Robust bounded feasibility verification of piecewise affine systems via reachability computations

Riccardo Desimini and Maria Prandini *

* *Politecnico di Milano, Piazza Leonardo Da Vinci 32, 20133 Milano, Italy (e-mail: {riccardo.desimini, maria.prandini}@polimi.it).*

Abstract: We address robust bounded feasibility verification for a discrete-time PieceWise Affine (PWA) system whose evolution can be influenced by some input. The aim is to determine an input sequence that makes the system satisfy a certain property within a finite time horizon, while maximizing the amount of perturbation that can be applied to the input without violating the given property. This is important to assess the robustness of the solution to numerical errors. We focus on the case of a property expressed in terms of the output of the system taking values in a certain spec set, and propose a verification method resting on reachability computations. The idea is to determine the set of states that the system can reach through all its possible evolutions, with the input taking values in its whole range, and check if the computed reach sets intersect with the one corresponding to the output spec set. If this is the case, then, an input sequence driving the output to the spec set exists and can be determined together with its robustness level by solving a linear optimization program.

1. INTRODUCTION

In this paper, verification of dynamical systems evolving under the effect of some input is addressed. We focus on the class of the discrete-time PieceWise Affine (PWA) systems, whose evolution is governed by a collection of affine dynamics that become activated according to the value assumed by state and input inside a polyhedral partition of the state cross input space.

PWA systems have been extensively studied by researchers because of their significant modelling capabilities, despite their simple mathematical description made of linear equalities and inequalities. In Heemels et al. (2001), PWA systems have been proven to be equivalent to a class of linear hybrid systems called Mixed Logical Dynamical (MLD), a modelling framework that allows to describe various classes of systems like, e.g., finite state machines interacting with dynamical systems and systems with mixed discrete/continuous inputs and states, Bemporad and Morari (1999a).

Among the broad variety of approaches available in the literature for hybrid systems verification, Schupp et al. (2015), a well-known methodology is the one that performs reachability analysis through state sets computation to check if a certain property related to the temporal evolution of a given system is satisfied, Asarin et al. (2006). The idea is to compute the sets of states that the system can reach during its evolution (reach sets) by propagating the set of initial states through the system dynamics under the influence of all the possible inputs affecting the system. The effectiveness of a set-based reachability approach in terms of both computability and conservativeness of the result depends on the adopted representation of the reach sets, jointly with the system dynamics through which reach sets evolve.

According to such set-based reachability paradigm, a verification algorithm for discrete-time MLD systems based on Linear and Mixed Integer Linear Programming (LP/MILP) has been proposed in Bemporad and Morari (1999b) to assess if the system executions can exit a provided safe region in a bounded time horizon, and also to estimate the maximal output range of the system in such an horizon. In the proposed method, sets are modelled through convex polyhedra and interval outer-approximations are introduced to simplify sets representations while they are propagated through the system dynamics.

Here, we address a verification problem for discrete-time PWA systems where the aim is to determine if a certain property (called specification) expressed in terms of reachability of a polyhedral region in the output space can be satisfied at the earliest time instant of a given finite time horizon.

Differently from Bemporad and Morari (1999b), we model the state sets through zonotopes, i.e., centrally symmetric convex polytopes. By using zonotopes, propagation of sets through an affine dynamics subject to an input varying inside a zonotope can be done efficiently, even for high dimensional systems, Girard (2005). Moreover, to perform sets outer-approximations, we adopt and extend some methods in the literature that provide tighter approximations than intervals. Inspired by Vignali et al. (2014), we also introduce a robustness measure, and look for the maximum amount of perturbation that can be applied to the input while still satisfying the specification when computing a solution to our verification problem. In this way, the numerical reliability of the solution is improved.

Reachability analysis of hybrid systems has been also addressed in Frehse et al. (2011), where a tool modelling

piecewise affine systems by means of hybrid automata is presented. Such a tool also combines polyhedra and support function representations of continuous sets to compute an over-approximation of the reachable states. However, differently from our setup, the tool makes use of continuous-time models.

The tool CORA, Althoff (2015), can be applied to discrete-time nonlinear dynamical systems verification but requires some regularity of the nonlinear dynamics (see Althoff et al. (2018)), whereas we consider PWA systems and do not require their dynamics to be continuous.

Basic notions and notations

Given two positive integers m and n, the symbol $\mathbb{R}^{m\times n}$ denotes the space of the $m\times n$ real matrices and $\mathbb{R}^m$ stands for $\mathbb{R}^{m\times 1}$. The symbols $O_{m,n}$ and I_m denote respectively the $m \times n$ matrix with all zero entries and the identity matrix of order m, O_m stands for $O_{m,m}$ and 0_m stands for $O_{m,1}$. A (convex) polyhedron $\mathcal{P} \subset \mathbb{R}^h$ is defined as the intersection of q half-spaces (H-representation, Ziegler (2012)), and can be expressed through $P_a \in \mathbb{R}^{q\times h}$ and $p_b \in \mathbb{R}^q$ as $\mathcal{P} = \{z \in \mathbb{R}^h | P_a z \leq p_b\}$ or $\mathcal{P} = (P_a, p_b)$ for ease of notation. A polytope is a (convex) bounded polyhedron. Zonotopes are centrally symmetric convex polytopes. More precisely, a convex polytope in $\mathbb{R}^h$ is a zonotope if it can be written as $\mathcal{Z} = \{z \in \mathbb{R}^h | z = c + \sum_{i=1}^r \beta_i g_i, \beta_i \in [-1, 1]\}$, where $c \in \mathbb{R}^h$ is the center and $g_i \in \mathbb{R}^h, i \in \{1, \dots, r\}$, are the generators. We shall then use $\langle c, G\rangle$ as a more concise notation of $\mathcal{Z}$, where $G \in \mathbb{R}^{h\times r}$ is the generator matrix, which contains the generators as its columns. The ratio r/h is the order of the zonotope. Intervals in $\mathbb{R}^h$ are zonotopes. The propagation of a zonotopic set through an affine dynamics with input taking values in a zonotope leads to a zonotope (closure property of zonotopes with respect to affine transformations and Minkowski sum).

