

# Set-based Control for Disturbed Piecewise Affine Systems with State and Actuation Constraints

Bastian Schürmann<sup>a,\*</sup>, Riccardo Vignali<sup>b</sup>, Maria Prandini<sup>b</sup>, Matthias Althoff<sup>a</sup>

<sup>a</sup>*Department of Informatics, Technische Universität München, Boltzmannstr. 3, 85748 Garching, Germany*

<sup>b</sup>*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milano, 20133 Italy*

---

## Abstract

We address the finite horizon control of a discrete time piecewise affine (PWA) system, which is affected by an additive bounded disturbance. The goal is to robustly drive the state of the system to some target region, while satisfying state and actuation constraints. This problem is challenging due to the mixed discrete and continuous dynamics, requiring the exploration of many mode sequences. We address this problem by proposing a two-step approach which is based on designing a reference trajectory followed by a feedback controller that tries to make all solutions adhere to the reference mode sequence. As a result, the number of considered mode sequences can be drastically reduced. Termination is guaranteed since the time horizon and the number of modes are finite. Key to the computational efficiency of this procedure is the choice of the reference trajectory, which should be designed to avoid the branching of mode sequences from the reference sequence. The resulting set-based feedback control law is simple and easy to implement. Due to the use of reachability analysis, our results are formally correct. A quadruple-tank benchmark example shows the effectiveness of our approach.

### *Keywords:*

Piecewise affine systems, set-based control, constrained systems, disturbed systems, reachability analysis.

---

## 1. Introduction

Hybrid systems, i.e., systems with both discrete and continuous dynamics, are becoming a commonly adopted modeling framework, as there are more and more applications where discrete computation tasks closely interact with the continuous, physical world. Since many safety-critical applications are developed using hybrid models, their control design methods require robust safety guarantees that account for uncertainties.

In this paper, we focus on the class of piecewise affine (PWA) systems that evolve according to a different affine dynamics based on the value taken by the state in a polyhedral partition defining the so-called modes. PWA systems are quite common since they naturally arise as models in various application contexts. They are equivalent to the class of mixed logical dynamical

---

\*Corresponding author.

*Email addresses:* [bastian.schuermann@tum.de](mailto:bastian.schuermann@tum.de) (Bastian Schürmann), [riccardomaria.vignali@polimi.it](mailto:riccardomaria.vignali@polimi.it) (Riccardo Vignali), [maria.prandini@polimi.it](mailto:maria.prandini@polimi.it) (Maria Prandini), [althoff@tum.de](mailto:althoff@tum.de) (Matthias Althoff)

*Preprint submitted to Nonlinear Analysis: Hybrid Systems*

*December 13, 2019*

systems [1]. In addition to their capability in modeling non-smooth dynamics, they can also be used as a model for nonlinear (smooth) systems by using hybridization methods such as those in [2–5]. Various analysis and control problems, such as model reduction [6, 7], controllability and observability [8], identification [9–11], fault detection and estimation, tracking, stabilization, and control [12–16], [17, Chapters 4 and 5], have been studied for the class of PWA systems. The vast literature on PWA systems is primarily motivated by their significant modeling capabilities, but also by the fact that their description via affine equations and constraints can simplify analysis and design.

Our goal in this paper is to design a feedback controller that drives a PWA system to some target region in a given finite horizon, while satisfying state and actuation constraints. Since the exact initial state of the system is unknown and its evolution is affected by some additive bounded disturbance, the controller has to compensate for these sources of uncertainty, within its actuation capabilities, to meet state constraints. We propose adopting set-based control design for this purpose. As mentioned above, hybridization methods can be used to obtain PWA abstractions for nonlinear systems, so that our newly developed method is also applicable to this class of systems. We demonstrate this with a numerical example at the end of the paper.

### *1.1. Set-Based Approaches in Formal Verification*

Set-based approaches have been introduced in formal verification to check if a certain property related to the temporal evolution of a system is satisfied, see for example [18, 19]. The idea is to compute the sets of states that the system can reach by propagating them over time. The efficiency of set propagation depends on the adopted representation of reachable sets. Different set representations are known in the literature, such as polytopes [20], ellipsoids [21], support functions [22], Taylor models [23], zonotopes [24], zonotope bundles [25], polynomial zonotopes [26], constrained zonotopes [27], and combinations of support functions and zonotopes [28]. We propose zonotopes, which are a special class of convex polytopes that are represented by a center and a linear combination of a finite set of generators. They are closed under linear transformations and the Minkowski sum, so that if the system dynamics is affine and the initial state and the system inputs take values in zonotopes, reachable sets are zonotopes as well and easily computable, even in the case of high-dimensional systems [24, 28]. Unfortunately, zonotopes are not closed under intersection, which makes their adoption for the formal verification of PWA systems challenging since reachable sets can intersect with multiple modes. If this happens, different solutions have been proposed in the literature; however, all with some drawback: one way is to split sets, which requires over-approximating the intersection of the reachable set within each mode and starting a new set-based computation for each new set. Besides the resulting conservatism due to possible over-approximation of the intersections by simpler geometric sets, the computation complexity increases with each new split [29, 30]. Alternatively, instead of splitting the set, one can apply the combined dynamics of the intersected modes in an over-approximative way. This strategy is used in [31], which deals with continuous time systems, and leads to computations with interval matrices, which can be rather conservative. Another alternative is proposed in [32], where instead of computing geometric intersections explicitly, these intersections are over-approximated by nonlinear maps which consider the reachable set during a possible intersection time. While this approach has several advantages, its over-approximations are rather conservative if the potential intersection duration is too long.

### 1.2. Related Contributions on PWA System Control Design

Combinatorial problems often appear in control design for PWA systems [33], including when there is no uncertainty affecting its evolution (see for example [34–41]). To limit the complexity of solving these problems, some approaches exist which use branch-and-bound methods to reduce the number of combinations [42]. A feedback control policy can be computed by resorting to set invariance and multiparametric programming for constrained PWA systems affected by a bounded disturbance, as proposed in [43]. The resulting PWA state feedback control law is optimal, but quite complex to represent and apply online (a polyhedral partition of 417 regions is required for the PWA controller for the two-dimensional example with two modes in [43]). Robust control for PWA systems is also addressed in [44] by imposing some robust mode control restriction (see [45]) to ensure that the mode sequence of the controlled system is independent of the disturbance realization. This restriction may limit the possibility of finding a solution. Other optimal control approaches for hybrid systems, such as [46], require solving a Hamilton-Jacobi equation, which is often prohibitively hard in practice. More recent techniques, on the other hand, use hybrid control Lyapunov functions to stabilize hybrid systems [47]. Reachability and controller synthesis for PWA systems using simplices can be found in [48], and an extension to output feedback for general polytopes is given in [49]. An important class of formal control algorithms are abstraction-based controllers, which discretize the state and input spaces and use automata theory to compute controllers for the resulting finite state abstraction. They are not only able to consider complex specifications, they can also be applied to switching dynamics [50–55].

There is a large interest in model predictive control (MPC) [56] of PWA systems [57–59], including those with bounded disturbances [60–62]; however, many approaches rely on min-max optimizations which can be very costly. Tightening constraints is one method for reducing computational costs [63]. In tube-based MPC [64] of constrained linear systems with bounded disturbances, the optimization problem (i.e., minimization of the cost for the nominal system) is separated from robustness issues by bounding the deviation from the nominal system by a sequence of invariant sets. A generalization of [64] to PWA systems is proposed in [65] under the rather restrictive assumption that a common Lyapunov function for the affine dynamics exists. The approach requires tightening of constraints and the online solution of a mixed integer program (MIP), where binary variables account for mode switching. Alternatively, multiparametric programming makes an explicit MPC solution possible, with the same drawbacks related to the control law representation and online application as in [43]. A variant of the standard extension of [64] to PWA systems is proposed in [66], where the actual mode of the real system is used when defining the nominal trajectory.

### 1.3. Proposed Set-Based Control Approach

To tackle the present control problem at hand, we propose a new approach which is able to provide formal guarantees while scaling better than alternative existing approaches, as shown in Sec. 7. Our set-based control approach consists of two parts: a smart choice of the reference trajectory, which directly takes the distances to mode boundaries into account and thereby aims at reducing the number of splits, as demonstrated in Fig. 1, and trajectory tracking. To track the reference trajectory, we extend the set-based control approach for purely continuous systems in [67] to hybrid systems, and, in particular, to uncertain PWA systems subject to state and actuation constraints. The extension is not trivial due to the hybrid dynamics of the system and the possible occurrence of mode splitting as discussed before.

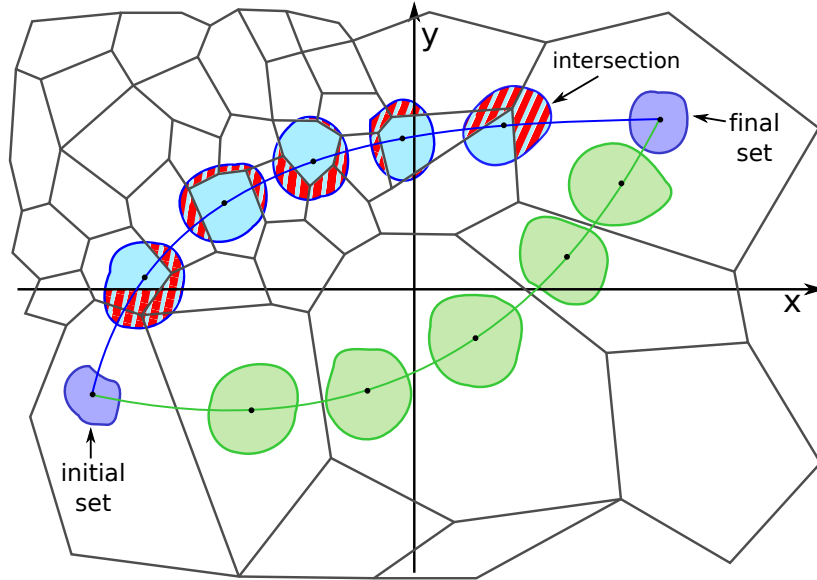


Figure 1: Example of two reference trajectories (blue and green lines) and reachable set propagation according to the dynamics associated with the reference mode sequence: tracking the green trajectory is much easier because reachable sets continue to evolve within the reference mode sequence. While the reachable sets of the blue trajectory have a similar size, they often intersect with the mode boundaries shown in gray.

In contrast to verification, however, the control input can be chosen to avoid splitting completely, or at least reduce the number of splittings. Without splits, the PWA system becomes a time-varying affine system which drastically simplifies the set-based feedback control design. Our approach has several more advantages:

- no need to tighten constraints and no need to compute invariant sets;
- mixed integer programs are solved offline;
- feedback control law is easy to store and apply online;
- no stability proof needed because we are addressing a finite horizon control problem for which we check constraint satisfaction via reachability analysis;
- the possibility that the reachable set may be split is explicitly taken into account and some countermeasures are provided when designing the reference trajectory and the feedback controller, which is not the case in robust tube-based MPC.

