\}$ , and  $\mathcal{X} \oplus \mathcal{Y} = \{x + y \mid x \in \mathcal{X}, y \in \mathcal{Y}\}$  (Minkowski sum). A *halfspace*  $\mathcal{H} \subseteq \mathbb{R}^n$  is the set of points satisfying a linear constraint

$$\mathcal{H} = \{x \mid a^\top x \leq b\},$$

where  $a = (a_1 \dots a_n) \in \mathbb{R}^n$  and  $b \in \mathbb{R}$ . A *polyhedron*  $\mathcal{P} \subseteq \mathbb{R}^n$  is the intersection of a finite number of half spaces

$$\mathcal{P} = \left\{x \mid \bigwedge_{i=1}^m a_i^\top x \leq b_i\right\},$$

where  $a_i \in \mathbb{R}^n$  and  $b_i \in \mathbb{R}$ . If such  $a_i$  and  $b_i$  are known, we say the polyhedron is given in *constraint representation*. A *polytope* is a polyhedron that is bounded. The *convex hull*  $\text{CH}(\mathcal{X}) \subseteq \mathbb{R}^n$  of a set  $\mathcal{X}$  is

$$\text{CH}(\mathcal{X}) = \left\{\sum_{i=1}^{n+1} \lambda_i v_i \mid v_i \in \mathcal{X}, \lambda_i \in \mathbb{R}^{\geq 0}, \sum_{i=1}^{n+1} \lambda_i = 1\right\}.$$

The *support function* of a nonempty, closed and bounded continuous set  $\mathcal{X} \subseteq \mathbb{R}^n$  with respect to a direction vector  $\ell \in \mathbb{R}^n$  is

$$\rho_{\mathcal{X}}(\ell) = \max\{\ell^\top x \mid x \in \mathcal{X}\}.$$

The set of *support vectors* (or *maximizers*) of  $\mathcal{X}$  in direction  $\ell$  is denoted by

$$\sigma_{\mathcal{X}}(\ell) = \{x^* \in \mathcal{X} \mid \ell^\top x^* = \rho_{\mathcal{X}}(\ell)\}.$$

We are interested in support functions because many set operations are computationally cheaper to carry out on support functions than on, say, polyhedra [9]. For instance, the operations  $M\mathcal{X}$ ,  $\mathcal{X} \oplus \mathcal{Y}$ , and  $\text{CH}(\mathcal{X} \cup \mathcal{Y})$  can be very expensive for polyhedra in constraint representation, while for support functions they are simple:

$$\begin{aligned} \rho_{M\mathcal{X}}(\ell) &= \rho_{\mathcal{X}}(M^\top \ell), \\ \rho_{\mathcal{X} \oplus \mathcal{Y}}(\ell) &= \rho_{\mathcal{X}}(\ell) + \rho_{\mathcal{Y}}(\ell), \\ \rho_{\text{CH}(\mathcal{X} \cup \mathcal{Y})}(\ell) &= \max\{\rho_{\mathcal{X}}(\ell), \rho_{\mathcal{Y}}(\ell)\}. \end{aligned}$$

Because support functions are cheap, we would like to use them in our flowpipe approximation. However, in our construction it is not always possible or efficient to compute the exact value of the support function. Instead, we allow for interval bounds on the support function. Furthermore, we consider that those bounds are only computed for a finite number of directions. In the following, we examine how such bounds provide an outer approximation of the actual set and characterize the approximation error. Given a set  $\mathcal{X}$ , a *support sample*  $r = (\ell, [r^-, r^+])$  pairs a direction  $\ell \in \mathbb{R}^n$  with a real-valued interval  $[r^-, r^+]$  that contains the value of the support function of  $\mathcal{X}$ , i.e.,

$$\rho_{\mathcal{X}}(\ell) \in [r^-, r^+]. \quad (2)$$

A *support sampling* is a set of support samples

$$R = \{r_1, \dots, r_K\}, \quad \text{with } r_k = (\ell_k, [r_k^-, r_k^+]).$$

Its *outer approximation* is the polyhedron

$$[R] = \left\{x \mid \bigwedge_k \ell_k^\top x \leq r_k^+\right\}, \quad (3)$$

i.e., given a support sampling  $R$  of  $\mathcal{X}$ , we have that  $\mathcal{X} \subseteq [R]$ .

From the lower bounds in the support samples we can derive a lower bound on the support function in any direction, which allows us to bound the approximation error of the outer approximation. By definition, a support sample  $r_k$  implies that there is at least one point  $x \in \mathcal{X}$  such that  $\ell_k^\top x \geq r_k^-$ , as illustrated in Fig. 3. Let the *facet slab* of  $r_k$  be

$$[R]_k = [R] \cap \{\ell_k^\top x \geq r_k^-\}, \quad (4)$$

then the support function in direction  $\ell$  cannot be lower than

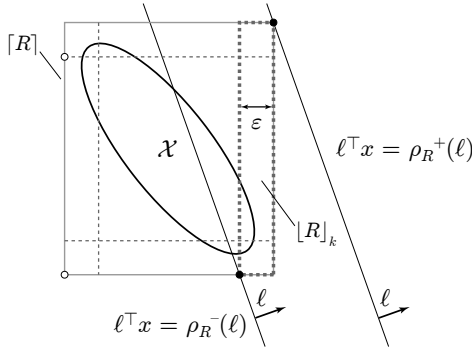
$$\min\{\ell^\top x \mid x \in [R]_k\} = -\rho_{[R]_k}(-\ell).$$

Combining the lower bounds from all facet slabs, we obtain the following result:

**LEMMA 2.1.** *Given a support sampling  $R$  of a nonempty compact convex set  $\mathcal{X}$ , the support function of  $\mathcal{X}$  is bounded in any direction  $\ell$  by  $\rho_R^-(\ell) \leq \rho_{\mathcal{X}}(\ell) \leq \rho_R^+(\ell)$ , where*

$$\rho_R^+(\ell) = \rho_{[R]}(\ell), \quad (5)$$

$$\rho_R^-(\ell) = \max_{k=1, \dots, K} -\rho_{[R]_k}(-\ell). \quad (6)$$



**Figure 3: A support sampling  $R$  of a set  $\mathcal{X}$  (solid black) defined over the axis directions, with lower and upper bound being  $\varepsilon$  apart. The outer approximation  $\lceil R \rceil$  (solid grey) is shown together with the facet slabs  $\lfloor R \rfloor_k$  (dashed), each of which contains at least one point of  $\mathcal{X}$ . For every direction  $\ell$ ,  $\lceil R \rceil$  provides an upper bound  $\rho_R^+(\ell)$  on the support function, and the facet slabs a lower bound  $\rho_R^-(\ell)$**

For a given direction  $\ell$ , the lower bound (6) can be reformulated as a linear program with  $O(Kn)$  variables and  $O(K^2)$  constraints by introducing an additional variable  $z$ :

$$\rho_R^-(\ell) = \min \left\{ z \in \mathbb{R} \mid \bigwedge_{k=1, \dots, K} z \geq \ell^\top x_k \wedge x_k \in \lfloor R \rfloor_k \right\}. \quad (7)$$