2. PROBLEM FORMULATION

Consider a PieceWise Affine (PWA) system described by

$$\begin{aligned} x_{k+1} &= A_i x_k + B_i u_k + e_i \quad \text{if } (x_k, u_k) \in \mathcal{M}_i \\ y_k &= C_i x_k + D_i u_k + f_i \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input, $y \in \mathbb{R}^p$ is the output, and $\mathcal{M}_i$ is the i-th mode of the system, $i = 1, \dots, s$. The mode collection $\{\mathcal{M}_i\}_{i=1}^s$ forms a polyhedral partition of the state cross input space, and matrices and vectors $A_i, B_i, e_i, C_i, D_i, f_i$ have appropriate dimensions. System (1) is initialized at $x_0 = \hat{x}_0$, and the input is constrained to the interval $\mathcal{U} = [\underline{u}, \overline{u}] \subset \mathbb{R}^m$. Given a positive integer k, the ordered collection $(i_0, \dots, i_k)$ of the active mode indices up to time k is the switching sequence of system (1) in the horizon $[0, k]$.

The goal is to determine if there exists a time instant $\bar{k} \in [0, N]$ and an input sequence $u_k^* \in \mathcal{U}$, $k = 0, \dots, \bar{k}$, such that the output satisfies $y_{\bar{k}} \in Y_{sp}$, where $Y_{sp} \subset \mathbb{R}^p$ is a polyhedral set. Among all possible time instants $\bar{k}$, we look for the minimum one. Given $\bar{k}$, we look for a sequence $\{u_k^*\}_{k=0}^{\bar{k}}$ with the maximum robustness level, where robustness is evaluated in terms of amount of perturbation that can be given to each input component u_{ki}^* $(i = 1, \dots, m)$ along the time horizon $[0, \bar{k}]$ while still satisfying the specification at time $\bar{k}$, i.e., $y_{\bar{k}} \in Y_{sp}$.

3. PROPOSED METHOD

In this section, we describe a method to solve the robust bounded feasibility problem for PWA systems that is based on reach sets computations. Consistently with the mode definition in (1), reach sets are computed in the state cross input space. It is then convenient to introduce vector $z = [x^T\ u^T]^T$ and rewrite the system equations (1) as:

$$\begin{aligned} z_{k+1} &= \tilde{A}_i z_k + \tilde{B}_i v_k + \tilde{e}_i \qquad \text{if } z_k \in \mathcal{M}_i \\ y_k &= \tilde{C}_i z_k + \tilde{f}_i \end{aligned} \tag{2}$$

where

$$\tilde{A}_i = \begin{bmatrix} A_i & B_i \\ O_{m,n} & 0_m \end{bmatrix} \quad \tilde{B}_i = \begin{bmatrix} O_{n,m} \\ I_m \end{bmatrix} \quad \tilde{e}_i = \begin{bmatrix} e_i \\ 0_m \end{bmatrix}$$
$$\tilde{C}_i = [C_i\ D_i] \quad \tilde{f}_i = f_i$$

for $i = 1, \dots, s$, and $v \in \mathcal{U}$ is a fictitious input vector used to assign to the u component of z all values in $\mathcal{U}$. [1]

The proposed approach rests on the computation of the set of all state-input values that can be reached starting from $\mathcal{R}_0 = \{\hat{x}_0\} \times \mathcal{U}$ at time $k = 0$ by applying all admissible input values. As soon as a time t is found where the state-input reach set $\mathcal{R}_t$ maps into a set of output values that intersects the spec set Y_{sp}, then, the bounded feasibility problem has a solution with $\bar{k} = t$, and the input sequence $(u_0^*, \dots, u_{\bar{k}}^*)$ with maximal robustness level can be computed by solving a linear optimization program.

The steps involved in the proposed method are sketched in the following algorithm:

Algorithm 1 Reachability-based verification algorithm

Require: $\hat{x}_0$, $\mathcal{U}$, PWA dynamics in (2), Y_{sp}
  for $k = 0, \dots, N$ **do**
    Compute $\mathcal{R}_k = \{R_{ki}\}_{i=1}^{n_k}$ (Section 3.1)
    for $i = 1, \dots, n_k$ **do**
      Test if $(\tilde{C}_{j_i} R_{ki} \oplus \tilde{f}_{j_i}) \cap Y_{sp} = \emptyset$ (Section 3.2)
      if $(\tilde{C}_{j_i} R_{ki} \oplus \tilde{f}_{j_i}) \cap Y_{sp} \neq \emptyset$ **then**
        Compute $(u_0^*, \dots, u_k^*)$ with maximal robustness level (Section 3.3)
        return $(u_0^*, \dots, u_k^*)$
      end if
    end for
  end for

3.1 Reach sets computation through zonotopes

We start observing that the input constraint set $\mathcal{U}$ is a zonotope with center $c_u = 0.5(\overline{u}+\underline{u})$ and generator matrix $G_u = 0.5\,\mathrm{diag}(\overline{u} - \underline{u})$.