#### 1.4. Organization of the Paper

The remainder of the paper is organized as follows: We start by introducing some basic notations in Section 2. The problem formulation is given in Section 3. After this, we provide an overview of our approach in Section 4. Section 5 describes the computation of the reference trajectory and Section 6 deals with the computation of the feedback controller. The effectiveness of the proposed control design is illustrated by a numerical example of a quadruple tank benchmark in Section 7, and some conclusions are drawn in Section 8.

## 2. Basic Notations

We define a (convex) polyhedron  $\mathcal{P} \subset \mathbb{R}^h$  as the intersection of  $q$  half-spaces (H-representation [68]), which can be expressed through  $P_A \in \mathbb{R}^{q \times h}$  and  $P_B \in \mathbb{R}^q$  as follows:  $\mathcal{P} = \{z \in \mathbb{R}^h | P_A z \leq P_B\}$ . For ease of notation we shall refer to  $(P_A, P_B)$  as the H-representation of  $\mathcal{P}$ . The intersection of  $k$  polyhedra  $\mathcal{P}^{(i)} \subset \mathbb{R}^h$  with H-representation  $(P_A^{(i)}, P_B^{(i)})$ ,  $i = 1, \dots, k$ , is a polyhedron  $\mathcal{P}$  with H-representation  $(P_A, P_B)$ , where

$$P_A = \begin{bmatrix} P_A^{(1)} \\ \vdots \\ P_A^{(k)} \end{bmatrix} \quad P_B = \begin{bmatrix} P_B^{(1)} \\ \vdots \\ P_B^{(k)} \end{bmatrix}.$$

For computational reasons, redundant half-spaces should be removed from the resulting H-representation. A polytope is a (convex) closed and bounded polyhedron. Zonotopes are a special class of centrally symmetric convex polytopes.

**Definition 1** (Zonotope). *A convex polytope in  $\mathcal{Z} \in \mathbb{R}^h$  is called a zonotope if it can be written as*

$$\mathcal{Z} = \left\{ z \in \mathbb{R}^h \mid z = c + \sum_{i=1}^r \alpha_i g_i, \alpha_i \in [-1, 1] \right\},$$

where  $c \in \mathbb{R}^h$  is the center,  $g_i \in \mathbb{R}^h$ ,  $i \in \{1, \dots, r\}$ , are called generators, and  $\frac{r}{h}$  is the order of the zonotope. We use  $\langle c, g_1, \dots, g_r \rangle$  as an interchangeable, more concise notation for  $\mathcal{Z}$ . The generators can also be combined into the generator matrix  $G \in \mathbb{R}^{h \times r}$ , which contains the generators as its columns, this leading to  $\langle c, G \rangle$  as an alternative notation for  $\mathcal{Z}$ . Correspondingly, the  $\alpha_i$  coefficients are collected in a vector  $\alpha$ .

A parallelopete is a zonotope of order 1 with  $h$  linearly independent generators. Note that in a parallelopete, every element  $z \in \mathcal{Z} = \langle c, G \rangle$  is uniquely defined by the vector

$$\alpha(z) = G^{-1}(z - c) \quad (1)$$

because  $G$  is invertible.

## 3. Problem Formulation

We consider a discrete time PWA system whose state  $x \in \mathbb{R}^n$  evolves according to the following equation:

$$x(k+1) = \begin{cases} A^{(1)}x(k) + B_u^{(1)}u(k) + B_w^{(1)}w(k) + f^{(1)}, & x(k) \in \mathcal{M}^{(1)} \\ \vdots \\ A^{(s)}x(k) + B_u^{(s)}u(k) + B_w^{(s)}w(k) + f^{(s)}, & x(k) \in \mathcal{M}^{(s)} \end{cases} \quad (2)$$

where the control input  $u \in \mathbb{R}^m$  takes values in the convex polytope  $\mathcal{U} = (U_A, U_B) \subset \mathbb{R}^m$  and the disturbance  $w \in \mathbb{R}^p$  takes values in the zonotope  $\mathcal{W} = \langle c_w, G_w \rangle \subset \mathbb{R}^p$ , respectively. The initial state  $x(0)$  is uncertain and takes values in the zonotope  $\mathcal{X}_0 = \langle c_{x,0}, G_{x,0} \rangle \subseteq \mathbb{R}^n$ . If the initial set

or disturbance set is given as a convex polytope instead, we can always over-approximate these sets with zonotopes and our methods still work. Note that no extra conservatism is added in the most common case when  $\mathcal{X}_0$  and  $\mathcal{W}$  are boxes.

The system dynamics in each set  $\mathcal{M}^{(i)}$  representing mode  $i$  is defined by the matrices  $A^{(i)}$ ,  $B_u^{(i)}$ ,  $B_w^{(i)}$  and vector  $f^{(i)}$ . We therefore distinguish between the continuous values of state  $x(k)$ , which evolves in discrete time, and the discrete mode  $i$ . We assume that each set  $\mathcal{M}^{(i)} \subset \mathbb{R}^n$  is a polyhedron with H-representation  $(M_A^{(i)}, M_B^{(i)})$ , where  $M_A^{(i)} \in \mathbb{R}^{q^{(i)} \times n}$  and  $M_B^{(i)} \in \mathbb{R}^{q^{(i)}}$ . We refer to the half-spaces defining the boundary of  $\mathcal{M}^{(i)}$  as the mode boundary.

System (2) is said to be *well posed* if the collection of modes  $\{\mathcal{M}^{(i)}\}_{i=1}^s := \{\mathcal{M}^{(1)}, \dots, \mathcal{M}^{(s)}\}$  forms a *polyhedral subdivision* of the state space, that is if  $\cup_{i=1}^s \mathcal{M}^{(i)} = \mathbb{R}^n$ , each  $\mathcal{M}^{(i)}$  is of dimension  $n$ , and the intersection  $\mathcal{M}^{(i)} \cap \mathcal{M}^{(j)}$ ,  $i \neq j$ , is either empty or a common proper face of both polyhedra. We assume that the PWA system we are dealing with is well posed.

We address finite-horizon control of system (2) over  $[0, N]$ . Specifically, we assume that the system is subject to time-varying state constraints that are given in terms of the following polyhedral constraints on the state:

$$x(k) \in \mathcal{S}_k = \{x \in \mathbb{R}^n \mid S_A(k)x \leq S_B(k)\}, \quad k = 1, \dots, N. \quad (3)$$

Our aim is to design a controller that robustly guarantees safety while minimizing the distance from a given terminal state  $x_f$  at time  $N$ , subject to the actuation constraint  $u(k) \in \mathcal{U}$ ,  $k = 0, \dots, N-1$ . For this purpose we consider a static state feedback control law

$$u(k) = \mu_k(x(k)), \quad k = 0, \dots, N-1. \quad (4)$$

Let  $\rho : \mathbb{R}^n \rightarrow \{1, \dots, s\}$  be the map that associates the number of the corresponding active mode to  $x \in \mathbb{R}^n$ , i.e.,  $\rho(x) = i \Leftrightarrow x \in \mathcal{M}^{(i)}$ . Then, the reachable sets can be recursively defined as

$$\mathcal{R}_{k+1}^\mu = \left\{ x^+ \in \mathbb{R}^n \mid \exists x \in \mathcal{R}_k, \exists w \in \mathcal{W} \text{ such that } x^+ = A^{(\rho(x))}x + B_u^{(\rho(x))}\mu_k(x) + B_w^{(\rho(x))}w + f^{(\rho(x))} \right\},$$

initialized with  $\mathcal{R}_0 = \mathcal{X}_0$ . Our control design problem can then be expressed as the following optimization problem:

$$\begin{aligned} \min_{\{\mu_k(\cdot)\}_{k=0}^{N-1}} \|\mathcal{R}_N^\mu - x_f\|_1 \quad \text{subject to:} \quad (5) \\ \mathcal{R}_k^\mu \subseteq \mathcal{S}_k, \quad k = 1, \dots, N, \\ \forall x \in \mathcal{R}_k^\mu : \mu_k(x) \in \mathcal{U}, \quad k = 0, \dots, N-1, \end{aligned}$$

where  $\|\mathcal{R} - x_f\|_1 = \max_{x \in \mathcal{R}} \|x - x_f\|_1$  is the maximum 1-norm to the desired final state  $x_f$  of any state in the set  $\mathcal{R} \subset \mathbb{R}^n$ .

In addition to its numerical advantages, the reason for using the  $\ell_1$  norm for our cost function is that many target sets are axis-aligned boxes. Therefore, we prefer a set whose bounding axis-aligned box is as small as possible. When using the  $\ell_1$  norm, we minimize the sum of the side lengths of this bounding box. If we used, for example, the  $\ell_2$  norm instead, we would minimize the sum of the squared sides, therefore not appropriately considering smaller dimensions. The extreme case would be using the infinity norm, which would result in only minimizing the largest side of the bounding box and thereby neglecting the other dimensions. In the next section, we present an overview of our approach to solving (5).