We consider two ways to measure the error between the actual set and its outer approximation: a directional error and the Hausdorff distance. The *directional error* of a support sampling  $R$  is the width of the bounds on the support function,

$$\varepsilon_R(\ell) = \rho_R^+(\ell) - \rho_R^-(\ell). \quad (8)$$

Let  $\mathcal{B}_k = \{x \mid \|x\|_k = 1\}$  be the unit ball in the  $k$ -norm. The *directed Hausdorff distance* between sets  $\mathcal{X}, \mathcal{Y}$  is

$$\vec{d}_H(\mathcal{X}, \mathcal{Y}) = \sup_{x \in \mathcal{X}} \inf_{y \in \mathcal{Y}} \|x - y\|_2 = \inf \{ \varepsilon > 0 \mid \mathcal{X} \subseteq \mathcal{Y} \oplus \varepsilon \mathcal{B}_2 \},$$

and the *Hausdorff distance* is

$$d_H(\mathcal{X}, \mathcal{Y}) = \max(\vec{d}_H(\mathcal{X}, \mathcal{Y}), \vec{d}_H(\mathcal{Y}, \mathcal{X})).$$

LEMMA 2.2. Given a support sampling  $R$  of  $\mathcal{X}$ ,

$$d_H(\mathcal{X}, \lceil R \rceil) \leq \max_{\|\ell\|_2=1} \varepsilon_R(\ell). \quad (9)$$

Using the LP formulation (7), the above bound on the Hausdorff distance can be rewritten as a quadratic maximization problem with bilinear constraints. This bound is generally quite costly to compute. But it implies that, as more and more directions are sampled, the largest directional error tends towards a bound on the Hausdorff distance (assuming directions are uniformly distributed).

## 2.2 Approximating Flowpipes with Support Samples over Time

Our space-time construction is a natural extension of the support function representation of sets. For a given convex and bounded set of initial states  $\mathcal{X}_0$ , we define the flowpipe as the states reachable from this set.

Formally, let  $\mathcal{X}_t$  be the states reachable from  $\mathcal{X}_0$  after exactly time  $t$ ,

$$\mathcal{X}_t = \{x(t) \mid x(0) \in \mathcal{X}_0, \forall \tau \in [0, t] \exists u(\tau) \in \mathcal{U} : \dot{x}(\tau) = Ax(\tau) + u(\tau)\}. \quad (10)$$

A *flowpipe segment* over a time interval  $[t_1, t_2]$  is the set

$$\mathcal{X}_{t_1, t_2} = \bigcup_{t_1 \leq t \leq t_2} \mathcal{X}_t.$$

In this paper, we assume a finite time horizon  $T$  and refer to  $\mathcal{X}_{0, T}$  as the *flowpipe*. Given that  $\mathcal{X}_0$  is convex and that the dynamics are affine,  $\mathcal{X}_t$  is convex at any time  $t$ . For a fixed value of  $t$ , we can approximate  $\mathcal{X}_t$  with a support sampling

$$R = \{(\ell_1, r_1), \dots, (\ell_K, r_K)\},$$

where the  $\ell_k$  given template directions, and the  $r_k$  are intervals containing the support function of  $\mathcal{X}_t$ . Recall that  $R$  allows us to construct an outer approximation of  $\mathcal{X}_t$  and quantify the approximation error.

We describe the nonconvex flowpipe over the time interval  $[0, T]$  in a similar way. Letting  $t$  vary in the time interval  $[0, T]$ , we consider the bounds of the interval  $r_k(t) = [r_k^-(t), r_k^+(t)]$  to be continuous functions over time. For every  $t$ ,  $r_k(t)$  contains the support function of  $\mathcal{X}_t$  in direction  $\ell_k(t)$ . A *flowpipe sampling* over  $K$  directions is a function  $F$  that attributes to each  $t$  a support sampling

$$F(t) = \{(\ell_1(t), r_1(t)), \dots, (\ell_K(t), r_K(t))\}. \quad (11)$$

The pairs  $(\ell_k(\cdot), r_k(\cdot))$  are called *flowpipe samples*. In this paper, we consider the directions to be constant over time, and simply write  $\ell_k$  instead of  $\ell_k(t)$ . By combining the outer approximation of the support sampling  $F(t)$  at each time point, we obtain an outer approximation of a flowpipe segment  $\mathcal{X}_{t_1, t_2}$ . With Lemma 2.2 it is straightforward to derive a bound on the Hausdorff distance between the flowpipe segment and its outer approximation.

LEMMA 2.3. Let  $F$  be a flowpipe sampling (11) and let

$$\lceil F \rceil_{t_1, t_2} = \bigcup_{t_1 \leq t \leq t_2} \lceil F(t) \rceil, \quad (12)$$

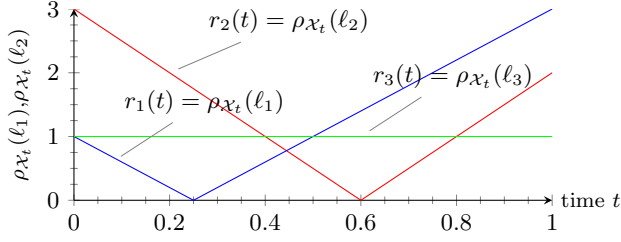
$$\varepsilon_{t_1, t_2} = \max_{t_1 \leq t \leq t_2} \max_{\|\ell\|_2=1} \varepsilon_{F(t)}(\ell). \quad (13)$$

Then  $\mathcal{X}_{t_1, t_2} \subseteq \lceil F \rceil_{t_1, t_2}$  and the Hausdorff distance between  $\lceil F \rceil_{t_1, t_2}$  and  $\mathcal{X}_{t_1, t_2}$  is bounded by  $\varepsilon_{t_1, t_2}$ .

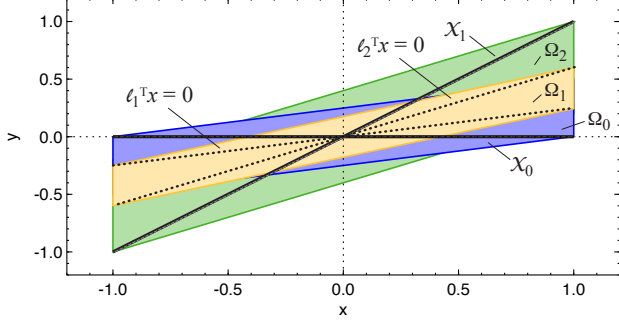
EXAMPLE 2.4. In Ex. 1.2 (hourglass), the trajectories are  $x(t) = x(0), y(t) = t \cdot x(0)$ , with initial states  $\mathcal{X}_0 = \{-1 \leq x(0) \leq 1, y(0) = 0\}$ . The support function over time for a direction vector  $\ell = (\alpha \ \beta)$  is

$$\begin{aligned} \rho_{\mathcal{X}_0}(\ell) &= \max_{x(0) \in \mathcal{X}_0} (\alpha \ \beta) \cdot (x(0) \ t \cdot x(0)) \\ &= \max(\alpha + \beta t, -\alpha - \beta t). \end{aligned} \quad (14)$$

Let's assume that flowpipe samples have been computed for directions  $\ell_1 = (-1 \ 4)$ ,  $\ell_2 = (-3 \ 5)$ ,  $\ell_3 = (1 \ 0)$ , as well as their negatives. Assuming the computation is exact, the lower and upper bounds of the flowpipes are identical. The flowpipe samples  $r_1(t)$ ,  $r_2(t)$ ,  $r_3(t)$  are shown in Fig. 4(a). The resulting outer approximation  $\lceil F \rceil_{0, T}$  is shown in Fig. 4(b).