If a reach set is a zonotope and it is contained in a mode, its propagation through the system dynamics is easy. More precisely, let $\mathcal{Z}_k = \langle c_{z,k}, G_{z,k}\rangle$ be a zonotopic reach set computed at time k. If $\mathcal{Z}_k$ is contained within mode $\mathcal{M}_i$, then, an arbitrary state $z_k = c_{z,k} + G_{z,k}\alpha_{z,k}$ in $\mathcal{Z}_k$ subject to input $v_k = c_u + G_u\alpha_{v,k}$ in $\mathcal{U}$ maps into

$$\begin{aligned} z_{k+1} &= \tilde{A}_i c_{z,k} + \tilde{A}_i G_{z,k}\alpha_{z,k} + \tilde{B}_i c_u + \tilde{B}_i G_u \alpha_{v,k} + \tilde{e}_i = \\ &= c_{z,k+1} + G_{z,k+1}\alpha_{z,k+1}, \end{aligned}$$

[1] Note that reformulation (2) is not needed if system (1) has modes described in the state-space only. In such a case, $z = x$ and $v = u$.

where we set

$$\begin{aligned} c_{z,k+1} &= \tilde{A}_i c_{z,k} + \tilde{B}_i c_u + \tilde{e}_i \\ G_{z,k+1} &= \begin{bmatrix} \tilde{A}_i G_{z,k} & \tilde{B}_i G_u \end{bmatrix} \\ \alpha_{z,k+1} &= \begin{bmatrix} \alpha_{z,k}^T & \alpha_{v,k}^T \end{bmatrix}^T \quad \|\alpha_{z,k+1}\|_\infty \leq 1. \end{aligned} \tag{3}$$

This shows that the reach set $\mathcal{Z}_{k+1}$ originated from $\mathcal{Z}_k$ when all possible values for v_k in $\mathcal{U}$ are applied to (2) is given by the zonotope $\mathcal{Z}_{k+1} = \langle c_{z,k+1}, G_{z,k+1} \rangle$. If each of the reach sets $\mathcal{Z}_1, \mathcal{Z}_2, \ldots, \mathcal{Z}_N$ is contained within one single mode, then, they are all zonotopes. Their centers and generators can be recursively computed by the equations in (3) initialized with

$$c_0 = \begin{bmatrix} \hat{x}_0 \\ c_u \end{bmatrix} \quad G_0 = \begin{bmatrix} O_{n,j_u} \\ G_u \end{bmatrix},$$

where j_u denotes the number of columns of G_u. Note that $\tilde{A}_i G_0 = O_{n+m,j_u}$, which corresponds to j_u degenerate generators that are identically zero and should then be removed from G_1. This is due to the fact that the initial state x_0 is not affected by uncertainty and the zonotope to which the input belongs is fixed.

If the reach set at time k splits over different modes, then, each part is a polytope (not necessarily a zonotope) and evolves according to a different affine dynamics. Different branches along which the reach sets are propagated are then generated. To ease the reach set propagation along each branch, it is convenient to adopt a zonotopic outer-approximation of each polytopic set P originated from the splitting. P is approximated by a tightly enclosing zonotope $Z_P = \langle c_{Z_P}, G_{Z_P} \rangle$ with an invertible generator matrix. More precisely, the center and the generator matrix are computed as follows:

$$c_{Z_P} = \frac{R}{2}\left(w_j^{\max} + w_j^{\min}\right), \quad G_{Z_P} = \frac{R}{2}\operatorname{diag}\left(w_j^{\max} - w_j^{\min}\right),$$

with $w_j^{\max} = \max_j w_j$ and $w_j^{\min} = \min_j w_j$, where w_j is the j-th vertex of the polytope obtained by applying the coordinate transformation matrix R^{-1} to P, with R representing a suitably chosen (invertible) matrix. R needs to be best chosen to get a tight over-approximation. To this purpose, one can adopt two methods:

Principal Component Analysis (PCA): the set of the vertices of P is interpreted as a set of data and R provides the transformation to a new orthogonal coordinate system where the greatest variance of the data is along the first axis, the second greatest variance of the data is along the second axis and so on and so forth. The method in Althoff (2010) can be adopted.

Maximum volume inner ellipsoid: the largest ellipsoidal inner approximation of P is computed and its axes are taken as directions of the new coordinate system. The algorithm presented in Zhang and Gao (2003) can be adopted to determine the ellipsoidal inner approximation of a full-dimensional polyhedron by solving a convex optimization program. We extended such an algorithm so as to deal with lower-dimensional polyhedra too.

Note that the outer approximation procedure can be also useful to maintain fixed the maximum number of generators of all zonotopes: whenever a zonotope has a larger number of generators, it is outer-approximated by a zonotope with the chosen maximum number of generators.