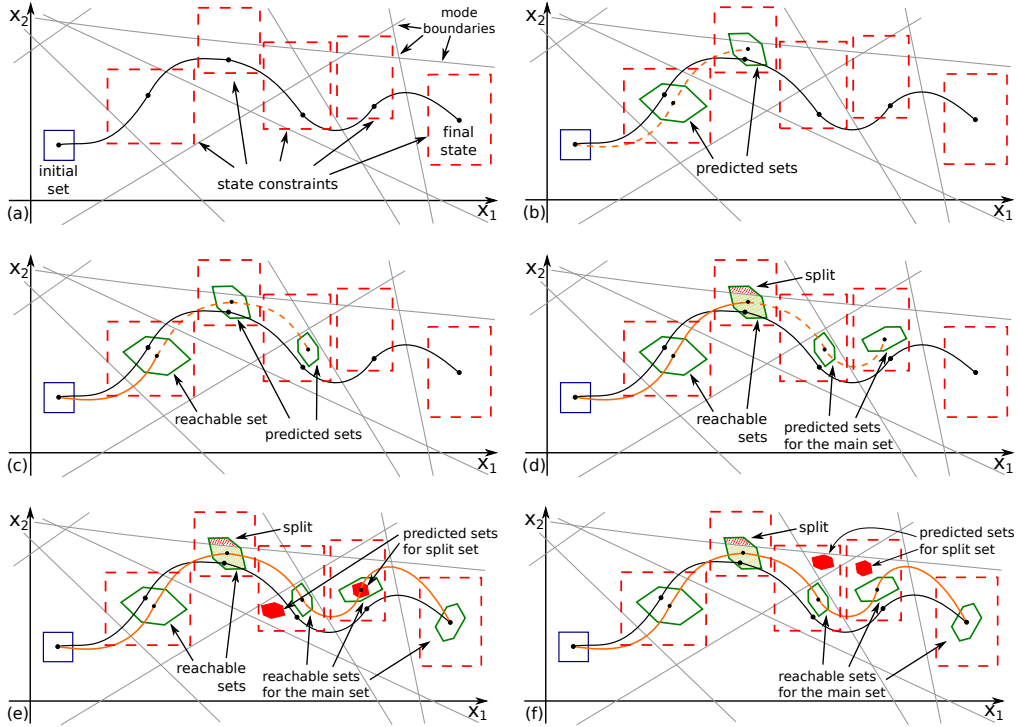


Figure 2: Overview of the overall approach with prediction horizon  $M = 2$  and horizon after splits  $M_{split} = 2$ . (a) Computation of the reference trajectory and main switching sequence  $\mathcal{M}_0, \dots, \mathcal{M}_N$ . (b) Computation of the first  $M$  predicted reachable sets  $\mathcal{R}_1, \dots, \mathcal{R}_M$  starting from the initial set. (c) Application of the first control law and computation of a new collection of  $M$  sets  $\mathcal{R}_2, \dots, \mathcal{R}_{M+1}$ , starting from  $\mathcal{R}_1$ . The set  $\mathcal{R}_1$  has now been reached (orange solid line). (d) Application of the first control law and computation of  $M$  new sets  $\mathcal{R}_3, \dots, \mathcal{R}_{M+2}$  starting from  $\mathcal{R}_2 \cap \mathcal{M}_2$ . (e) Case I: The split set is steered back inside the main reachable set in  $M_{split}$  steps. No branching is needed. (f) Case II: The split set is not steered back inside the main reachable set in  $M_{split}$  steps: it will be used as a new initial set.

#### 4. Overview of the Approach

Let us introduce our approach based on Fig. 2, which illustrates the following steps:

**Step 1. Compute reference trajectory (Fig. 2(a)).** We start by computing a reference trajectory for the *nominal* PWA system (i.e., without uncertainties, see Sec. 5), which at time  $N$  should be close to the desired final state  $x_f$  while satisfying the input and state constraints as well as considering the distance to the mode boundaries to avoid splitting of reachable sets. The sequence of modes  $\mathcal{M}_0, \dots, \mathcal{M}_N$  followed by the reference trajectory is called the *main mode sequence*. Each  $\mathcal{M}_k$  takes values in the set of the PWA system modes  $\{\mathcal{M}^{(i)}\}_{i=1}^s$ .

**Step 2. Compute reachable sets along mode sequence (Fig. 2(b)).** At each iteration  $k$ , we design a control law for a prediction horizon of  $M$  steps ahead, which steers all the states of the reachable sets close to the corresponding reference trajectory. We minimize the size of reachable sets and weight their distance to the boundaries of the main mode sequence so that, where possible, splits are avoided. At the same time, the controller ensures that the input and state constraints are satisfied despite the presence of disturbances. The result of this optimization is a sequence of reachable sets  $\mathcal{R}_{k+1}, \dots, \mathcal{R}_{k+M}$  together with the corresponding time-varying control

law.

**Step 3. Apply the first input (Fig. 2(c)).** At each iteration, we only retain the control law at time  $k$  and the reachable set  $\mathcal{R}_{k+1}$ . We only use the portion of the reachable set that lies inside the main mode as an initial set for each iteration, i.e.,  $\mathcal{R}_k^* = \mathcal{R}_k \cap \mathcal{M}_k$ . We repeat the procedure of Steps 2 and 3 for all  $k$ 's until we reach the final time step. A sequence of sets  $\mathcal{R}_1^*, \dots, \mathcal{R}_N^*$  is determined that is called the *main reachable set sequence*.

**Step 4. Check for splitting (Fig. 2(d)).** If the main reachable set is completely inside the modes of the reference trajectory (no split occurred) and the state and input constraint have been satisfied, the problem is solved. If not, we start from the first time step  $k^0$  where a split has been detected ( $k^0 = 2$  in Fig. 2(d)).

**Step 5a. Easy recovery possible (Fig. 2(e)).** If we can steer the sets  $\mathcal{R}_{k^0} \cap \mathcal{M}^{(i)} \neq \emptyset$  with  $\mathcal{M}^{(i)} \neq \mathcal{M}_{k^0}$  (i.e., those parts of the main reachable set outside of the main mode sequence) back to the main reachable set in  $M_{split}$  time steps, we only have to consider the main reachable set, as it contains the split set.

**Step 5b. Easy recovery not possible (Fig. 2(f)).** If Step 5a is not possible, we branch the split set and treat it as a new initial set for which we compute a new reference trajectory (computed as in Step 1) for the residual time horizon  $[k^0, N]$ . It is also necessary to recompute the reachable sets along the mode sequence at Step 2 starting from  $\mathcal{R}_{k^0} \cap \mathcal{M}_{k^0}$  at time  $k^0$ .

Next, we explain the algorithms for the reference trajectory computation and the set-based control design.

## 5. Reference Trajectory Computation

We now explain how to build the main reference trajectory (Fig. 2(a)) over the time horizon  $[0, N]$ ; the computation for  $[k^0, N]$  when splitting occurs (Fig. 2(f)) is done analogously. We consider the PWA system operating in nominal conditions, i.e., disturbance is  $w(k) = c_w$ ,  $k = 0, \dots, N-1$ , and the initial state is  $x(0) = c_{x,0}$ , with  $c_w$  and  $c_{x,0}$  denoting the centers of the disturbance set and the initial set, respectively. We design a trajectory that satisfies the actuation constraint  $u(k) \in \mathcal{U}$ ,  $k = 0, \dots, N-1$ , and the state constraints (3), while getting as close as possible to the desired final state  $x_f$  and while keeping the state evolution as distant as possible from the mode and state constraints boundaries for each time step.

The computation of the reference trajectory is formulated as a Mixed Integer Linear Program (MILP) and makes use of the following equivalent Mixed Logical Dynamical (MLD) form of the (nominal) PWA system (2):

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_\delta \delta(k) + B_z z(k) + B_w c_w, \\ E_x x(k) + E_u u(k) + E_\delta \delta(k) + E_z z(k) &\leq E_{aff}, \end{aligned} \quad (6)$$

where  $\delta(k) \in [0, 1]^{n_\delta}$  is a binary vector that determines which of the  $s$  modes is active at time  $k$  and  $z(k) \in \mathbb{R}^{n_z}$  is an auxiliary real-valued variable (see [1] for details). Each mode  $\mathcal{M}^{(i)}$  in (2) is encoded via a binary vector  $\delta^{(i)}$ , so that  $\delta(k) = \delta^{(i)}$  if and only if  $x(k) \in \mathcal{M}^{(i)}$ . Linear actuation constraints are encoded in the MLD representation via  $E_u$  and  $E_{aff}$ . The idea behind the MLD reformulation can be understood as follows: inequalities in (6) express in a compact form the different dynamics of the original PWA system (2) and, via *big-M* technique, can be structured so that when  $\delta(k) = \delta^{(i)}$  they collapse in  $z(k) = A^{(i)}x(k) + B_u^{(i)}u(k) + f^{(i)}$  and  $M_A^{(i)}x(k) \leq M_B^{(i)}$ . The original PWA dynamics can then be retrieved by simply choosing  $A = 0$ ,  $B_u = 0$ ,  $B_\delta = 0$ ,  $B_w = 0$  and  $B_z = I$  in (6).





imposing that  $x(N)$  gets close to the desired final state  $x_f$  according to the 1-norm:

$$\max_{u(1), \dots, u(N-1)} \sum_{k=0}^{N-1} \sum_{i=1}^s d^{(i)}(k) - \gamma_f \|x(N) - x_f\|_1$$

subject to: (11)

$$\left\{ \begin{array}{l} V(k) \begin{bmatrix} x(k) \\ d^{(1)}(k) \\ \vdots \\ d^{(s)}(k) \\ \delta(k) \end{bmatrix} \leq R(k), k = 1, \dots, N, \\ \text{MLD dynamics (6), } k = 0, \dots, N-1, \text{ with } x(0) = c_{x,0}, \\ \text{state constraints (3),} \end{array} \right.$$

with  $V(k)$  and  $R(k)$  as

$$V = \begin{bmatrix} \begin{bmatrix} P_A^{(1)} \\ 0_{2n_s, n} \end{bmatrix} & H^{(1)} & 0 & D^{(1)} \\ \vdots & \ddots & \vdots & \vdots \\ \begin{bmatrix} P_A^{(s)} \\ 0_{2n_s, n} \end{bmatrix} & 0 & H^{(s)} & D^{(s)} \end{bmatrix}, R = \begin{bmatrix} R^{(1)} \\ R^{(2)} \\ \vdots \\ R^{(s)} \end{bmatrix}$$

with

$$H^{(i)} = \begin{bmatrix} \|P_{A,1}^{(i)}\|_2 \\ \|P_{A,2}^{(i)}\|_2 \\ \vdots \\ \|P_{A,J}^{(i)}\|_2 \\ 1 \\ -1 \\ 1 \\ -1 \\ \vdots \\ 1 \\ -1 \end{bmatrix}, \quad R^{(i)} = \begin{bmatrix} P_{B,1}^{(i)} + \Gamma \|\delta^{(i)}\| \\ P_{B,2}^{(i)} + \Gamma \|\delta^{(i)}\| \\ \vdots \\ P_{B,J}^{(i)} + \Gamma \|\delta^{(i)}\| \\ \Gamma - \Gamma \delta_1^{(i)} \\ \Gamma - \Gamma \delta_1^{(i)} \\ \Gamma - \Gamma \delta_2^{(i)} \\ \Gamma - \Gamma \delta_2^{(i)} \\ \vdots \\ \Gamma - \Gamma \delta_{n_s}^{(i)} \\ \Gamma - \Gamma \delta_{n_s}^{(i)} \end{bmatrix},$$

and

$$D^{(i)} = \begin{bmatrix} & \Gamma(2\delta^{(i)} - \mathbf{1})^T & & & \\ & \Gamma(2\delta^{(i)} - \mathbf{1})^T & & & \\ & \vdots & & & \\ & \Gamma(2\delta^{(i)} - \mathbf{1})^T & & & \\ \Gamma(1 - 2\delta_1^{(i)}) & 0 & \dots & 0 & \\ \Gamma(1 - 2\delta_1^{(i)}) & 0 & \dots & 0 & \\ 0 & \Gamma(1 - 2\delta_2^{(i)}) & \dots & 0 & \\ 0 & \Gamma(1 - 2\delta_2^{(i)}) & \dots & 0 & \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \dots & \Gamma(1 - 2\delta_{n_\delta}^{(i)}) & \\ 0 & 0 & \dots & \Gamma(1 - 2\delta_{n_\delta}^{(i)}) & \end{bmatrix}.$$

Since the 1-norm can be rewritten as a linear term by introducing auxiliary variables and linear constraints [69, 70], problem (11) is a MILP.