(a) Flowpipe samples for directions  $\ell_1, \ell_2, \ell_3$



(b) Outer flowpipe approximation  $[F]_{0,T}$  in the state space, consisting of convex polyhedra  $\Omega_0, \Omega_1, \Omega_2$

**Figure 4: Approximating the flowpipe of Ex. 1.2 for a given set of directions  $\ell_1, \ell_2, \ell_3$ , and their negatives**

### 2.3 Polyhedral Approximations of Flowpipes

A flowpipe sampling describes a flowpipe in the same way that a support sampling describes a convex set, only that the flowpipe and its outer approximation can be nonconvex. We now show that the outer approximation of a flowpipe sampling with piecewise linear upper bounds is a set of convex polyhedra, one for each segment for which all upper bounds are concave.

Let  $F$  be a flowpipe sampling (11), such that for all  $k$ ,  $r_k^+(t), r_k^-(t)$  are piecewise linear. For constructing the polyhedra, we need some technical notation for describing the linear pieces. Let the  $i$ -th pieces of  $r_k^+(t), r_k^-(t)$  be

$$\begin{aligned} r_k^+(t) &= \alpha_{k,i}^+ t + \beta_{k,i}^+ \text{ for } \tau_{k,i}^+ \leq t \leq \tau_{k,i+1}^+, \\ r_k^-(t) &= \alpha_{k,i}^- t + \beta_{k,i}^- \text{ for } \tau_{k,i}^- \leq t \leq \tau_{k,i+1}^-. \end{aligned}$$

In the following, we consider the time interval  $[t_1, t_2]$  and will assume for simplicity that  $r_k^+(t), r_k^-(t)$  have breakpoints at the boundary of  $[t_1, t_2]$ , i.e., for some  $i' < i''$ ,  $t_1 = \tau_{k,i'}^+$  and  $t_2 = \tau_{k,i''}^-$ . Let  $I_k^+, I_k^-$  be the sets of indices  $i$  of the pieces of  $r_k^+(t)$ , respectively  $r_k^-(t)$ , that lie completely inside the time interval  $[t_1, t_2]$ .

If all computed flowpipe samples are concave over  $[t_1, t_2]$ , their outer approximation is convex:

LEMMA 2.5. *If for all  $k$ ,  $r_k^+(t)$  is concave on the time interval  $[t_1, t_2]$ , then  $[F]_{t_1, t_2}$  is the convex polyhedron*

$$\begin{aligned} [F]_{t_1, t_2} &= M [F]_{t_1, t_2}, \quad \text{where} \\ [F]_{t_1, t_2} &= \left\{ (x, t) \mid t_1 \leq t \leq t_2 \wedge \bigwedge_{k, i \in I_k^+} \ell_k^T x \leq \alpha_{k,i}^+ t + \beta_{k,i}^+ \right\}, \end{aligned} \quad (15)$$

and the matrix  $M$  maps  $(x, t) \in \mathbb{R}^{n+1}$  to  $x \in \mathbb{R}^n$ .

With Lemma 2.5, we can take a flowpipe sampling  $F$  and compute the support function of  $[F]_{t_1, t_2}$  by solving a single LP with  $O(n)$  variables and  $O(KZ)$  constraints, where  $K$  is the number of template directions and  $Z$  is a bound on the number of pieces of the  $r_k^+(t)$  in the time interval  $[t_j, t_{j+1}]$ .

With the above, we can construct a flowpipe approximation consisting of convex polyhedra  $\Omega_0, \dots, \Omega_N$  as follows:

1. Compute a piecewise linear flowpipe sample for each template direction.
2. Identify time intervals  $[t_0, t_1], \dots, [t_N, t_{N+1}]$ , with  $t_0 = 0$  and  $t_{N+1} = T$ , such that in each interval all samples have concave upper bounds.
3. Construct for each interval  $[t_i, t_{i+1}]$  its convex polyhedron  $\Omega_i = [F]_{t_i, t_{i+1}}$  using (15).

EXAMPLE 2.6. *The flowpipe samples of Ex. 2.9, shown in Fig. 4(a), are all concave on the time intervals  $[0, 0.25]$ ,  $[0.25, 0.6]$ , and  $[0.6, 1]$ . The outer approximation of the flowpipe consists of three convex polyhedra  $\Omega_0 = [F]_{0, 0.25}$ ,  $\Omega_1 = [F]_{0.25, 0.6}$ , and  $\Omega_2 = [F]_{0.6, 1}$ , shown in Fig. 4(b). The facet normals of  $\Omega_0$  are  $\ell_1, \ell_3$ , those of  $\Omega_2$  are  $\ell_2, \ell_3$ , and those of  $\Omega_1$  are a linear combination of  $\ell_1$  and  $\ell_2$ .*

The above approach produces a precise flowpipe approximation, but the number of polyhedra can be very large, especially if the concave intervals of the different flowpipe samples do not coincide. If an upper bound of a flowpipe samples is not concave on an interval, we can replace it by its concave envelope. The concave envelope of a piecewise linear function with  $N$  points, sorted along the time axis, can be computed in  $O(N)$  with the Graham scan. The approximation error can be measured via the distance to the envelope. In Sect. 3, we will present a clustering technique that establishes the largest concave intervals that can be created by relaxing the upper bounds, under a desired error bound.

Note that a convex outer approximation does not imply that the flowpipe segment is convex. We now derive a bound on the approximation error by using the lower bounds of the flowpipe samples.

LEMMA 2.7. *Let for all  $k$ ,  $r_k^+(t)$  be concave and  $r_k^-(t)$  be convex on the time interval  $[t_1, t_2]$ . Let the  $k$ -th facet slab of  $F$  be*

$$[F]_{t_1, t_2}^k = \left\{ (x_k, t) \in [F]_{t_1, t_2} \mid \bigwedge_{i \in I_k^-} \ell_k^T x_k \geq \alpha_{k,i}^- t + \beta_{k,i}^- \right\}, \quad (16)$$

*Then the Hausdorff distance between  $[F]_{t_1, t_2}$  and  $\mathcal{X}_{t_1, t_2}$  is bounded by*

$$\varepsilon_{t_1, t_2} = \max_{\|\ell\|_2=1} \varepsilon_{t_1, t_2}(\ell), \quad \text{where} \quad (17)$$

$$\varepsilon_{t_1, t_2}(\ell) = \max \left\{ \ell^T x - z \mid (x, t) \in [F]_{t_1, t_2} \wedge \bigwedge_{k=1, \dots, K} z \geq \ell^T x_k \wedge (x_k, t) \in [F]_{t_1, t_2}^k \right\} \quad (18)$$

For a given direction  $\ell$ , (18) can be formulated as a linear program. Consequently, Lemma 2.7, allows us to compute a bound on the directional approximation error  $\varepsilon_{t_1, t_2}(\ell)$  by solving a single LP with  $O(Kn)$  variables and  $O(K^2 Z)$  constraints. If we can solve the program for all  $\ell$ , we obtain a bound on the Hausdorff distance of  $[F]_{t_1, t_2}$  to the actual flowpipe segment.

## 2.4 Computing Flowpipe Samples for Affine Dynamics

We now present a way to construct flowpipe samples for affine dynamics of the form (1), i.e., an interval-valued function that bounds the support function of the reachable states at time  $t$  for a given direction. Our construction takes as input the initial set  $\mathcal{X}_0$ , a time horizon  $T$ , a template direction  $\ell$ , and an error bound  $\epsilon$ . It produces a flowpipe sample  $(\ell, [r^-(t), r^+(t)])$ , such that for all  $0 \leq t \leq T$ ,

$$r^+(t) - r^-(t) \leq \epsilon.$$

The sample is piecewise quadratic and can easily be approximated by a piecewise linear sample so that the techniques of the previous section can be applied. The construction is based on the approach in [4], from which it differs in two ways: First, we include a lower bound on the support function, which is used to evaluate the approximation error at all stages including clustering. Second, instead of computing forward with a certain time-step, we start with a time step that covers the whole time horizon and recursively refine with smaller steps on subdomains where the difference between upper and lower bound exceeds the error bound.