3.2 Testing the intersection with the spec set

At each time step k when propagating the reach set, we need to check if the specification is met for at least one of the currently computed sets, say R_{cur}, which can be either a polytope or a zonotope and is within a mode, say $\mathcal{M}_{cur}$. To this purpose, we first need to specify the set Z_{sp}^{cur} of states z that are within mode $\mathcal{M}_{cur}$ and satisfy the spec. A H-representation $(Z_{sp,a}, z_{sp,b})$ of Z_{sp}^{cur} is readily obtained through its definition:

$$Z_{sp,a} z = \begin{pmatrix} M_{cur,a} \\ Y_{sp,a}\tilde{C}_{cur} \end{pmatrix} z \leq \begin{pmatrix} m_{cur,b} \\ y_{sp,b} - Y_{sp,a}\tilde{f}_{cur} \end{pmatrix} = z_{sp,b}$$

where $(M_{cur,a}, m_{cur,b})$ and $(Y_{sp,a}, y_{sp,b})$ are respectively the H-representation of $\mathcal{M}_{cur}$ and Y_{sp}, and we use the output transformation in (2) when mode $\mathcal{M}_{cur}$ is active. Depending on the fact that R_{cur} is a zonotope, i.e., $R_{cur} = \langle c_{cur}, G_{cur} \rangle$, or a polytope, i.e., $R_{cur} = (H_{cur,a}, h_{cur,b})$, one of the following two (linear) feasibility tests can be used to check if it intersects Z_{sp}^{cur}:

find α	find z
subject to:	subject to:
$\|\alpha\|_\infty \leq 1$	$H_{cur,a} z \leq h_{cur,b}$
$Z_{sp,a} G_{cur} \alpha \leq z_{sp,b} - Z_{sp,a} c_{cur}$	$Z_{sp,a} z \leq z_{sp,b}$

Note that to solve such problems no explicit representation of $R_{cur} \cap Z_{sp}^{cur}$ is required.

3.3 Optimal input sequences computation

Suppose that $R_{cur} \cap Z_{sp}^{cur} \neq \emptyset$ at time step $\bar{k}$. To determine an input sequence that leads to the satisfaction of the specification with the maximum robustness level, we first recover the switching sequence originating the intersection $R_{cur} \cap Z_{sp}^{cur}$, say $(i_0, i_1, \ldots, i_{\bar{k}})$. According to such a sequence, system (1) reduces to the following constrained affine time varying system:

$$\begin{aligned} x_{k+1} &= A_{i_k} x_k + B_{i_k} u_k + e_{i_k} \\ y_k &= C_{i_k} x_k + D_{i_k} u_k + f_{i_k} \\ (x_k, u_k) &\in \mathcal{M}_{i_k},\ k = 0, 1, \ldots, \bar{k} \end{aligned} \tag{4}$$

Then, we can adopt the method in Vignali et al. (2014) for the class of linear time invariant systems by suitably adapting it to the time-varying affine dynamics in (4). The key idea is to parametrize each input component as follows:

$$u_{ki} = (1 - \beta_i)\hat{u}_{ki} + \beta_i \frac{\overline{u}_i + \underline{u}_i}{2} + \beta_i \frac{\overline{u}_i - \underline{u}_i}{2} w_{ki}, \tag{5}$$

where w_{ki} is a set-valued auxiliary signal taking values in $[-1, 1]$, whereas $\beta_i \in [0, 1]$ and $\hat{u}_{ki} \in [\underline{u}_i, \overline{u}_i]$ are optimization variables to be determined so as to maximize the size $\beta_i(\overline{u}_i - \underline{u}_i)$ of the interval

$$I_{ki} = [\hat{u}_{ki} + \beta_i(\underline{u}_i - \hat{u}_{ki}), \hat{u}_{ki} + \beta_i(\overline{u}_i - \hat{u}_{ki})] \subseteq [\underline{u}_i, \overline{u}_i].$$

Note that I_{ki} collapses to the singleton $\{\hat{u}_{ki}\}$ when $\beta_i = 0$, and coincides with the whole interval $[\underline{u}_i, \overline{u}_i]$ when $\beta_i = 1$.

We can replace the bilinear term $(1 - \beta_i)\hat{u}_{ki}$ in (5) with $u_{\beta ki}$ taking values in $[(1-\beta_i)\underline{u}_i, (1-\beta_i)\overline{u}_i]$, thus obtaining

$$u_{ki} = u_{\beta ki} + \frac{\underline{u}_i + \overline{u}_i}{2}\beta_i + \frac{\overline{u}_i - \underline{u}_i}{2}\beta_i w_{ki}. \tag{6}$$

Accordingly, the range I_{ki} of u_{ki} can be expressed as $I_{ki} = [u_{\beta ki} + \beta_i \underline{u}_i,\ u_{\beta ki} + \beta_i \overline{u}_i]$, and $\hat{u}_{ki}$ can be recovered through:

$$\hat{u}_{ki} = \begin{cases} \dfrac{u_{\beta ki}}{1 - \beta_i} & \text{if } \beta_i \in [0, 1) \\ 0.5(\underline{u}_i + \overline{u}_i) & \text{if } \beta_i = 1. \end{cases}$$