We shall denote the reference trajectory that results from solving the MILP problem (11) by  $x_{ref}(k)$ ,  $k = 0, \dots, N$ , where  $x_{ref}(0) = c_{x,0}$ . Note that to compute the reference trajectory we are not adding extra binary variables other than the  $\delta$  variables that are already embedded in the MLD formulation to represent the modes structure.

**Remark 1.** *It is always possible to minimize the use of the control inputs in (11), e.g., for saving energy, by adding a term of the form  $-||u(k)||_l$ , for each time step, in the cost function. Choosing  $l = 2$  would lead to a Mixed Integer Quadratic Program, whereas choosing  $l = 1$  would still lead to a MILP.*

## 6. Set-Based Control Design

In this section, we address the synthesis of the state feedback control law (4) (Fig. 2(b)–(d)). The idea is to operate according to a receding horizon strategy, i.e., we design the control input  $u(h+k) = \bar{\mu}_k(x)$ ,  $\bar{\mu}_k(\cdot) : \mathcal{Z}_x(h) \rightarrow \mathbb{R}^m$ ,  $k = 0, \dots, M-1$  that makes the system robustly track the reference trajectory along the look-ahead time-horizon  $[h, h+M]$ , starting from  $x \in \mathcal{Z}_x(h)$  at time  $h$ , and then set

$$\mu_h(\cdot) = \bar{\mu}_0(\cdot). \quad (12)$$

Let us now consider the computation of the control law for one time instance. For a simpler notation and without loss of generality, we consider the computation at time  $h = 0$ . We assume that the zonotope  $\mathcal{Z}_x(0) = \langle c_x, g_{x,1}, \dots, g_{x,r_x} \rangle$  of possible values for  $x(0)$  belongs to a single mode.

Let us start by expressing the evolution of a single state  $x(0) \in \mathcal{Z}_x(0)$  as a function of the control input, based on our assumption that  $x(k) \subseteq \mathcal{M}^{\rho(c_x(k))}$ ,  $k = 0, \dots, M$ , i.e., it stays in the main mode sequence. For ease of notation, we denote by  $A_l, B_{u,l}, B_{w,l}, f_l$  matrices and vectors active in the mode  $\rho(x_{ref}(l))$ . Now, we can write the trajectory starting from  $x(0) \in \mathcal{Z}_x(0)$  at time  $k$  as:

$$x(k) = \bar{A}x(0) + \sum_{j=0}^{k-1} (\bar{B}_{u,j}u(j) + \bar{B}_{w,j}w(j) + \bar{A}_{f,j}f_j) \quad (13)$$

for  $k = 1, \dots, M$ , where we use the shorthand notations:

$$\begin{aligned}\bar{A} &= A_{k-1} \dots A_0, \\ \bar{A}_{f,j} &= A_{k-1} \dots A_{j+1}, \quad j = 0, \dots, k-2,\end{aligned}$$

with  $\bar{A}_{f,k-1} = I$ , i.e., the identity matrix,

$$\bar{B}_{u,j} = A_{k-1} \dots A_{j+1} B_{u,j}, \quad j = 0, \dots, k-2,$$

with  $\bar{B}_{u,k-1} = B_{u,k-1}$ , and  $\bar{B}_{w,j}$  being defined analogously to  $\bar{B}_{u,j}$ . Using the zonotope representation of the initial set  $\mathcal{Z}_x(0) = \langle c_x, g_{x,1}, \dots, g_{x,r_x} \rangle$  and of the disturbance set  $\mathcal{Z}_w = \langle c_w, g_{w,1}, \dots, g_{w,r_w} \rangle$ , we can express (13) as

$$x(k) = \bar{A} \left( c_x + \sum_{i=1}^{r_x} \alpha_i(x(0)) g_{x,i} \right) + \sum_{j=0}^{k-1} \left( \bar{B}_{u,j} u(j) + \bar{B}_{w,j} \left( c_w + \sum_{i=1}^{r_w} \beta_i(j) g_{w,i} \right) + \bar{A}_{f,j} f_j \right), \quad (14)$$

where  $\alpha_i(x(0))$  is such that  $x(0) = c_x + \sum_{i=1}^{r_x} \alpha_i(x(0)) g_{x,i}$  and  $\beta_i(j)$  is such that  $w(j) = c_w + \sum_{i=1}^{r_w} \beta_i(j) g_{w,i}$ . The structure of zonotopes as a superposition of generators for the initial state and disturbances allows us to express the evolution of any state as a weighted combination of the generator vectors. Therefore, if we choose a similar structure for the control input of the auxiliary finite horizon problem, we are able to control every trajectory starting from  $\mathcal{Z}_x(0)$  for any disturbance trajectory by simply finding input trajectories for the generators and interpolating between them:

$$u(\alpha, \beta(\cdot), j) = u_c(j) + \sum_{i=1}^{r_x} \alpha_i(x(0)) u_{x,i}(j) + \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} \beta_i(l) u_{w,i,l}(j). \quad (15)$$

We find the input trajectories  $u_c(\cdot), u_{x,1}(\cdot), \dots, u_{x,r_x}(\cdot)$  and  $u_{w,1}(\cdot), \dots, u_{w,r_w}(\cdot)$  along the finite horizon  $[0, M-1]$  by solving an optimization problem, where we use  $(\cdot)$  to refer to the whole trajectory. In the receding horizon implementation, only the first sample of this input trajectory is applied, which depends only on the initial state and involves only  $u_c(0)$  and  $u_{x,1}(0), \dots, u_{x,r_x}(0)$ . When computing the input trajectory, an upper bound on the distance of the reach set from the reference trajectory is minimized while taking the constraints into account. The upper bound for the distance of any state in the reachable set to the reference trajectory can be obtained as follows:

**Theorem 1.** *Let  $\mathcal{Z}_x(k)$  denote the reachable set resulting from control law (15) at time  $k$ , then the following inequality holds:*

$$\begin{aligned}& \left\| \mathcal{Z}_x(k) - x_{ref}(k) \right\|_1 \\ & \leq \left\| \bar{A} c_x - x_{ref}(k) + \sum_{j=0}^{k-1} \left( \bar{A}_{f,j} f_j + \bar{B}_{w,j} c_w + \bar{B}_{u,j} u_c(j) \right) \right\|_1 + \sum_{i=1}^{r_x} \left\| \bar{A} g_{x,i} + \sum_{j=0}^{k-1} \bar{B}_{u,j} u_{x,i}(j) \right\|_1 \\ & \quad + \sum_{j=0}^{k-1} \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} \left\| \bar{B}_{w,j} g_{w,i} + \bar{B}_{u,j} u_{w,i,l}(j) \right\|_1.\end{aligned} \quad (16)$$

*Proof.*

$$\begin{aligned}
& \|\mathcal{Z}_x(k) - x_{ref}(k)\|_1 \\
&= \max_{x \in \mathcal{Z}_x(k)} \|x - x_{ref}(k)\|_1 \\
&\stackrel{(14)}{=} \max_{\alpha_i, \beta_i(\cdot) \in [-1, 1]} \left\| \bar{A} \left( c_x + \sum_{i=1}^{r_x} \alpha_i(x(0)) g_{x,i} \right) \right. \\
&\quad \left. + \sum_{j=0}^{k-1} \left( \bar{B}_{u,j} u(\alpha, \beta(\cdot), j) + \bar{B}_{w,j} \left( c_w + \sum_{i=1}^{r_w} \beta_i(j) g_{w,i} \right) + \bar{A}_{f,j} f_j \right) - x_{ref}(k) \right\|_1 \\
&\stackrel{(15)}{=} \max_{\alpha_i, \beta_i(\cdot) \in [-1, 1]} \left\| \bar{A} c_x + \sum_{j=0}^{k-1} \bar{B}_{u,j} u_c(j) + \sum_{i=1}^{r_x} \left( \bar{A} \alpha_i(x(0)) g_{x,i} + \sum_{j=0}^{k-1} \bar{B}_{u,j} \alpha_i(x(0)) u_{x,i}(j) \right) \right. \\
&\quad \left. + \sum_{j=0}^{k-1} \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} \left( \bar{B}_{w,j} \beta_i(l) g_{w,i} + \bar{B}_{u,j} \beta_i(l) u_{w,i,l}(j) \right) + \sum_{j=0}^{k-1} \left( \bar{B}_{w,j} c_w + \bar{A}_{f,j} f_j \right) - x_{ref}(k) \right\|_1 \\
&\leq \max_{\alpha_i, \beta_i(\cdot) \in [-1, 1]} \left\| \bar{A} c_x - x_{ref}(k) + \sum_{j=0}^{k-1} \left( \bar{A}_{f,j} f_j + \bar{B}_{w,j} c_w + \bar{B}_{u,j} u_c(j) \right) \right\|_1 \\
&\quad + \sum_{i=1}^{r_x} \left\| \alpha_i(x(0)) \left( \bar{A} g_{x,i} + \sum_{j=0}^{k-1} \bar{B}_{u,j} u_{x,i}(j) \right) \right\|_1 + \sum_{j=0}^{k-1} \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} \left\| \beta_i(l) \left( \bar{B}_{w,j} g_{w,i} + \bar{B}_{u,j} u_{w,i,l}(j) \right) \right\|_1 \\
&\leq \left\| \bar{A} c_x - x_{ref}(k) + \sum_{j=0}^{k-1} \left( \bar{A}_{f,j} f_j + \bar{B}_{w,j} c_w + \bar{B}_{u,j} u_c(j) \right) \right\|_1 + \sum_{i=1}^{r_x} \left\| \bar{A} g_{x,i} + \sum_{j=0}^{k-1} \bar{B}_{u,j} u_{x,i}(j) \right\|_1 \\
&\quad + \sum_{j=0}^{k-1} \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} \left\| \bar{B}_{w,j} g_{w,i} + \bar{B}_{u,j} u_{w,i,l}(j) \right\|_1
\end{aligned}$$

□

We use  $cost(\mathcal{Z}_x(k))$  to refer to the upper bound of the cost function (16). This result can be interpreted graphically: as shown in Fig. 3, we exploit the available control capabilities and steer the center of the initial set towards the reference state and the generators towards the origin, thereby aiming at bringing all states close to  $x_{ref}(k)$ . Using this upper bound allows us to solve the problem by only considering the generators of a zonotope, which scale linearly with the dimension (for a fixed zonotope order). Using other set representations, such as polytopes, would be computationally much more expensive, e.g., the number of extreme points scales exponentially.