We exploit the superposition principle to adapt the approximation separately to the autonomous dynamics (created by  $\mathcal{X}_0$ ), and to the non-autonomous dynamics (created by  $\mathcal{U}$ ).  $\mathcal{X}_t$  can be decomposed into

$$\mathcal{X}_t = \mathcal{Z}_t \oplus \mathcal{Y}_t, \quad (19)$$

where  $\mathcal{Z}_t = e^{At} \mathcal{X}_0$  and  $\mathcal{Y}_t$  is the set of states reachable when starting from  $x = 0$  instead of  $\mathcal{X}_0$ :

$$\mathcal{Y}_t = \{x(t) \mid x(0) = 0, \forall \tau \in [0, t] \exists u(\tau) \in \mathcal{U} : \dot{x}(\tau) = Ax(\tau) + u(\tau)\}. \quad (20)$$

Note that  $\mathcal{Z}_0 = \mathcal{X}_0$  and  $\mathcal{Y}_0 = 0$ . We now turn to constructing a flowpipe sample  $\omega(t) = [\omega^-(t), \omega^+(t)]$  for  $\mathcal{Z}_t$ . We need the following notation. The *symmetric interval hull* of a set  $S$ , denoted  $\square(S)$ , is  $\square(S) = [-|x_1|; |x_1|] \times \dots \times [-|x_d|; |x_d|]$  where for all  $i$ ,  $|x_i| = \sup\{|x_i| \mid x \in S\}$ . Let  $M = (m_{i,j})$  be a matrix, and  $v = (v_i)$  a vector. We define  $|M|$  and  $|v|$  the absolute values of  $M$  and  $v$  respectively, i.e.,  $|M| = (|m_{i,j}|)$  and  $|v| = (|v_i|)$ . The approximation uses a transformation matrix  $\Phi_2$  defined as

$$\Phi_2(A, \delta) = \sum_{i=0}^{\infty} \frac{\delta^{i+2}}{(i+2)!} A^i, \quad (21)$$

which is computed similarly to a matrix exponential [4]. Our starting point is a linear interpolation between  $\mathcal{Z}_0$  and  $\mathcal{Z}_\delta$ . Using a forward, respectively backward, interpolation leads to error terms represented by sets  $\mathcal{E}_\Omega^+, \mathcal{E}_\Omega^-$ . The intersection of both error terms gives  $\mathcal{E}_\Omega$ . Using a normalized time variable  $\lambda = t/\delta$ , let

$$\begin{aligned} \mathcal{E}_\Omega(\delta, \lambda) &= (\lambda \mathcal{E}_\Omega^+(\delta) \cap (1 - \lambda) \mathcal{E}_\Omega^-(\delta)) \\ \mathcal{E}_\Omega^+(\delta) &= \square(\Phi_2(|A|, \delta) \square(A^2 \mathcal{X}_0)), \\ \mathcal{E}_\Omega^-(\delta) &= \square(\Phi_2(|A|, \delta) \square(A^2 e^{A\delta} \mathcal{X}_0)). \end{aligned}$$

The support function of  $\mathcal{E}_\Omega(\delta, \lambda)$  is piecewise linear,

$$\rho(\ell, \mathcal{E}_\Omega(\delta, \lambda)) = \sum_{i=1}^n \min(\lambda e_i^+, (1 - \lambda) e_i^-) |\ell_i|,$$

where vectors  $e^+$  and  $e^-$  are such that  $\rho(\ell, \mathcal{E}_\Omega^+) = |\ell|^\top e^+$  and  $\rho(\ell, \mathcal{E}_\Omega^-) = |\ell|^\top e^-$ .

An upper bound on the support function of  $\mathcal{Z}_t$  over a time interval  $[0, \delta]$  is easy to derive from the linear interpolation between  $\mathcal{Z}_0$ ,  $\mathcal{Z}_\delta$ , and the above error terms [4]. For deriving a lower bound, consider a support vector  $x^-$  of  $\mathcal{Z}_0 = \mathcal{X}_0$  in direction  $\ell$ . Since the support function of  $\mathcal{Z}_\delta = e^{A\delta} \mathcal{X}_0$  is the maximum of  $\ell^\top x$  over all  $x \in \mathcal{Z}_\delta$ , it is bounded below by the image of  $x^-$  at time  $\delta$ , i.e., by  $\ell^\top e^{A\delta} x^-$ . From the linear interpolation between  $x^-$  and  $e^{A\delta} x^-$  we derive a lower bound by subtracting a suitable error term. A similar argument can be made with the support vector  $x^+$  at the end of the interval, and we take the maximum of both lower bounds. Using the above error terms we obtain a flowpipe sample as follows.

**LEMMA 2.8.** *We consider the time interval  $[t_i, t_{i+1}]$ . Let  $\delta_i = t_{i+1} - t_i$ ,  $\ell_i = e^{A t_i} \ell$ ,  $\ell_{i+1} = e^{A \delta_i} \ell_i$ , let  $x^-$  be a support vector of  $\rho(\ell_i, \mathcal{X}_0)$ , and  $x^+$  be a support vector of  $\rho(\ell_{i+1}, \mathcal{X}_0)$ . Let  $\lambda = (t - t_i)/\delta_i$  and*

$$\omega^+(t) = (1 - \lambda) \rho_{\mathcal{X}_0}(\ell_i) + \lambda \rho_{\mathcal{X}_0}(\ell_{i+1}) + \rho_{\mathcal{E}_\Omega(\delta_i, \lambda)}(\ell_i) \quad (22)$$

$$\begin{aligned} \omega^-(t) &= \max\{(1 - \lambda) \ell_i^\top x^- + \lambda \ell_{i+1}^\top x^-, \\ &\quad (1 - \lambda) \ell_i^\top x^+ + \lambda \ell_{i+1}^\top x^-\} - \rho_{\mathcal{E}_\Omega(\delta_i, \lambda)}(\ell_i). \end{aligned} \quad (23)$$

Then  $\omega^-(t) \leq \rho_{\mathcal{Z}_t}(\ell) \leq \omega^+(t)$  for all  $t_i \leq t \leq t_{i+1}$ .