We are now in the position to formulate the constrained optimization problem to determine the input with the maximum range of variability such that the specification is satisfied with the PWA system evolving according to (4):

$$\max_{\beta_i\in[0,1],u_{\beta ki},i=1,\dots,m,k=0,\dots,\bar{k}} J(\beta) \quad (7)$$

subject to:

$$y_{\bar{k}} = C_{i_{\bar{k}}}x_{\bar{k}} + D_{i_{\bar{k}}}u_{\bar{k}} + f_{i_{\bar{k}}} \in Y_{sp}$$

$$x_{k+1} = A_{i_k}x_k + B_{i_k}\left[u_{k1}\ \dots\ u_{km}\right]^T + e_{i_k},\ x_0 = \hat{x}_0$$

$$u_{ki} = u_{\beta ki} + \frac{\underline{u}_i + \overline{u}_i}{2}\beta_i + \frac{\overline{u}_i - \underline{u}_i}{2}\beta_i w_{ki}$$

$$(1-\beta_i)\underline{u}_i \le u_{\beta ki} \le (1-\beta_i)\overline{u}_i$$

$$(x_k, u_k) \in \mathcal{M}_{i_k}$$

$$\forall w_{ki} \in [-1,1],\ i = 1,\dots,m,\ k = 0,\dots,\bar{k},$$

where the cost $J(\beta)$ can take one of the following forms: $J(\beta) = \min_{j=1,\dots,m}\beta_j$ or $J(\beta) = \sum_{j=1}^m \lambda_j\beta_j$ with $\lambda_j \in [0,1]$, $j = 1,\dots,m$, and $\sum_{j=1}^m \lambda_j = 1$. Irrespectively of the chosen form, problem (7) is a linear program.

If problem (7) is feasible, then, a solution to the robust bounded feasibility problem is found. If instead problem (7) is infeasible for all the sets R_{cur} computed at time $\bar{k}$ that intersect Z_{sp}^{cur}, then, one should move to time $\bar{k}+1$ and repeat similar computations. If the time step index reaches N and no solution is found, then, the robust bounded feasibility problem has no solution.

4. REDUCING THE COMPUTATIONAL EFFORT

The reach sets computation involves propagating a set through the PWA dynamics. Reach sets may intersect multiple modes, thus generating new branches in the reach set propagation associated with different switching sequences. Since the number of switching sequences grows exponentially with the time horizon length, the problem may become computationally challenging. In order to reduce the number of branches, we propose the following two methods:

Set-containment verification: if some newly computed set is contained within a previously computed reach set, then, its propagation is halted.

Maximal invariant set computation: whenever a new mode is visited when propagating the reach sets, we determine the maximal invariant set inside that mode, if there is any. If no state in the invariant set satisfies the specification, then, we halt the propagation of any reach set that is contained in such an invariant.

We now describe in more detail some computational aspects involved in the implementation of these methods.

4.1 Set-containment verification

Let R_{cur} be the currently computed set: such a set represents either a zonotope inside a single mode or a (polytopic) part of a zonotope that covers multiple modes. If no split occurred, then we must check inclusion between two zonotopes.

In order to check if a zonotope $R_{cur} = \langle c_{cur}, G_{cur}\rangle$ is contained inside a zonotope Z_{old}, a H-representation $(H_{old,a}, h_{old,b})$ of the container Z_{old} has to be computed. Such a conversion can be performed by means of the algorithm described in Althoff (2010). Then, the set-containment condition for R_{cur} is given by:

$$H_{old,a}c_{cur} + \max_{\alpha\in[-1,1]^r} H_{old,a}G_{cur}\alpha \le h_{old,b}$$

which is equivalent to

$$H_{old,a}c_{cur} + \|H_{old,a}G_{cur}\|_1 \le h_{old,b}$$

where the operators $\max(\cdot)$ and $\|\cdot\|_1$ are meant to be applied element-wise and row-wise respectively. If instead a split occurred, we have to check if a polytope $R_{cur} = (H_{cur,a}, h_{cur,b})$ is contained inside $Z_{old} = (H_{old,a}, h_{old,b})$ and the set-containment condition rewrites as $\max_{p\in R_{cur}} H_{old,a}p \le h_{old,b}$, where the maximum is precomputed by means of linear programs involving the rows of $H_{old,a}$ and matrices $H_{cur,a}$, $h_{cur,b}$.

The applicability of this procedure is limited by the complexity of the conversion of a G-representation to a H-representation, which is $O\left(n_g\binom{n_g}{d-1}\right)$, with n_g denoting the number of generators and d the space dimension, Althoff (2010).

4.2 Maximal invariant set computation

Once we fix a mode of the PWA system, say $\mathcal{M}$, the dynamics is affine and the maximal invariant set computation problem can be rephrased as follows.

Given the affine dynamical system

$$z_{k+1} = \tilde{A}z_k + \tilde{B}v_k + \tilde{e}, \quad (8)$$

determine (if it exists) the largest set $I \subseteq \mathcal{M}$ such that $\tilde{A}z + \tilde{B}v + \tilde{e} \in I$ for all $z \in I$ and for all $v \in \mathcal{U}$.