In the next step, we must ensure that all possible trajectories remain inside the state constraint  $S_k$ . Checking if the reachable set in the form of a zonotope is inside some convex set can be performed efficiently using the following result:

**Theorem 2.** A zonotope  $\mathcal{Z} = \langle c, g^{(1)}, \dots, g^{(p)} \rangle$  satisfies convex constraints of the form  $S = \{x \in \mathbb{R}^n | S_{Ax} \leq S_B\}$  if

$$S_{AC} + \sum_{i=1}^p |S_A g^{(i)}| \leq S_B, \quad (17)$$

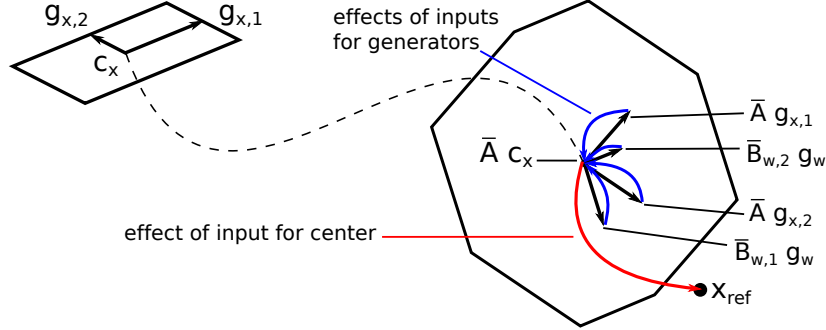


Figure 3: We obtain a small reachable set close to  $x_{ref}(k)$  by steering the center of the initial set to  $x_{ref}(k)$ , and the generators of the initial set and those resulting from disturbances to the origin.

where the absolute value and less than or equal operators are both performed element-wise.

*Proof.* Using the zonotope representation results in

$$S_A x \leq S_B, \forall x \in \mathcal{S} \Leftrightarrow S_{AC} + \sum_{i=1}^p \alpha_i S_A g^{(i)} \leq S_B, \forall \alpha_i \in [-1, 1].$$

The left side of the above inequality can be further bounded by

$$S_{AC} + \sum_{i=1}^p \alpha_i S_A g^{(i)} \leq S_{AC} + \sum_{i=1}^p |\alpha_i| S_A g^{(i)} \leq S_{AC} + \sum_{i=1}^p \underbrace{|\alpha_i|}_{\leq 1} |S_A g^{(i)}| \leq S_{AC} + \sum_{i=1}^p |S_A g^{(i)}|.$$

In fact, there exists an  $\alpha$  with  $\alpha_i = \pm 1$ , such that the computed bound is touched.  $\square$

We must also ensure that the sum of all possible inputs for the center and generators does not exceed the input bounds:

**Corollary 1.** *The zonotope control law  $u(\alpha, \beta(\cdot), j)$  from (15) satisfies the input constraint  $\mathcal{U} = \{u \in \mathbb{R}^m | U_A u \leq U_B\}$  if and only if*

$$U_A u_c(j) + \sum_{i=1}^{r_x} |U_A u_{x,i}(j)| + \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} |U_A u_{w,i,l}(j)| \leq U_B, \forall j \in \{0, \dots, M-1\}. \quad (18)$$

*Proof.* Using the same proof concept as in Theorem 2, it follows that

$$\begin{aligned} u_c(j) + \sum_{i=1}^{r_x} \alpha_i(x(0)) u_{x,i}(j) + \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} \beta_i(l) u_{w,i,l}(j) &\in \mathcal{U}, \\ \forall \alpha &\in [-1, 1]^{r_x}, \beta \in [-1, 1]^{r_w}, \forall j \in \{0, \dots, M-1\} \\ \Leftrightarrow U_A u_c(j) + \sum_{i=1}^{r_x} |U_A u_{x,i}(j)| + \sum_{l=0}^{j-1} \sum_{i=1}^{r_w} |U_A u_{w,i,l}(j)| &\leq U_B, \forall j \in \{0, \dots, M-1\}. \end{aligned}$$

$\square$

Lastly, we also wish to stay away from the mode boundaries to minimize the number of splits. To achieve this, we weight the minimal distance of the reachable sets to the mode boundaries, similarly to what we do in the optimization of the reference trajectory in (9):

$$\text{cost}(d) := - \sum_{k=1}^M d(k) \quad (19)$$

subject to

$$\|M_{A,j}(k)\|_2 d(k) \leq M_{A,j}(k)c_{\mathcal{Z}}(k) + \sum_{i=1}^{r_{\mathcal{Z}}(k)} |M_{A,j}(k)g_{\mathcal{Z},i}(k)| - M_{B,j}(k), \quad (20)$$

where  $M_{A,j}(k)$  and  $M_{B,j}(k)$  represent the  $j$ -th row of the matrices defining the half-space representation of the mode in the main switching sequence at time  $k$  and  $c_{\mathcal{Z}}(k), g_{\mathcal{Z},1}(k), \dots, g_{\mathcal{Z},r_{\mathcal{Z}}(k)}(k)$  denote the center and generators of the reachable sets  $\mathcal{Z}_k$ . Since additional generators from the disturbance effects are added in every time step, the order  $r_{\mathcal{Z}}(k)$  of the reachable sets increases over time.

Using (19) together with the results from Theorem 1, Theorem 2, and Corollary 1, allows us to express everything in a single optimization problem:

$$\begin{aligned} \min_{u_c(\cdot), u_x(\cdot), u_w(\cdot)} \sum_{k=1}^M \text{cost}(\mathcal{Z}_x(k)) + \gamma_d \text{cost}(d) \\ \text{s.t. } \mathcal{Z}_k \subseteq \mathcal{S}_k \text{ using (17), (18), (20), } k = 1, \dots, M, \end{aligned} \quad (21)$$

where  $\gamma_d \in \mathbb{R}_0^+$  is a weight used to adjust the importance of being close to the reference trajectory relative to being far away from the mode boundaries. It is shown in [69, Ch. 6] and [70] that the norm and absolute value can be transformed to linear constraints and cost functions. Therefore, this problem can be solved efficiently in a single linear program (LP).

Problem (21) provides the inputs  $u_c^*(\cdot)$  and  $u_x^*(\cdot)$ , which can be used to compute

i) the feedback law (4) at time  $k$  via (12):

$$\mu_k(x) = u_c^*(k) + \sum_{i=1}^{r_x} \alpha_i(x) u_{x,i}^*(k), \quad (22)$$

for  $x \in \mathcal{Z}_x(k) = \langle c_x, g_{x,1}, \dots, g_{x,r_x} \rangle$ , where  $\alpha_i(x), i = 1, \dots, r_x$ , are such that

$$x = c_x + \sum_{i=1}^{r_x} \alpha_i(x) g_{x,i}, \quad (23)$$

ii) the reachable set  $\mathcal{R}_{k+1}^* = \mathcal{Z}_x^*(k+1) \cap \mathcal{M}_{k+1}$ , where  $\mathcal{M}_{k+1}$  is the mode to which  $x_{ref}(k+1)$  belongs (see Step 2 in Section 4) where  $\mathcal{Z}_x^*(k+1)$  is the reachable set obtained by applying  $\mu_k(\cdot)$  at time  $k$  starting from  $\mathcal{Z}_x(k)$ .

### *Splits and Recovery*

The procedure described above is iteratively repeated for  $N$  time steps (with a shrinking horizon of  $N - k$  for the last  $M$  iterations). Afterwards, we check for each split that has occurred (Fig. 2(d)) whether the split set can be steered back into the main sequence or not (Figs. 2(e) and (f), respectively). We do this by solving optimization problem (21) for a time horizon of  $M_{split}$ , starting at the time step  $k^0$  when the split occurred and with the additional constraints  $d(k) \geq 0, k = 1, \dots, M_{split}$  and  $\mathcal{R}_{k^0+M_{split}}^{split} \subseteq \mathcal{R}_{k^0+M_{split}}^*$ . These additional constraints ensure that the reachable set of the split set is steered back into the main switching sequence after one time step and completely stays inside this mode sequence until it is a subset of the main reachable set  $\mathcal{R}_{k^0+M_{split}}^*$ . Without these constraints, we would have to consider the combinatorial problem of all possible modes the set could enter and therefore different dynamics which could be applied. The second constraint  $\mathcal{R}_{k^0+M_{split}}^{split} \subseteq \mathcal{R}_{k^0+M_{split}}^*$  can be easily checked using the formula from Theorem 2. Note that if  $k^0 + M_{split} < N$ , we use the horizon  $N - k^0$  instead.

If the optimization problem is not feasible, we cannot steer the reachable set back in  $M_{split}$  steps along the main switching sequence. In this case, we start a new optimization problem with a new reference trajectory and with a reduced time horizon of  $N - k^0$  steps.

### *Computational Complexity of the Algorithm*

The algorithm described in Section 4 solves a MILP every time we compute a reference trajectory, for which efficient solvers exist (see, e.g., CPLEX); nonetheless, the complexity of a MILP is in the worst case exponential in the number of binary variables. In our case, the binary variables are represented by the vector  $\delta(k)$  for each time step (indeed, the extra requirement of staying far from mode boundaries does not require the addition of extra binary variables), and therefore their number increases linearly with the problem horizon and also linearly with the number of modes. The dimension of states does not directly influence the number of binary variables; however, higher dimensional systems often have more modes. To reduce the computational effort, since the state constraints have to be satisfied at each time step, it is possible to prune the exploration tree by removing any switching sequence that contains modes that do not intersect the state constraint sets at some time step.

For the set-based controller design, if all reachable sets stay within the main mode sequence, we have to solve  $N$  linear programs whose variables scale linearly in the prediction horizon  $M$  and linearly in the state and input dimension, if we fix the zonotope order. This can be done by applying zonotope reduction methods [71] which reduce the number of generators in the considered zonotopes to lower the computational effort.

Note that we cannot give a fixed bound on the number of splits; however, we expect that splits do not occur too frequently in practical examples, especially because we try to avoid them by maximizing the distance to the mode boundaries. If the split set can be steered back in the main mode after  $M_{split}$  time steps, then this adds only a single linear program to the computational effort. If we cannot steer it back in  $M_{split}$  time steps, then a new reference trajectory has to be computed, which requires solving a new MILP.

The computation of the reference trajectory is the main bottleneck, as the overall complexity of the proposed algorithm is governed by the complexity of the MILP. This is because the computational effort for solving a MILP scales exponentially with the number of binary variables appearing in it, and in our case, this number grows linearly with the time horizon. On the other hand, the computational effort for solving linear programs only scales polynomially (see [72]) in the size of the problem; moreover, the size of our LPs is smaller than the size of the reference



trajectory computation problem (see (21)). Therefore, if we denote the number of splits which cannot be steered back by  $S$ , the overall complexity is  $(S + 1)O(\text{MILP})$ .