The approximation error of  $\omega^-(t), \omega^+(t)$  in the time interval  $[t_i, t_i + \delta_i]$  is

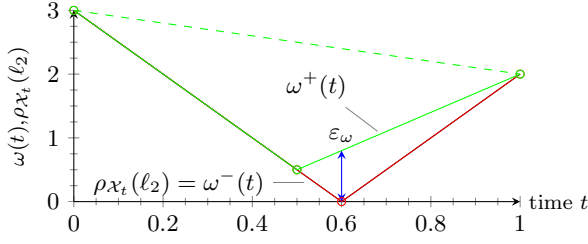
$$\varepsilon_\omega(t_i, t_{i+1}) = \max_{t_i \leq t \leq t_{i+1}} \omega^+(t) - \omega^-(t). \quad (24)$$

Note that the approximation error decreases at least linearly with  $\delta_i$ . To meet the given error bound  $\epsilon_\omega$ , we construct  $\omega^-(t), \omega^+(t)$  and the corresponding sequence of time points  $t_i$  by establishing a list of suitable intervals. We begin with a single interval  $[t_0, t_1] = [0, T]$ , which covers the entire time horizon. Each interval  $[t_i, t_{i+1}]$  in the list is processed in the following steps:

1. Construct  $\omega^-(t), \omega^+(t)$  on the interval  $[t_i, t_{i+1}]$  and compute  $\varepsilon_\omega(t_i, t_{i+1})$ .
2. If  $\varepsilon_\omega(t_i, t_{i+1}) > \epsilon_\omega$ , split the interval in two. Let  $t' = (t_i + t_{i+1})/2$ . Replace  $[t_i, t_{i+1}]$  with intervals  $[t_i, t']$ ,  $[t', t_{i+1}]$ , and process each starting with step 1.

**EXAMPLE 2.9.** *Consider computing a flowpipe sample of Ex. 1.2 (hourglass) for direction  $\ell_2$  and up to an error bound of  $\epsilon = 1$ , as illustrated by Fig. 5. There are no inputs, so  $r_2(t) = \omega(t)$ . We start with the interval  $[0, 1]$ , which yields as upper bound the linear interpolation between  $\omega^+(0) = 3$  and  $\omega^+(1) = 2$ , shown dashed in Fig. 5. In this example, the lower bound  $\omega^-(t)$  happens to coincide with  $r_2(t)$ . The initial approximation error is  $\varepsilon_\omega(0, 1) = 2.4$ . Since this exceeds  $\epsilon$ , the interval is split into two pieces,  $[0, 0.5]$  and  $[0.5, 1]$ . The approximation errors are  $\varepsilon_\omega(0, 0.5) = 0$  and  $\varepsilon_\omega(0.5, 1) = 0.6$ . They satisfy the error bound  $\epsilon$ , and we obtain the upper bound  $\omega^+(t)$  shown in Fig. 5.*

We now establish a flowpipe sample  $\psi(t) = [\psi^-(t), \psi^+(t)]$  for  $\mathcal{Y}_t$ . Computing  $\psi(t)$  is more difficult than computing  $\omega(t)$  because there is no analytic solution for the integral over the input  $u(t)$ . Because of the integration, the approximation error accumulates over time. In order to guarantee that for all  $t$  the (accumulated) error of  $\psi(t)$  is below a given bound



**Figure 5: Computing a flowpipe sample of Ex. 1.2 for direction  $\ell_2$**

$\epsilon_\psi$ , we impose that the accumulated error at the end of each interval  $[t_i, t_{i+1}]$  must lie below the *error rate*  $\epsilon_\psi \cdot t_{i+1}/T$ . We use a two-step process: We first compute  $\psi(t_i)$  at discrete points in time  $t_i$  such that the desired error rate is met. Based on these values we then define  $\psi(t)$  over continuous time. To bound the approximation error we use the error term

$$\mathcal{E}_\Psi(\mathcal{U}, \delta) = \square(\Phi_2(|A|, \delta) \square(A\mathcal{U})).$$

For the following two lemmas, we assume the sequence of time points  $t_i$  as given. Its construction is presented afterwards, when the required error terms have been defined. We have the following bounds on  $\rho_{y_i}(\ell)$  at the discrete time points  $t_i$ .

LEMMA 2.10. [4] *Let  $t_0 = 0, t_1, t_2, \dots, t_N = T$  be an increasing sequence of time points. Let  $\delta_i = t_{i+1} - t_i$ ,  $\ell_i = e^{A t_i} \ell$ ,  $\ell_{i+1} = e^{A \delta_i} \ell_i$ ,  $\psi_{t_0}^+ = 0$ ,  $\psi_{t_0}^- = 0$ , and*

$$\psi_{t_{i+1}}^+ = \psi_{t_i}^+ + \delta_i \rho_{\mathcal{U}}(\ell_i) + \rho_{\mathcal{E}_\Psi(\mathcal{U}, \delta_i)}(\ell_i) \quad (25)$$

$$\psi_{t_{i+1}}^- = \psi_{t_i}^- + \delta_i \rho_{\mathcal{U}}(\ell_i) - \rho_{-A\Phi_2(A, \delta_i)\mathcal{U}}(\ell_i). \quad (26)$$

Then for all  $t_i$ ,  $\psi_{t_i}^- \leq \rho_{y_i}(\ell) \leq \psi_{t_i}^+$ .

Based on the bounds on  $\rho_{y_i}(\ell)$  at the discrete times  $t_i$ , we obtain the following bounds over the intervals  $[t_i, t_{i+1}]$ .

LEMMA 2.11. *Let  $\lambda = (t - t_i)/\delta_i$  and*

$$\psi^+(t) = \psi_{t_i}^+ + \lambda \delta_i \rho_{\mathcal{U}}(\ell_i) + \lambda^2 \rho_{\mathcal{E}_\Psi(\mathcal{U}, \delta_i)}(\ell_i), \quad (27)$$

$$\begin{aligned} \psi^-(t) = & \psi_{t_i}^- + \lambda \delta_i \rho_{\mathcal{U}}(\ell_i) - \lambda \rho_{-A\Phi_2(A, \delta_i)\mathcal{U}}(\ell_i) \\ & - \lambda^2 \rho_{\mathcal{E}_\Psi(\mathcal{U}, \delta_i)}(\ell_i). \end{aligned} \quad (28)$$

Then  $\psi^-(t) \leq \rho_{y_i}(\ell) \leq \psi^+(t)$  for all  $t_i \leq t \leq t_{i+1}$ .

We construct the time intervals  $[t_i, t_{i+1}]$  by refinement until the error  $\psi(t)$  falls below  $\epsilon_\psi$ . According to the above Lemmas, the error bound on the interval  $[t_i, t_{i+1}]$  is defined by the following sequence, starting with  $\epsilon_\psi(t_0) = 0$ :

$$\begin{aligned} \epsilon_\psi(t_i, t_{i+1}) = & \epsilon_\psi(t_i) + \max_{0 \leq \lambda \leq 1} 2\lambda^2 \rho_{\mathcal{E}_\Psi(\mathcal{U}, \delta_i)}(\ell_i) \\ & + \lambda \rho_{-A\Phi_2(A, \delta_i)\mathcal{U}}(\ell_i), \quad \text{where} \end{aligned} \quad (29)$$

$$\epsilon_\psi(t_{i+1}) = \epsilon_\psi(t_i) + \rho_{\mathcal{E}_\Psi(\mathcal{U}, \delta_i)}(\ell_i) + \rho_{-A\Phi_2(A, \delta_i)\mathcal{U}}(\ell_i). \quad (30)$$

When choosing the time points  $t_i$ , we must ensure that largest error in the interval  $[t_i, t_{i+1}]$  lies below the error bound, i.e.,  $\epsilon_\psi(t_i, t_{i+1}) \leq \epsilon_\psi$ . To take into account that the error  $\epsilon_\psi(t_i)$  accumulates, we also ensure that the rate of the accumulated error stays below  $\epsilon_\psi/T$ , i.e.,  $\epsilon_\psi(t_{i+1}) \leq \epsilon_\psi \cdot t_{i+1}/T$ . Beginning with a single interval  $[t_0, t_1] = [0, T]$ , each interval  $[t_i, t_{i+1}]$  is processed in the following steps:

1. Compute  $\epsilon_\psi(t_{i+1})$  and  $\epsilon_\psi(t_i, t_{i+1})$ .
2. If  $\epsilon_\psi(t_{i+1}) > \epsilon_\psi \cdot t_{i+1}/T$  or  $\epsilon_\psi(t_i, t_{i+1}) > \epsilon_\psi$ , split the interval in two. Let  $t' = (t_i + t_{i+1})/2$ . Replace  $[t_i, t_{i+1}]$  with intervals  $[t_i, t']$ ,  $[t', t_{i+1}]$ , and process each starting with step 1.