If system (8) has an equilibrium $\bar{z} \in \mathcal{M}$ associated to some constant input $\bar{v} \in \mathcal{U}$, the coordinate transformation $(\tilde{z}, \tilde{v}) = (z, v) - (\bar{z}, \bar{v})$ can be applied so as to reformulate (8) as follows:

$$\tilde{z}_{k+1} = \tilde{A}\tilde{z}_k + \tilde{B}\tilde{v}_k, \quad (9)$$

where $\tilde{v} \in \tilde{\mathcal{U}} = \{\tilde{u} \in \mathbb{R}^m : \tilde{u} = v - \bar{v},\ v \in \mathcal{U}\}$. System (8) is then reduced to a linear system where both the input set $\tilde{\mathcal{U}}$ and the set $\tilde{\mathcal{M}} = \{\tilde{z} \in \mathbb{R}^{n+m} : \tilde{z} = z - \bar{z},\ z \in \mathcal{M}\}$ contain the origin. In this context, an iterative algorithm has been proposed in Kolmanovsky and Gilbert (1998) that computes (if it exists) the maximal invariant set $\tilde{I}$ inside $\tilde{\mathcal{M}}$, from which the maximal invariant set inside $\mathcal{M}$ for system (8) can be recovered as $I = \{z \in \mathbb{R}^{n+m} : z = \tilde{z} + \bar{z},\ \tilde{z} \in \tilde{I}\}$.

To check if there exists any state in I that satisfies the specification, we need to introduce first the set $Z_{sp}^{\mathcal{M}}$ that contains any state z such that $\tilde{C}z + \tilde{f} \in Y_{sp}$, where $(\tilde{C}, \tilde{f})$ denotes the output dynamics associated with mode $\mathcal{M}$. Then, we have to check if the intersection $I \cap Z_{sp}^{\mathcal{M}}$ is empty and we can use the computational procedure described in Section 3.2 to this purpose.

5. A NUMERICAL EXAMPLE

In this section we apply the method in Section 3 to a numerical example of a bouncing ball. The ball is thrown with a force that acts only at the first time instant by providing some acceleration input. Then, it evolves of free

motion. We aim at verifying if the acceleration input at the first time instant can be chosen so that the ball reaches a given region by a prescribed minimum amount of bounces.

The model of the system is a discrete-time PWA obtained by sampling the continuous ball dynamics with period $T_s = 0.2\,\text{s}$. The state x and the input u are given by $x = [n_b\, p_h\, p_v\, v_h\, v_v\, \delta_u]^T \in \mathbb{R}^6$, $u = [a_h\, a_v]^T \in \mathcal{U} = [\underline{u}, \overline{u}]$, where $\underline{u} = [0\ \ -100]^T$, $\overline{u} = [50\ 0]^T$, n_b is the number of bounces, p_h and p_v are respectively the horizontal and vertical component of the ball position, v_h and v_v are respectively the horizontal and vertical velocity components, and δ_u is a scalar used to transmit the acceleration u to the ball at the initial time instant only.

The system dynamics is completely described by four different behaviours: forced motion (FoM), forced bounce (FoB), free motion (FrM) and free bounce (FrB), described by the following equations and activation conditions:

$$\text{FoM}: \begin{cases} n_{b,k+1} = n_{b,k} \\ p_{h,k+1} = p_{h,k} + T_s v_{h,k} \\ p_{v,k+1} = p_{v,k} + T_s v_{v,k} \\ v_{h,k+1} = v_{h,k} + T_s a_{h,k} \\ v_{v,k+1} = v_{v,k} + T_s (a_{v,k} - a_g) \\ \delta_{u,k+1} = 1 \end{cases} \text{if } \begin{cases} \delta_{u,k} < 0.5 \\ p_{v,k} + T_s v_{v,k} > 0 \end{cases}$$

$$\text{FoB}: \begin{cases} n_{b,k+1} = n_{b,k} + 1 \\ p_{h,k+1} = p_{h,k} + T_s v_{h,k} \\ p_{v,k+1} = 0 \\ v_{h,k+1} = v_{h,k} + T_s a_{h,k} \\ v_{v,k+1} = -\beta_r v_{v,k} \\ \delta_{u,k+1} = 1 \end{cases} \text{if } \begin{cases} \delta_{u,k} < 0.5 \\ p_{v,k} + T_s v_{v,k} \le 0 \end{cases}$$

FrM and FrB are activated if

$$\text{FrM}: \begin{cases} \delta_{u,k} \ge 0.5 \\ p_{v,k} + T_s v_{v,k} > 0 \end{cases} \qquad \text{FrB}: \begin{cases} \delta_{u,k} \ge 0.5 \\ p_{v,k} + T_s v_{v,k} \le 0 \end{cases}$$

and have the same equations as FoM and FoB respectively, but with the input set to zero ($a_{h,k} = a_{v,k} = 0$). The gravity acceleration a_g and the restitution coefficient of the ball β_r are set equal to $a_g = 9.8\,\frac{\text{m}}{\text{s}^2}$ and $\beta_r = 0.8$. FoM and FoB model respectively the ball motion and bounce when an external force is acting on it, while FrM and FrB model respectively the ball motion and bounce when the external force is not affecting the ball.

The system modes are defined in the state-space through the polyhedral partition associated to the following half-spaces: $HS_1 : -\delta_u \le -0.5$ and $HS_2 : p_v + T_s v_v \le 0$. To identify the i-th mode $\mathcal{M}_i$ of the system ($i = 1, \dots, 4$), we can use the following relation: $i = 2\delta_{i,1} + \delta_{i,2} + 1$, where $\delta_{i,j} = 1$ if mode $\mathcal{M}_i$ is contained in the half-space HS_j, and 0, otherwise. FoM is then associated to mode 1, FoB is associated to mode 2, FrM to mode 3 and FrB to mode 4. The above equations, conditions and half-spaces are then expressed in a compact way through the equation:

$$x_{k+1} = A_i x_k + B_i u_k + e_i \quad \text{if } (x_k, u_k) \in \mathcal{M}_i.$$