#### *Obtaining the Online Control Law*

The solution of the MILP for the reference trajectory and all the following computations can be computationally demanding and therefore are performed offline, as it is hard to guarantee finding a feasible solution fast enough online. As the outcome of the overall offline control design procedure described in this section, we obtain a collection of zonotopes per time step representing the reachable set (possibly split over different modes and over-approximated) and the associated set-based controllers. Both of them are finitely parameterized and have a low space complexity (which scales linearly with the number of states, time steps, and splits). Once the value  $x$  of the state at time  $k$  is available, it is possible to determine the input  $\mu_k(x)$  using (22). Several possibilities exist for obtaining the parameters  $\alpha_i$  in (22): the easiest way is by solving the system of inequalities (23). Since this is a linear feasibility program, it can be solved fast online.

If faster computation times are required, then we can take advantage of the fact that parameters for expressing a state in a polytope can be computed in a closed-form expression [73]. By interpolating the  $\alpha$  values of the extreme points of our zonotope, we can use the same technique to obtain them through a closed-form expression. In the case of a parallelotope, we can obtain the  $\alpha$  values by simply inverting the generator matrix, see (1). This allows us to obtain linear controllers, which work analogously to the disturbance feedback controllers discussed in [74].

## 7. Numerical Example

We demonstrate the effectiveness of our approach on the quadruple tank benchmark described in [75]. The system is nonlinear with 4 states  $h_1, \dots, h_4$  and 2 control inputs  $v_1, v_2$ , that respectively denote the tank levels and the voltage inputs to the two pumps. The equations of the systems are

$$\begin{aligned} \dot{h}_1 &= -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1, \\ \dot{h}_2 &= -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2, \\ \dot{h}_3 &= -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2, \\ \dot{h}_4 &= -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1, \end{aligned}$$

where the values of the various parameters can be found in [75]. The states are constrained between 0 and 20, and the inputs are bounded between 0 and 12. To obtain a PWA system, we linearize the dynamics around different points and consider the approximation error as an additive disturbance that we want to counteract (one disturbance for each dynamics equation). The state space is therefore partitioned into 16 modes obtained by splitting each axis into the two intervals  $[0, 3]$  and  $(3, 20]$ . For each mode, a linear approximation is obtained by Taylor expansion around the centers of the intervals.

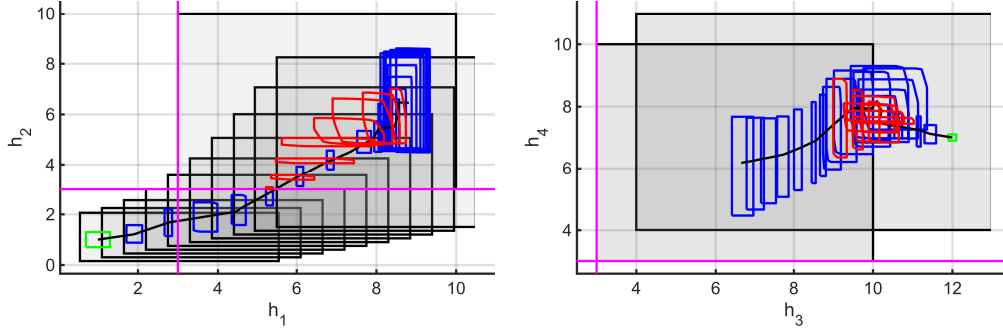


Figure 4: Results from numerical example: Reference trajectory (black), initial set (green), main reachable sets (blue), and state constraints (gray) at all time steps, projected on the  $(h_1, h_2)$  plane (left) and  $(h_3, h_4)$  plane (right). The mode edges are plotted in magenta. At time  $k = 5$ , a split set originates from an intersection with the edge  $h_2 = 3$  (in red). The corresponding reachable sets (in red) merge back with the main reachable set after  $M_{split} = 7$  steps.

We choose a sampling time of 1 second,  $N = 20$  for the time horizon, and  $M = 3$  for the prediction horizon. The time-varying state constraints are

$$\left. \begin{aligned} 0.55k \leq h_i(k) \leq 5 + 0.55k, & \quad i = 1 \\ 0.15k \leq h_i(k) \leq 2 + \frac{k^2}{16}, & \quad i = 2 \\ 4 \leq h_i(k) \leq 13, & \quad i = 3 \\ 4 \leq h_i(k) \leq 11, & \quad i = 4 \end{aligned} \right\} k = 1, \dots, 10$$

$$3 \leq h_i(k) \leq 10, \quad i = 1, \dots, 4 \quad \left. \vphantom{\begin{aligned} 0.55k \leq h_i(k) \leq 5 + 0.55k, \\ 0.15k \leq h_i(k) \leq 2 + \frac{k^2}{16}, \\ 4 \leq h_i(k) \leq 13, \\ 4 \leq h_i(k) \leq 11, \end{aligned}} \right\} k = 11, \dots, 20.$$

The initial set  $\mathcal{X}_0$  is given as a box with edges of lengths  $[0.6, 0.6, 0.2, 0.2]$  around its center  $x(0) = [1, 1, 12, 7]^T$  and the terminal state  $x_f$  is set to  $[9, 6.5, 6.5, 6.5]^T$ . The disturbances vary freely in a box of size  $\mathcal{W} = [-1, 2]^4$ , which contains the linearization errors as well as external disturbances. We model the disturbances acting on the system through the same input matrix as the inputs, i.e.,  $B_w^{(s)} = B_u^{(s)}$ ,  $s = 1, \dots, 16$ .

This example also demonstrates the idea of hybridization of nonlinear systems to obtain a PWA system. Note that due to the discrete-time nature of the PWA dynamics, we check the constraints only at sampling times. Under the assumption of convex state constraints it is even possible to ensure the constraint satisfaction at inter-sampling times by tightening the constraints (or equivalently increasing the size of the reachable) sets by the maximum curvature the trajectories can have in this time interval. As this is beyond the focus of the paper, we refer the interested reader to [76] and focus on the PWA dynamics in this example.

### 7.1. Results

We run our algorithm on a computer equipped with an Intel Core i7 @3.1GHz processor and 16GB of RAM; the MILP for reference trajectory computation is solved via CPLEX, whereas the linear programming problems for the set-based controller computation run on CVX [77] with the solver SDPT3 [78]. We use the CORA toolbox for the zonotope computations [79]. The overall computations take 2.5 minutes.

The results are presented in Fig. 4, where we project the resulting sets on the  $(h_1, h_2)$  and  $(h_3, h_4)$  planes. Here, we show the reference trajectory in black, the initial set in green, the

reachable sets at different time steps in blue, and the state constraints at different points in time in gray. We see that the initial set is steered along the reference trajectory and always stays inside the state constraints. It can also be seen that both the reference trajectory and the sets stay away from the mode edges, if possible. In particular, this can be seen in Fig. 4 (left) where the reachable set at the second time step is entirely maintained below the mode boundary  $h_1 = 3$  (magenta), and is entirely steered above the boundary at the third time step to prevent splitting. This is not possible for  $k = 5$ , as can be seen in Fig. 4 (left) when  $h_2$  crosses the line  $h_2 = 3$ , so that a split occurs. However, the split set is rather small (shown in red). Due to the different dynamics and disturbances, the evolution of the split set over time is different from the evolution of the main set (red sets propagating); however, we are able to steer the split set inside the main reachable set in  $M_{split} = 7$  time steps, thus avoiding branching. As a result, we can compute the controller via one MILP for the reference trajectory, 20 linear programs for the main reachable set, and a single linear program to steer the split set back to the main reachable set.

In addition, we also present the results for 5,000 simulations of the controller with random initial states and random disturbances in Fig. 5. We see that all state trajectories are indeed inside the reachable sets and therefore also satisfy the state constraints. The same is true for the input trajectories, which all satisfy the input constraints despite disturbances.

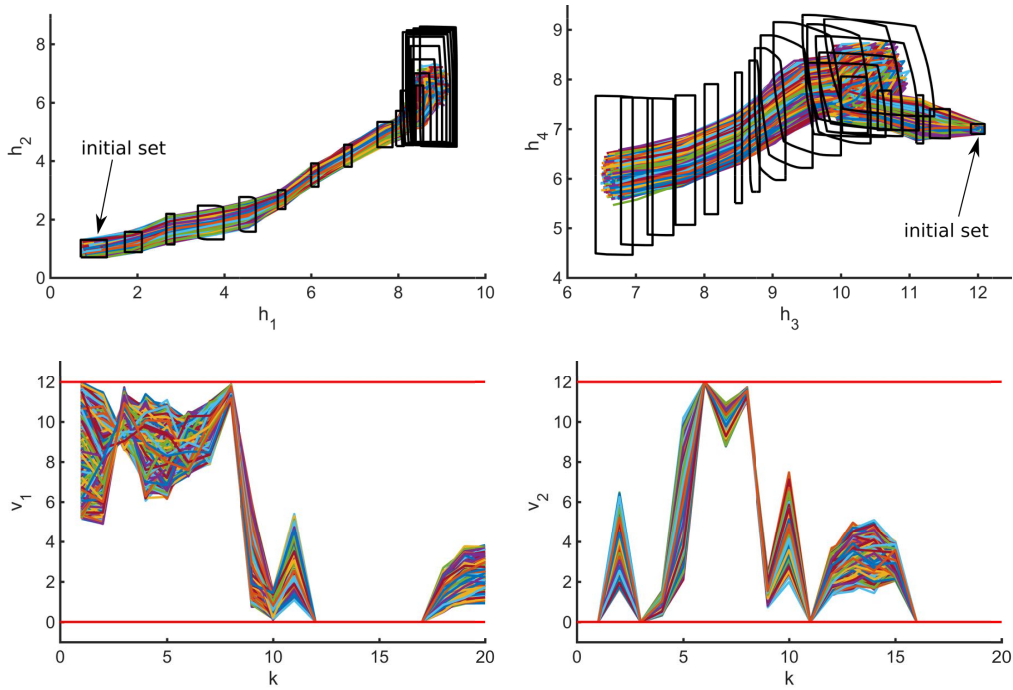


Figure 5: 5,000 simulation runs for the resulting controller with random initial states and random disturbances. Top: Resulting state trajectories and reachable sets from Fig. 4. Bottom: Corresponding input trajectories with inputs bounds plotted in red.

### 7.2. Comparison with Other Implementation

For comparison, we also solve the problem with a similar implementation, but without weighting the distance to the boundaries of modes or state constraints in the cost function when computing the reference trajectory. The resulting reference trajectory stays close to the lower boundaries of the time-varying state constraints, while just barely satisfying them. The set-based controller tries to stay close to this trajectory, while still satisfying the state constraints. This works for a few steps; however, after that, the MPC optimization problem becomes infeasible, and it is impossible to find a set-based controller which steers the whole reachable set inside the state constraints of the next time step because the lower bound of the state constraints increases faster than the controller can steer all states from the current reachable set.