Using the superposition principle (19), we finally combine  $\omega(t)$  and  $\psi(t)$  to obtain a flowpipe sample for  $\mathcal{X}_t$  as

$$r(t) = [r^-(t), r^+(t)] = \omega(t) + \psi(t).$$

The error bound on  $r(t)$  is below  $\epsilon = \epsilon_\omega + \epsilon_\psi$ .

### 3. CLUSTERING IN SPACE-TIME

Given a flowpipe sampling, our goal is to construct a sequence of convex sets that cover the flowpipe and that are no further than a given distance  $\epsilon$  from it. For computational efficiency, our distance measure is the directional error in each of the sampled directions, but this implies also a distance in the Hausdorff sense.

We now given an informal description of our clustering algorithm, deferring a formal discussion to the subsections that follow. The algorithm takes as input a flowpipe sampling  $F = \{(\ell_1, r_1(t)), \dots, (\ell_K, r_K(t))\}$  and an error bound  $\epsilon$ , and produces a flowpipe sampling  $F'$  by replacing the upper bounds  $r_k^+(t)$  with a piecewise concave envelope with as few pieces as possible for the given error bound. The basic principle is to (over)approximate the upper bounds  $r_i^+(t)$  of the flowpipe samples with a set of piecewise concave hulls  $y_i(t)$ , which are constructed such that they are concave over the same pieces. Recalling from in Sect. 2.3 that an outer approximation in the form of a convex polyhedron can be constructed for each concave piece, this effectively reduces the number of convex sets.

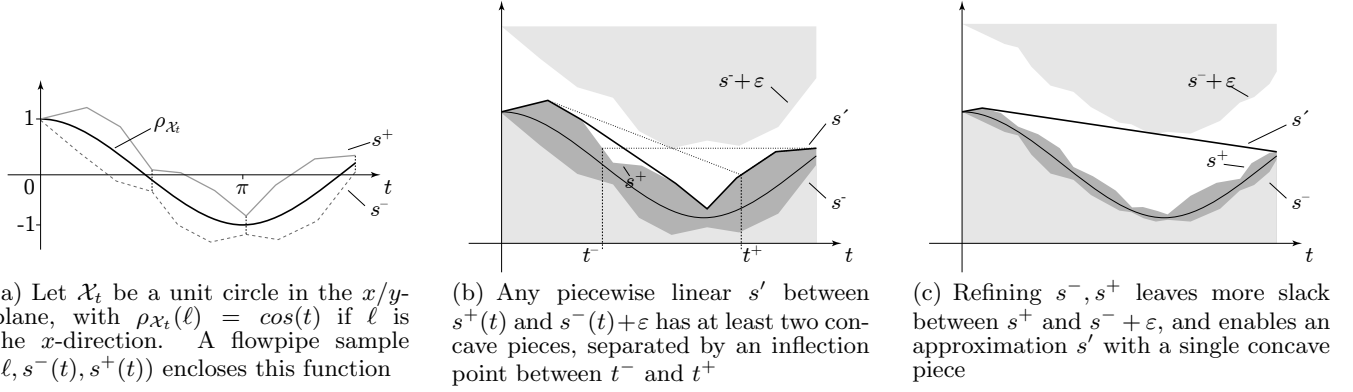
Let  $\rho_i(t) = \rho_{\mathcal{X}_t}(\ell_i)$  be the actual value of the support function over time. By definition,  $r_i^-(t) \leq \rho_i(t) \leq r_i^+(t)$ . The goal of our clustering is to produce a piecewise concave hull  $y_i(t)$  that is no farther than  $\epsilon$  away from the actual value  $\rho_i(t)$ , i.e., such that  $\rho_i(t) \leq y_i(t) \leq \rho_i(t) + \epsilon$ . Since only  $r_i^-(t)$  and  $r_i^+(t)$  are known, we must construct the  $y_i(t)$  such that

$$r_i^+(t) \leq y_i(t) \leq r_i^-(t) + \epsilon. \quad (31)$$

Finding the minimal number of concave pieces for a function between a lower and an upper bound is possible by establishing the *inflection intervals* of  $(r_i^+(t), r_i^-(t) + \epsilon)$ , which will be presented in Sect. 3.1. The set of inflection intervals has the following property: Any piecewise concave function between  $r_i^+(t)$  and  $r_i^-(t) + \epsilon$  has at least one inflection point inside every inflection interval. The number of inflection intervals is thus equal to the minimum number of concave pieces of any  $y_i(t)$ . To have the minimum number of concave pieces, we must find the minimum number of points such that there is at least one point in every inflection interval of every sample. This turns out to be a graph coloring problem that is described in Sect. 3.2. The clustering step itself terminates with the construction of a piecewise concave hull of the flowpipe samples with a minimum number of pieces. Convex polyhedra can be derived from the concave pieces as previously described in Sect. 2.3.

If the clustering results in a number of concave pieces that is still considered too high, one can try to reduce the number further by recomputing the flowpipe samples with higher accuracy. This brings the  $r_i^-(t)$  and  $r_i^+(t)$  closer together,





**Figure 6:** Given a flowpipe sample with bounds  $s^-(t), s^+(t)$  and a desired approximation error  $\epsilon$ , we construct a piecewise concave function  $s'(t)$  that lies between  $s^+(t)$  and  $s^-(t) + \epsilon$ . Fewer concave pieces in  $s'(t)$  mean fewer convex sets produced by the clustering

which increases the slack in the bounds (31) used for clustering. As illustrated by Fig. 6, the new bounds may be wide enough to admit a fewer pieces. We can obtain a lower bound on the number of pieces by computing the number of inflection intervals for the bounds

$$r_i^-(t) \leq y_i(t) \leq r_i^+(t) + \epsilon. \quad (32)$$

### 3.1 Inflection Intervals of a Flowpipe Sample

Let  $l(t), u(t)$  be a pair of piecewise linear functions with domain  $[0, T]$  such that  $l(t) \leq u(t)$ . An *inflection interval* is an interval over  $t$  that contains at least one inflection point of any piecewise concave function  $y(t)$  lying on or between  $l(t)$  and  $u(t)$ , and no points that are not inflection points of a piecewise concave  $y(t)$  with a minimum number of pieces. As a consequence of this definition, the minimum number of pieces of any  $y(t)$  is equal to the number of inflection intervals of  $(l(t), u(t))$ .

Let the breakpoints of  $l(t)$  be  $l_0$  to  $l_N$ . We propose the following algorithm for finding inflection intervals (for simplicity we omit some special cases), see Fig. 7 for an illustration:

1. Perform a greedy piecewise concave minimal function construction from  $l_0$  to  $l_N$ : choose at each step the point on the lower bound farthest towards  $l_N$  that is still visible.
2. Denote the breakpoints where the function is convex by  $b_0, \dots, b_z$ .
3. Perform a greedy piecewise concave minimal function construction in reversed direction, from  $l_N$  to  $l_0$ .
4. Denote the breakpoints where the function is convex by  $a_z, \dots, a_0$ .
5. Return the inflection intervals  $I_0 = [a_0, b_0], \dots, I_z = [a_z, b_z]$ .