The system is initialized at $x_0 = \hat{x}_0 = [0\ 0\ 10\ 0\ 0\ 0]^T$, which activates mode $\mathcal{M}_1$ (FoM) at time $k = 0$. By introducing $y = [n_b^+\, p_h^+\, p_v^+]^T$ and the output transformation:

$$y_k = C_i x_k + D_i u_k + f_i \quad \text{if } (x_k, u_k) \in \mathcal{M}_i$$

where $C_i = M_y A_i$, $D_i = M_y B_i$, $f_i = M_y e_i$, $M_y = [I_3\ O_3]$, we can express the output specifications through the set

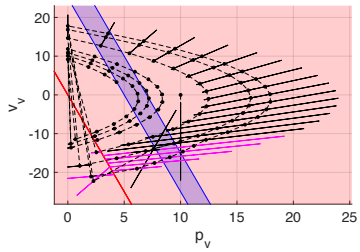

Fig. 1. Projections onto the p_v-v_v space of the computed sets in the horizon $[0, \bar{k}]$ associated to the switching sequence SS^*: in red the mode boundaries, in black the computed sets with their centers trajectory, in magenta the outer-approximations of the split ones and in blue the projection onto p_v-v_v of the set of states with output $y \in Y_{sp}$ for some $u \in \mathcal{U}$.

$$Y_{sp} = [\underline{n}_b, +\infty) \times [20, 22] \times [5, 7],$$

i.e., the ball is required to reach the square $[20, 22] \times [5, 7]$ after at least $\underline{n}_b$ bounces. The goal then is determining u at time $k = 0$ such that y enters Y_{sp} at the earliest time instant $\bar{k}$.

Consider $\underline{n}_b = 7$. In our implementation on a personal computer equipped with a dual-core 2.6 GHz Intel Core i5 processor and 8 GB of RAM, the value $\bar{k} = 103$ with an admissible switching sequence $SS^* = (i_0^*, \dots, i_{\bar{k}}^*)$ is found in about 2 minutes. Projections of the computed sets in $[0, \bar{k}]$ associated to SS^* are shown in Figure 1. Outer-approximations of split sets are computed using the PCA technique and linear programs are solved with CPLEX. After computing $\bar{k}$ and SS^*, problem (7) is solved by applying SS^* and minimizing $J(\beta) = \min_{j=1,\dots,m} \beta_j$, obtaining $\beta^* = [0.0097\ \ 0.0071]^T$ and $u_{\beta 0}^* = [4.8544\ \ -95.0682]^T$, which correspond to the maximal input range:

$$I_0^* = [4.8544, 5.3398] \times [-95.7797, -95.0682] \subseteq \mathcal{U}.$$

If we apply $u_0^* = [5.0971\ \ -95.4240]^T$, i.e., the center of I_0^*, the corresponding state trajectory in the horizon $[0, \bar{k}]$ is made of the centers of the (zonotopic) state sets obtained by applying all the inputs in I_0^*. Figure 2 represents the simulated values of p_h^+ and p_v^+ when u_0 is chosen both in I_0^* and $\mathcal{U} \setminus I_0^*$: as one can see, when $u_0 \in I_0^*$ the specifications at time $\bar{k}$ are always satisfied, while this is not necessarily true when $u_0 \in \mathcal{U} \setminus I_0^*$.

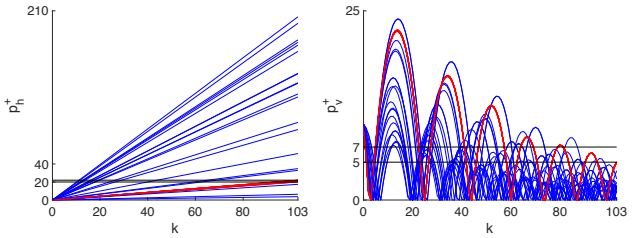

Fig. 2. Values of p_h^+ and p_v^+ for $k \in [0, \bar{k}]$ obtained by choosing $u_0 \in I_0^*$ (red) and $u_0 \in \mathcal{U} \setminus I_0^*$ (blue) with the boundaries of Y_{sp} (black).

An alternative approach to compute $\bar{k}$ and SS^* based on Mixed Integer Linear Programming (MILP) is now presented and then applied to our example. Consider the equivalent Mixed Logical Dynamical (MLD) reformulation of system (1) (see Bemporad and Morari (1999a) for the details):

$$
\begin{aligned}
x_{k+1} &= Ax_k + B_u u_k + B_\delta \delta_k + B_z z_k + f \\
y_k &= Cx_k + D_u u_k + D_\delta \delta_k + D_z z_k + g \\
E_x x_k &+ E_u u_k + E_\delta \delta_k + E_z z_k \leq e
\end{aligned} \tag{10}
$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$ and $y \in \mathbb{R}^p$ are respectively the state, input and output of system (1), while $z \in \mathbb{R}^q$ and $\delta \in \{0,1\}^d$ are vectors of auxiliary continuous and binary variables respectively. More precisely, δ_k is the binary conversion of the active mode index at time k. All the matrices and vectors in (10) are of suitable dimensions. To compute $\bar{k}$ and SS^* with reformulation (10), we proceed as follows. At step $k \in [0, N]$, a feasibility problem is solved where variables u, δ and z are chosen in $[0, k]$ so as to satisfy the constraints (10), $u \in \mathcal{U}$ and $y_k \in Y_{sp}$. Since (10) is a set of linear equalities and inequalities and Y_{sp} is polyhedral, the problem at hand is a MILP. If the MILP is feasible, then $\bar{k} = k$ and SS^* is readily obtained from the optimized binary vectors $\{\delta_j^*\}_{j=0}^k$, otherwise the same MILP is solved by choosing u, δ and z in the horizon $[0, k+1]$ and so on, until $k = N$. If for $k = N$ no feasible solution is found, then the bounded feasibility problem has no solution.