We also consider another variant of our algorithm, where we weight the distances to the state constraints boundaries for the reference trajectory and the set-based controller, but we do not weight the distances  $d(k)$  to the mode boundaries, i.e., we set  $\gamma_d = 0$  in (21). In this case, we obtain a feasible solution for the main reachable set; however, we have to split two times. Moreover, it is not possible to steer any of these split sets into the main reachable set sequence after  $M_{split} = 7$  time steps. This therefore leads to two additional MILP and 27 additional linear programs and more than twice the computation time. Even if we consider the distance  $d(k)$  to the mode boundaries during the computation of the reference trajectory, we end up with two splits, one of which cannot be steered back into the main sequence if we do not consider the distance to the mode boundaries during the computation of the set-based controller. While decreasing the computation time, our approach increases the costs for the size of the reachable set  $cost(\mathcal{Z}_x(k))$  by only 10%, compared to the approach where we ignore the costs for the distance to the mode boundaries. This shows how directly considering the distance to the boundaries of the mode sets and the state constraints is crucial for obtaining good controller results and how we are able to minimize the overall computational effort and time by carefully choosing the cost functions.

### 7.3. Comparison with Other Tools

For comparison, we try to solve the problem at hand with Pessoa [80], a tool which first builds a discrete abstraction of the state and input set and afterwards computes a discrete controller using this abstraction. The advantage of this correct-by-construction method is that one obtains a finite state machine and can use tools which are able to handle linear temporal logic specifications. The general bottleneck of these approaches is the requirement to compute an abstraction, which scales exponentially in the state and input dimensions. To reduce the size of the abstraction, we confine the considered state space to the set  $\mathcal{S} = [0, 10] \times [0, 10] \times [3, 15] \times [3, 11]$ . We discretize the states with a resolution of 0.5 and the inputs with a resolution of 3 and thereby obtain a discrete abstraction with 187,425 states and 25 inputs. For simplicity, we do not include any time-dependence in the problem, i.e., we do not consider the time-varying state constraints nor we specify at what time the final set has to be reached. We could include this time-dependency by adding the time as an additional state, which would increase the number of states and therefore the computation time by a factor of 20. Since Pessoa synthesizes controllers of the form “stay in set  $\mathcal{A}$ ” or “reach set  $\mathcal{A}$  while staying in set  $\mathcal{B}$ ”, we change the task to “reach set  $\mathcal{G}$  while staying in set  $\mathcal{S}$ ”. Even if we choose the goal set to be the whole final state constraint, i.e.,  $\mathcal{G} = [3, 10] \times [3, 10] \times [3, 10] \times [3, 10]$ , we are not able to find a controller which works for all states of the initial set, not even if we ignore disturbances. We have also tried it with a state and input resolution of 1 each, which also leads to the same result that no controller exists which satisfies the specifications for all initial states with the abstracted model. With a finer resolution

of the abstraction, e.g., a state resolution of 0.5 and an input resolution of 1, it might be possible to find a controller. However, with a resolution of 1 for states and inputs, for which we could not find a controller, the computation of the abstraction already takes around 8 hours. If we use an input resolution of 1, we need 2.6 times as many inputs per state compared to the input resolution of 3, which would lead to a computation time of over 54 hours. If we also include the time-dependencies, we would even have a computation time of 1.5 months. This shows that for this example our approach scales much better. Please note that if one obtains a controller with Pessoa, one can reuse it for different initial states, while we need to recompute everything for a new initial state.

We also compare our approach with the one proposed in [81]. There, the authors reformulate the robust control as a MILP problem that simultaneously takes into account all the possible (unknown) switching sequences that the system may follow while satisfying the specification. We tested the approach on our case study and were not able to find a solution after several hours of computation: this is not unexpected since the number of binary variables appearing in the MILP used in [81] scales exponentially with the number of time instants. The computational burden of a time horizon of 20 steps is therefore already too high. The approach proposed in this paper also resorts to MILP, but the number of binary variables required in our case only scales linearly with the time horizon – thus, our approach is able to cope with larger instances of the problem.

## 8. Conclusion

We consider the problem of driving a PWA system with additive disturbances to some target set in a given finite horizon, while satisfying state and actuation constraints. To this purpose, we develop a set-based control design approach for a hybrid setting. In our approach, we account for the possible increase of the (offline) computation effort of the state feedback controller by explicitly aiming to prevent the disturbance from activating too many modes. If we are able to avoid splitting, our synthesis method reduces to a single MILP for the reference trajectory computation and  $N$  LPs, where  $N$  is the horizon length. Otherwise a mode recovery procedure (LP problem) and, possibly, a branching of the reference trajectory (MILP on the residual time horizon) are activated.

Notably, most computations are performed offline. The resulting control law is very simple to store and apply online. This is a main advantage of our approach in comparison with alternative methods that either require the online solving of computationally intensive problems or have explicit forms that are complex to store and to apply.

## Acknowledgment

The authors gratefully acknowledge financial support from the European Commission project UnCoVerCPS under grant number 643921.

## References

- [1] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica* 35 (3) (1999) 407–427.
- [2] E. Asarin, T. Dang, A. Girard, Hybridization methods for the analysis of nonlinear systems, *Acta Informatica* 43 (7) (2007) 451–476.

- [3] T. Dang, O. Maler, R. Testylier, Accurate hybridization of nonlinear systems, in: Proc. Hybrid Systems: Computation and Control, ACM, 2010, pp. 11–20.
- [4] A. Girard, S. Martin, Synthesis for constrained nonlinear systems using hybridization and robust controllers on simplices, *IEEE Transactions on Automatic Control* 57 (4) (2012) 1046–1051.
- [5] S. Bak, S. Bogomolov, T. A. Henzinger, T. T. Johnson, P. Prakash, Scalable static hybridization methods for analysis of nonlinear systems, in: Proc. Hybrid Systems: Computation and Control, ACM, 2016, pp. 155–164.
- [6] T. Geyer, F. D. Torrisi, M. Morari, Optimal complexity reduction of polyhedral piecewise affine systems, *Automatica* 44 (7) (2008) 1728–1740.
- [7] A. V. Papadopoulos, M. Prandini, Model reduction of switched affine systems, *Automatica* 70 (2016) 57 – 65.
- [8] A. Bemporad, G. Ferrari-Trecate, M. Morari, Observability and Controllability of Piecewise Affine and Hybrid Systems, *IEEE Transactions on Automatic Control* 45 (10) (2000) 1864–1876.
- [9] G. Ferrari-Trecate, M. Muselli, D. Liberati, M. Morari, A clustering technique for the identification of piecewise affine systems, *Automatica* 39 (2) (2003) 205 – 217.
- [10] J. Roll, A. Bemporad, L. Ljung, Identification of piecewise affine systems via mixed-integer programming, *Automatica* 40 (1) (2004) 37 – 50.
- [11] A. Bemporad, A. Garulli, S. Paoletti, A. Vicino, A bounded-error approach to piecewise affine system identification, *IEEE Transactions on Automatic Control* 50 (10) (2005) 1567–1580.
- [12] M. Rubagotti, S. Trimboli, A. Bemporad, Stability and invariance analysis of uncertain discrete-time piecewise affine systems, *IEEE Transactions on Automatic Control* 58 (9) (2013) 2359–2365.
- [13] M. Gholami, V. Cocquempot, H. Schiøler, T. Bak, Active fault tolerant control of piecewise affine systems with reference tracking and input constraints, *International Journal of Adaptive Control and Signal Processing* 28 (11) (2014) 1240–1265.
- [14] J. Xu, L. Xie, Control and Estimation of Piecewise Affine Systems, Elsevier Science, 2014.
- [15] A. Bemporad, W. P. M. H. Heemels, B. D. Schutter, On hybrid systems and closed-loop MPC systems, *IEEE Transactions on Automatic Control* 47 (5) (2002) 863–869.
- [16] M. Lazar, W. P. M. H. Heemels, S. Weiland, A. Bemporad, Stabilizing model predictive control of hybrid systems, *IEEE Transactions on Automatic Control* 51 (11) (2006) 1813–1818.
- [17] J. Lunze, F. Lamnabhi-Lagarigue, Handbook of hybrid systems control: theory, tools, applications, Cambridge University Press, 2009.
- [18] E. Asarin, T. Dang, G. Frehse, A. Girard, C. Le Guernic, O. Maler, Recent progress in continuous and hybrid reachability analysis, in: Proc. of the IEEE Conference on Computer Aided Control Systems Design, 2006, pp. 1582–1587.
- [19] M. Althoff, S. Bak, D. Cattaruzza, X. Chen, G. Frehse, R. Ray, S. Schupp, ARCH-COMP17 category report: Continuous and hybrid systems with linear continuous dynamics, in: Proc. of the 4th International Workshop on Applied Verification of Continuous and Hybrid Systems, 2017, pp. 143–159.
- [20] A. Chutinan, B. H. Krogh, Computational techniques for hybrid system verification, *IEEE Transactions on Automatic Control* 48 (1) (2003) 64–75.
- [21] A. B. Kurzhanski, P. Varaiya, Ellipsoidal techniques for reachability analysis, in: Hybrid Systems: Computation and Control, LNCS 1790, Springer, 2000, pp. 202–214.
- [22] A. Girard, C. Le Guernic, Efficient reachability analysis for linear systems using support functions, in: Proc. of the 17th IFAC World Congress, 2008, pp. 8966–8971.
- [23] X. Chen, S. Sankaranarayanan, E. Abrahám, Taylor model flowpipe construction for non-linear hybrid systems, in: Proc. of the 33rd IEEE Real-Time Systems Symposium, 2012, pp. 183–192.
- [24] A. Girard, Reachability of uncertain linear systems using zonotopes, in: Hybrid Systems: Computation and Control, 2005, pp. 291–305.
- [25] M. Althoff, B. H. Krogh, Zonotope bundles for the efficient computation of reachable sets, in: 50th IEEE Conference on Decision and Control, 2011, pp. 6814–6821.
- [26] M. Althoff, Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets, in: Hybrid Systems: Computation and Control, 2013, pp. 173–182.
- [27] J. K. Scott, D. M. Raimondo, G. R. Marseglia, R. D. Braatz, Constrained zonotopes: A new tool for set-based estimation and fault detection, *Automatica* 69 (2016) 126–136.
- [28] M. Althoff, G. Frehse, Combining zonotopes and support functions for efficient reachability analysis of linear systems, in: Proc. of the 55th IEEE Conference on Decision and Control, 2016, pp. 7439–7446.
- [29] A. Chutinan, B. H. Krogh, Verification of Polyhedral-Invariant Hybrid Automata Using Polygonal Flow Pipe Approximations, Springer, 1999, pp. 76–90.
- [30] G. Frehse, R. Ray, Flowpipe-guard intersection for reachability computations with support functions, in: Analysis and Design of Hybrid Systems, 2012, pp. 94–101.
- [31] M. Althoff, C. L. Guernic, B. H. Krogh, Reachable set computation for uncertain time-varying linear systems, in: Hybrid Systems: Computation and Control, 2011, pp. 93–102.