**PROPOSITION 3.1.** *Given intervals  $I_0, \dots, I_z$  returned by the above algorithm, there exists a piecewise concave function between  $l(t)$  and  $u(t)$  with  $z+1$  inflection points, one in every interval  $I_i$ . There exists no piecewise concave function between  $l(t)$  and  $u(t)$  with less than  $z+1$  inflection points.*

**PROOF.** Let us consider one of the  $b_i$ , let us denote it  $b$  for simplicity, and  $l_b$  the previous vertex in the piecewise

concave piecewise linear function.  $l_b$  is on  $l(t)$  (but not necessarily one of the  $l_i$ ) otherwise the function would not be minimal.  $b$  is on a segment  $]l_i, l_{i+1}[$  and there is a point  $u' \in u(t)$  on the segment  $[l_b, b]$  otherwise the function would not have been greedily constructed. Let us take  $x$  in  $]b, l_{i+1}[$ , any concave function on  $[b, x]$  above  $l(t)$  and below  $u(t)$  must be above  $l_b$  and  $x$  and below  $u'$  which is not possible since  $u'$  is below  $[l_b, x]$ . Thus any piecewise concave function must contain at least one inflection point on  $]l_b, x[$ , and thus at least one on  $]b, b]$ , and one on each  $]l_{b_i}, b_i]$ . Since the intervals are disjoint, the greedy algorithm reaches a minimum number of inflection point. Similarly any piecewise concave function must contain at least one inflection point on each  $[a_i, l_{a_i}]$ .  $\square$

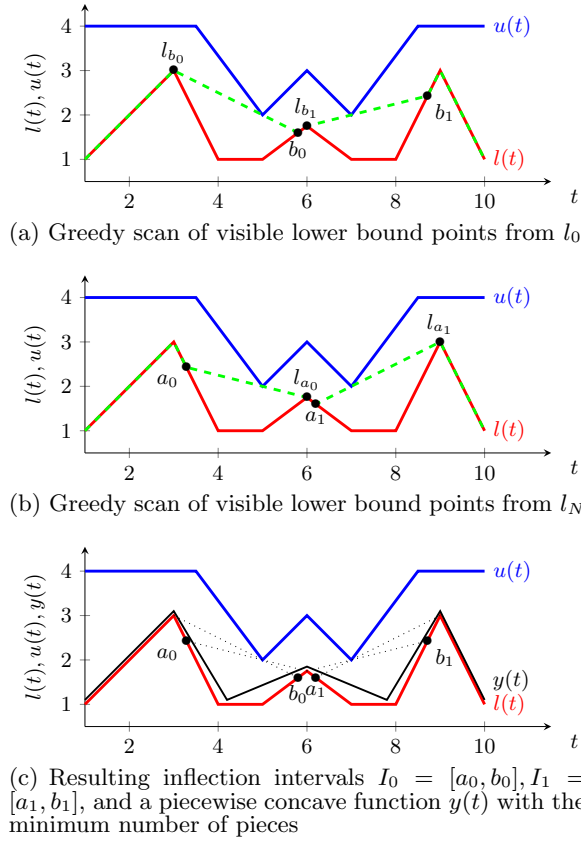
### 3.2 Combining Inflection Intervals

Having established the inflection intervals for each flowpipe sample, we combine them to find the minimum number of inflection points, as well as their possible positions, for our piecewise cover of all samples. Recall that the pieces of the piecewise cover we seek are common to all samples. We therefore need to pick at least one point from every inflection interval of every sample. To minimize their number, we construct their common sub-intervals, which we call *overlap intervals*.

For each function we have a (possibly empty) set of inflection intervals  $I_i$  obtained using the algorithm of Sect. 3.1. For the following it is not relevant that the inflection intervals originate from different functions, so let  $I_0, I_1, \dots, I_z$  simply be the set of all inflection intervals. We need to partition the intervals into groups inside which all intervals overlap. The output of the algorithm consists of the groups and for each group their common *overlap intervals*  $J_j$ .

Finding maximal groups of overlapping intervals is equivalent to a coloring problem. Each color  $j$  corresponds to one of the groups and defines an overlap interval, which consists of the overlap between all members of the group. Two intervals  $I_1, I_2$  need to be colored differently if they do not overlap, i.e., if  $I_1^+ < I_2^-$  or  $I_2^+ < I_1^-$ . This relationship is captured by the *comparability graph*, whose vertices are the intervals  $I_i$ . Its edges are given by  $I_i \rightarrow I_j \Leftrightarrow I_i^+ < I_j^-$ , which is the so-called interval ordering (a strict partial order). Our problem is equivalent to finding a coloring of the





**Figure 7: Finding the set of inflection intervals, inside each of which any piecewise concave function between  $l(t)$  and  $u(t)$  has at least one inflection point**

comparability graph with the smallest number of colors such that no two adjacent vertices have the same color. Once the intervals have been colored, each color corresponds to a set of intervals that all overlap. We may freely choose an inflection point from inside the common region for that color.

It is known that the interval ordering is a perfect elimination ordering of the comparability graph of a set of intervals. Consequently, a greedy coloring algorithm produces the optimal result if it chooses the vertices in an order that satisfies the interval ordering [11]. Such an order of the vertices can be obtained by a topological sort, i.e., a depth-first search in the graph. The total complexity is determined by the size of the comparability graph and therefore  $O(z^2)$ , where  $z$  is the number of intervals.

#### Choosing Inflection Points.

Our final set of inflection points consists of one point from each overlap interval  $J_j$ . The approximation error of this choice can be measured as the distance of the lower bounds to the resulting piecewise concave functions. The choice in one inflection interval generally influences the approximation error of its neighboring intervals as well, so the optimal choice is a multivariate optimization problem. In our experiments, we have observed that the overlap between the intervals of different directions is usually small, and that choosing inflection points in the middle of each interval yields results that are close to the local optimum.

## 4. EXPERIMENTAL RESULTS

In this section, we present experiments with the proposed flowpipe approximation and clustering algorithms and compare it with the approximation from [4]. These algorithms are intended to be used for reachability of hybrid systems in the verification tool SpaceEx [4]. There, each flowpipe approximation is followed by the computation of the image of all enabled discrete transitions. This image computation involves intersecting the flowpipe with the invariants of source and target locations, as well as the transition guard, which can be carried out efficiently on polyhedra. The final result of a flowpipe approximation therefore the polyhedral outer approximation as described in Sect. 2.3. Note that other variants of the reachability algorithm avoid polyhedra, e.g., by carrying out the intersection on the support function level through transformation into an optimization problem [9, 6].