Table 1 shows the time required to compute SS^* for different values of $\underline{n}_b$ when using the set-based reachability computation method and the introduced MILP method, including also the case with $\bar{k}$ known. The MILP approach is affordable only when $\underline{n}_b$ (and thus $\bar{k}$) is small. Note that a single MILP needs to be solved when $\bar{k}$ is known. Yet, this is more costly than performing reachability computations for large values of $\bar{k}$. Mixed integer programs are solved with CPLEX, which by default employs parallel MIP optimization. A parallel implementation of reach set propagation along different branches should be investigated to speed up the set-based reachability computation method proposed in this paper.

Table 1. Time comparison between set-based reachability (SBR) and MILP approach.

	$\underline{n}_b = 2$	$\underline{n}_b = 3$	$\underline{n}_b = 5$	$\underline{n}_b = 7$
SBR	$\sim 6\,\mathrm{s}$	$\sim 15\,\mathrm{s}$	$\sim 53\,\mathrm{s}$	$\sim 82\,\mathrm{s}$
MILP ($\bar{k}$ unknown)	$\sim 1\,\mathrm{s}$	$\sim 6\,\mathrm{s}$	$\sim 165\,\mathrm{s}$	$\sim 5280\,\mathrm{s}$
MILP ($\bar{k}$ known)	$\sim 0.2\,\mathrm{s}$	$\sim 1\,\mathrm{s}$	$\sim 10\,\mathrm{s}$	$\sim 225\,\mathrm{s}$

Finally, we can assess the combinatorial complexity of the adopted methods by referring to the number of switching sequences for the set-based reachability method and the number of binary variables for the MILP method. The maximum number of switching sequences in $[0, \bar{k}]$ is given by $s^{\bar{k}} = 2^{d\bar{k}}$. In the MILP method, since $\bar{k}$ MILPs are solved where the j-th MILP has 2^{dj} binary variables, $j = 1, \ldots, \bar{k}$, we have $\frac{2^d(1-2^{d\bar{k}})}{1-2^d}$ binary variables in total. If $\bar{k}$ is known, the number of binary variables is $2^{d\bar{k}}$, which equals the number of switching sequences in the set-based reachability method. It then appears that by propagating the reach sets through the system dynamics, we are better exploiting the structure of the problem with respect to mathematical programming, where this is embedded within the constraints.

6. CONCLUSIONS

In this paper, robust bounded feasibility verification of PWA systems has been addressed through an approach based on reach sets computation. When the time horizon length is large, then, the proposed approach represents a valid alternative to other approaches for PWA systems based, e.g., on a MILP formulation.

REFERENCES

Althoff, M. (2010). *Reachability analysis and its application to the safety assessment of autonomous cars*. Ph.D. thesis, Technische Universität München, Germany.

Althoff, M. (2015). An introduction to CORA 2015. In *ARCH14-15. 1st and 2nd International Workshop on Applied Verification for Continuous and Hybrid Systems*, 120–151.

Althoff, M., Grebenyuk, D., and Kochdumper, N. (2018). Implementation of Taylor models in CORA 2018. In *ARCH18. 5th International Workshop on Applied Verification of Continuous and Hybrid Systems*, 145–173.

Asarin, E., Dang, T., Frehse, G., Girard, A., Le Guernic, C., and Maler, O. (2006). Recent progress in continuous and hybrid reachability analysis. In *2006 IEEE Conference on Computer Aided Control System Design*, 1582–1587.

Bemporad, A. and Morari, M. (1999a). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Bemporad, A. and Morari, M. (1999b). Verification of hybrid systems via mathematical programming. In *Hybrid Systems: Computation and Control*, 31–45.

Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., and Maler, O. (2011). SpaceEx: scalable verification of hybrid systems. In *Computer Aided Verification*, 379–395.

Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In M. Morari and L. Thiele (eds.), *Hybrid Systems: Computation and Control*, 291–305. Springer Berlin Heidelberg, Berlin, Heidelberg.

Heemels, W.P.M.H., De Schutter, B., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.

Kolmanovsky, I. and Gilbert, E.G. (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering*, 4(4), 317–367.

Schupp, S., Ábrahám, E., Chen, X., Ben Makhlouf, I., Frehse, G., Sankaranarayanan, S., and Kowalewski, S. (2015). Current challenges in the verification of hybrid systems. In *Cyber Physical Systems. Design, Modeling, and Evaluation*, 8–24.

Vignali, R., Deori, L., and Prandini, M. (2014). Control input design: detecting non influential inputs while satisfying a reachability specification. *IFAC Proceedings Volumes*, 47(3), 1416–1421.

Zhang, Y. and Gao, L. (2003). On numerical solution of the maximum volume ellipsoid problem. *SIAM Journal on Optimization*, 14(1), 53–76.

Ziegler, G.M. (2012). *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag New York, New York, NY, USA.