- [32] M. Althoff, B. H. Krogh, Avoiding geometric intersection operations in reachability analysis of hybrid systems, in: *Hybrid Systems: Computation and Control*, 2012, pp. 45–54.
- [33] J. Till, S. Engell, S. Panek, O. Stursberg, Applied hybrid system optimization: An empirical investigation of complexity, *Control Engineering Practice* 12 (10) (2004) 1291–1303.
- [34] M. S. Branicky, V. S. Borkar, S. K. Mitter, A unified framework for hybrid control: model and optimal control theory, *IEEE Transactions on Automatic Control* 43 (1) (1998) 31–45.
- [35] M. S. Branicky, S. K. Mitter, Algorithms for optimal hybrid control, in: *Proc. of IEEE Conference on Decision and Control*, Vol. 3, 1995, pp. 2661–2666.
- [36] M. S. Shaikh, P. E. Caines, On the optimal control of hybrid systems: Optimization of trajectories, switching times, and location schedules, *Lecture Notes in Computer Science* 2623 (2003) 466–481.
- [37] M. S. Shaikh, P. E. Caines, On the hybrid optimal control problem: theory and algorithms, *IEEE Transactions on Automatic Control* 52 (9) (2007) 1587–1603.
- [38] A. Bemporad, F. Borrelli, M. Morari, Optimal controllers for hybrid systems: Stability and piecewise linear explicit form, in: *Proc. of the Conference on Decision and Control*, Vol. 2, IEEE, 2000, pp. 1810–1815.
- [39] A. Bemporad, F. D. Torrisi, M. Morari, Optimization-based verification and stability characterization of piecewise affine and hybrid systems, in: *Proc. of Hybrid Systems: Computation and Control*, Springer, 2000, pp. 45–58.
- [40] A. Bemporad, F. Borrelli, M. Morari, Piecewise linear optimal controllers for hybrid systems, in: *Proc. of the American Control Conference*, 2000, pp. 1190–1194.
- [41] F. Borrelli, M. Baotić, A. Bemporad, M. Morari, Dynamic programming for constrained optimal control of discrete-time linear hybrid systems, *Automatica* 41 (10) (2005) 1709–1721.
- [42] B. Passenberg, O. Stursberg, Graph search for optimizing the discrete location sequence in hybrid optimal control, *IFAC Proceedings Volumes* 42 (17) (2009) 304–309.
- [43] S. Raković, P. Grieder, M. Kvasnica, D. Mayne, M. Morari, Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances, in: *Proc. of the Conference on Decision and Control*, Vol. 2, IEEE, 2004, pp. 1418–1423.
- [44] M. P. Silva, L. Pina, A. Bemporad, M. A. Botto, J. S. da Costa, Robust optimal control of linear hybrid systems: an MLD approach, in: *Proc. 6th Portuguese Conference on Automatic Control*, 2004, pp. 208–213.
- [45] M. P. Silva, A. Bemporad, M. A. Botto, J. S. da Costa, Optimal control of uncertain piecewise affine/mixed logical dynamical systems, in: *Proc. of the European Control Conference*, IEEE, 2003, pp. 1573–1578.
- [46] J. Lygeros, C. Tomlin, S. Sastry, Controllers for reachability specifications for hybrid systems, *Automatica* 35 (3) (1999) 349–370.
- [47] S. Di Cairano, W. M. H. Heemels, M. Lazar, A. Bemporad, Stabilizing dynamic controllers for hybrid systems: a hybrid control Lyapunov function approach, *IEEE Transactions on Automatic Control* 59 (10) (2014) 2629–2643.
- [48] L. C. G. J. M. Habets, P. J. Collins, J. H. van Schuppen, Reachability and control synthesis for piecewise-affine hybrid systems on simplices, *IEEE Transactions on Automatic Control* 51 (6) (2006) 938–948.
- [49] L. C. G. J. M. Habets, P. J. Collins, J. H. van Schuppen, Control to facet by piecewise-affine output feedback, *IEEE Transactions on Automatic Control* 57 (11) (2012) 2831–2843.
- [50] M. Zamani, G. Pola, M. Mazo Jr., P. Tabuada, Symbolic models for nonlinear control systems without stability assumptions, *IEEE Transactions on Automatic Control* 57 (7) (2012) 1804–1809.
- [51] A. Girard, S. Martin, Motion planning for nonlinear systems using hybridizations and robust controllers on simplices, in: *Proc. of the 47th Conference on Decision and Control*, 2008, pp. 239–244.
- [52] A. Girard, Controller synthesis for safety and reachability via approximate bisimulation, *Automatica* 48 (5) (2012) 947–953.
- [53] Y. Li, J. Liu, N. Ozay, Computing finite abstractions with robustness margins via local reachable set over-approximation, *IFAC-PapersOnLine* 48 (27) (2015) 1–6.
- [54] J. Liu, N. Ozay, U. Topcu, R. M. Murray, Synthesis of reactive switching protocols from temporal logic specifications, *IEEE Transactions on Automatic Control* 58 (7) (2013) 1771–1785.
- [55] N. Ozay, J. Liu, P. Prabhakar, R. M. Murray, Computing augmented finite transition systems to synthesize switching protocols for polynomial switched systems, in: *Proc. of the American Control Conference*, 2013, pp. 6237–6244.
- [56] A. Bemporad, M. Morari, Robust model predictive control: A survey, in: *Robustness in identification and control*, Springer, 1999, pp. 207–226.
- [57] D. Q. Mayne, S. Raković, Model predictive control of constrained piecewise affine discrete-time systems, *International Journal of Robust and Nonlinear Control* 13 (3–4) (2003) 261–279.
- [58] A. Alessio, A. Bemporad, *A Survey on Explicit Model Predictive Control*, Springer, 2009, pp. 345–369.
- [59] E. Camacho, D. Ramirez, D. Limon, D. M. de la Peña, T. Alamo, Model predictive control techniques for hybrid systems, *Annual Reviews in Control* 34 (1) (2010) 21 – 31.
- [60] M. Mukai, T. Azuma, M. Fujita, Robust receding horizon control for piecewise linear systems based on constrained positively invariant, in: *Proc. of the American Control Conference*, Vol. 3, 2002, pp. 2348–2353.
- [61] E. C. Kerrigan, D. Q. Mayne, Optimal control of constrained, piecewise affine systems with bounded disturbances,

- in: Proc. of the 41st IEEE Conference on Decision and Control, 2002, pp. 1552–1557.
- [62] I. Necoara, B. De Schutter, T. J. van den Boom, J. Hellendoorn, Model predictive control for perturbed continuous piecewise affine systems with bounded disturbances, in: Proc. of the 43rd IEEE Conference on Decision and Control, 2004, pp. 1848–1853.
- [63] Y. Zou, S. Li, Robust constrained model predictive control of piecewise linear systems with bounded additive disturbances, in: Proc. of the IEEE Conference on Decision and Control, 2009, pp. 6339–6344.
- [64] D. Mayne, M. Seron, S. Raković, Robust model predictive control of constrained linear systems with bounded disturbances, *Automatica* 41 (2) (2005) 219 – 224.
- [65] K. Hariprasad, S. Bhartiya, A computationally efficient robust tube based MPC for linear switched systems, *Nonlinear Analysis: Hybrid Systems* 19 (2016) 60 – 76.
- [66] M. S. Ghasemi, A. A. Afzalian, Robust tube-based MPC of constrained piecewise affine systems with bounded additive disturbances, *Nonlinear Analysis: Hybrid Systems* 26 (2017) 86 – 100.
- [67] B. Schürmann, M. Althoff, Guaranteeing constraints of disturbed nonlinear systems using set-based optimal control in generator space, in: Proc. of the 20th IFAC World Congress, 2017, pp. 12020–12027.
- [68] G. M. Ziegler, Lectures on polytopes, Vol. 152, Springer Science & Business Media, 2012.
- [69] G. B. Dantzig, M. N. Thapa, *Linear Programming 1: Introduction*, Springer, 2006.
- [70] M. Grant, S. Boyd, Y. Ye, Disciplined convex programming, in: *Global Optimization: from Theory to Implementation*, Springer, 2006, pp. 155–210.
- [71] A.-K. Kopetzki, B. Schürmann, M. Althoff, Methods for order reduction of zonotopes, in: Proc. of the 56th IEEE Conference on Decision and Control, 2017, pp. 5626–5633.
- [72] P. M. Vaidya, An algorithm for linear programming which requires  $O(((m+n)n^2 + (m+n)^{1.5}n)L)$  arithmetic operations, in: Proc. of the Annual Symposium on Theory of Computing, 1987, pp. 29–38.
- [73] B. Schürmann, A. El-Guindy, M. Althoff, Closed-form expressions of convex combinations, in: American Control Conference, 2016, pp. 2795–2801.
- [74] P. J. Goulart, E. C. Kerrigan, J. M. Maciejowski, Optimization over state feedback policies for robust control with constraints, *Automatica* 42 (2006) 523–533.
- [75] K. H. Johansson, The quadruple-tank process: A multivariable laboratory process with an adjustable zero, *Control Systems Technology, IEEE Transactions on* 8 (3) (2000) 456–465.
- [76] M. Althoff, Reachability analysis and its application to the safety assessment of autonomous cars, Ph.D. thesis, Technische Universität München (2010).
- [77] M. Grant, S. Boyd, CVX: Matlab software for disciplined convex programming, version 2.1, <http://cvxr.com/cvx> (2014).
- [78] K. C. Toh, M. J. Todd, R. H. Tütüncü, SDPT3 – a Matlab software package for semidefinite programming, Version 1.3, *Optimization Methods and Software* 11 (1-4) (1999) 545–581.
- [79] M. Althoff, An introduction to CORA 2015, in: Proc. of the Workshop on Applied Verification for Continuous and Hybrid Systems, 2015, pp. 120–151.
- [80] M. Mazo, A. Davitian, P. Tabuada, Pessoa: A tool for embedded controller synthesis, in: *Computer Aided Verification*, Springer, 2010, pp. 566–569.
- [81] R. M. Vignali, D. Ioli, M. Prandini, A data-driven approach to stochastic constrained control of piecewise affine systems, in: Proc. of the American Control Conference, 2018, pp. 1424–1429.