We compare the following variants:

LGG (state-space approximation without clustering) variable time-step flowpipe construction in the state-space, then outer polyhedral approximation, both as in [4],

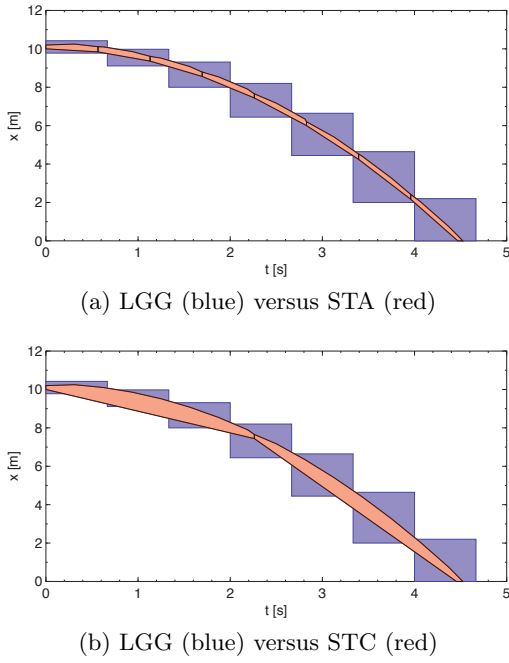
STA (space-time approximation with all pieces) flowpipe construction as in Sect. 2, then outer polyhedral approximation of all pieces as in Sect. 2.3 (no clustering),

STC (space-time approximation with clustering) flowpipe construction as in Sect. 2 and clustering as in Sect. 3, then outer polyhedral approximation as in Sect. 2.3.

Note that the STA/STC implementation is still a prototype, and we expect that memory consumption and clustering runtime can be reduced. The parameter settings are not entirely comparable between LGG and STA/STC, since the error bounds in STA/STC are conservative, while in LGG they are mere estimates that do not take the nonconvexity error in account. The error bound in STC measures the total error, including both flowpipe approximation and clustering. We choose that 80% of the error can be taken up by the flowpipe approximation, so that at least 20% of the error bound remain as slack for the clustering step.

To avoid a lengthy description of the models, they are available for download on the SpaceEx website [3]. For illustration, consider a ball in free-fall together with a clock, with 3 variables  $x, v, t$ , dynamics  $\dot{x} = v$ ,  $\dot{v} = -1$ ,  $\dot{t} = 1$ , and initial states  $10 \leq x \leq 10.2$ ,  $v = t = 0$ . We construct the flowpipe until  $x$  falls below 0. The axis directions are used as template directions, so LGG creates bounding boxes of flowpipe segments in the state space. The flowpipe approximation of STA/STC creates a bounding box for each point in time, which projected to the state space yields polyhedra with facet normals other than the template directions, as Fig. 8 illustrates. The error bound  $\epsilon = 1$  is fairly large, and the clustering step in STC uses the slack to reduce the number of sets from 8 sets (STA) to 2 sets.

Table 1 shows performance results obtained on a laptop with i7 processor and 8 GB RAM. All examples use the axis directions as template directions. For each algorithm, the table shows the time taken for flowpipe approximation, the time taken for clustering and constructing the polyhedral approximation, and the memory consumption. As an implementation-independent indicator of the computational cost, it shows the total number of times the support function of the initial set has been evaluated. The key column is the total number of convex sets covering the flowpipe, and the



**Figure 8: Flowpipe approximation of a ball in free fall (position over time), with axis directions as template directions and a directional error bound  $\epsilon = 1$**

goal of the STC algorithm is to reduce it as much as possible for the given error bound.

The results indicate that the flowpipe approximation in space-time with clustering (STC) can produce a flowpipe cover with a small number of sets, while meeting the desired directional error bounds. The construction uses template directions, with which the reachable set is approximated in space-time, pointwise for every time instant. The projection onto the state space produces polyhedra with facet normals that are linear combinations of the template directions. Compared to our previous work, which approximates the flowpipe directly in the state space, this can improve precision and reduce the number of sets at the same time.

## 5. ACKNOWLEDGMENTS

The authors thank Oded Maler for valuable discussions and support and the anonymous reviewers for their comments and corrections.

## 6. REFERENCES

- [1] E. Asarin, T. Dang, and O. Maler. The d/dt tool for verification of hybrid systems. In E. Brinksma and K. G. Larsen, editors, *CAV*, volume 2404 of *LNCS*, pages 365–370. Springer, 2002.
- [2] A. Chutinan and B. H. Krogh. Computational techniques for hybrid system verification. *IEEE Trans. Automat. Contr.*, 48(1):64–75, 2003.
- [3] G. Frehse. SpaceEx state space explorer. Verimag, Grenoble, <http://spaceex.imag.fr>, 2010.
- [4] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid

**Table 1: Performance results for different examples**

Algo	Fl. T. [s]	Cl. T. [s]	Mem. [MB]	#eval	#sets
Helicopter with controller & clock, 29 variables, $\epsilon = 0.025$					
LGG	5.9	–	14	85144	1440
STA	9.0	3.6	2390	103726	2649
STC	9.4	4.8	203	103726	32
Helicopter with diff. controller & clock, 29 variables, $\epsilon = 0.025$					
LGG	10.4	–	15	150278	2563
STA	14.5	0.0	2620	154026	2568
STC	14.3	19.0	225	154026	10
Ball in free-fall & clock, 3 variables, $\epsilon = 0.001$					
LGG	0.04	–	11	1770	261
STA	0.02	0	14	1453	85
STC	0.02	0	12	1453	56
Ball i. ff. w. input disturbance & clock, 3 variables, $\epsilon = 0.001$					
LGG	6.3	–	94	344676	17208
STA	29.9	0	15	1580834	256
STC	29.9	0	12	1580834	64
Three-tank system, 3 variables, $\epsilon = 0.125$					
LGG	0.03	–	2	1032	105
STA	0.03	0	15	756	137
STC	0.03	0	2	756	9
Overhead crane, 4 variables, $\epsilon = 0.05$					
LGG	0.12	–	11	3944	369
STA	0.17	0	37	4896	633
STC	0.17	0	13	4896	15

FLT: flowpipe construction time, Cl.T.: clustering and polyhedral approximation time, Mem.: memory consumption, #eval: number of evaluations of the support function of the initial set, #sets: number of convex sets covering the flowpipe

systems. In G. Gopalakrishnan and S. Qadeer, editors, *CAV*, volume 6806 of *LNCS*, pages 379–395. Springer, 2011.

- [5] G. Frehse, C. Le Guernic, and R. Kateja. Flowpipe approximation and clustering in space-time. Technical Report TR-2013-1, Verimag, February 2013.
- [6] G. Frehse and R. Ray. Flowpipe-guard intersection for reachability computations with support functions. In *IFAC ADHS*, pages 94–101, 2012.
- [7] A. Girard. Reachability of uncertain linear systems using zonotopes. In M. Morari and L. Thiele, editors, *HSCC*, volume 3414 of *LNCS*, pages 291–305. Springer, 2005.
- [8] A. B. Kurzhanski and P. Varaiya. Ellipsoidal techniques for reachability under state constraints. *SIAM J. Control and Optimization*, 45(4):1369–1394, 2006.
- [9] C. Le Guernic and A. Girard. Reachability analysis of hybrid systems using support functions. In A. Bouajjani and O. Maler, editors, *CAV*, volume 5643 of *LNCS*, pages 540–554. Springer, 2009.
- [10] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev. *Interactive Decision Maps*, volume 89 of *Applied Optimization*. Kluwer, 2004.
- [11] F. Maffray. On the coloration of perfect graphs. In B. A. Reed and C. L. Sales, editors, *Recent Advances in Algorithms and Combinatorics*, CMS Books in Mathematics, pages 65–84. Springer, 2003.
- [12] P. Prabhakar and M. Viswanathan. A dynamic algorithm for approximate flow computations. In M. Caccamo, E. Frazzoli, and R. Grosu, editors, *HSCC*, pages 133–142. ACM, 2011.
- [13] P. Varaiya. Reach set computation using optimal control. In *Proc. KIT Workshop*, pages 377–383, 1997.