Proceeding of the 2004 American Control Conference
Boston, Massachusetts June 30 - July 2, 2004

Optimal Complexity Reduction of Piecewise Affine Models based on Hyperplane Arrangements

Tobias Geyer, Fabio D. Torrisi and Manfred Morari

Abstract— **This paper presents an algorithm that, given a piecewise affine (PWA) model, derives an *equivalent* PWA model that is *minimal* in the number of regions. The algorithm is based on the cells of the hyperplane arrangement that are already given when the PWA model is the result of the mode enumeration algorithm [9]. In particular, the algorithm executes a branch and bound search on the markings of the cells of the hyperplane arrangement assuring optimality. As we refrain from solving additional LPs, the algorithm is not only optimal but also computationally attractive. The applicability of the algorithm can be extended to derive minimal PWA representations of general PWA models by first computing the hyperplane arrangement. An example illustrates the algorithm and shows its computational effectiveness.**

I. Introduction

This paper focuses on the problem of finding a *minimal representation* of piecewise affine (PWA) models [12]. More specifically, for a given PWA model, we solve the problem of deriving a PWA model that is both equivalent to the former and minimal in the number of regions.

PWA models represent a universal modelling framework to describe hybrid systems. As shown in [11], they are equivalent under mild assumptions to various other hybrid systems frameworks. In general, hybrid systems feature thresholds or guardlines defined on states, inputs and internal variables. Associated logic signals are either true or false according to the fulfillment of these thresholds. Additionally, hybrid systems often encompass logic states that are part of a finite state machine or an automaton. A (feasible) combination of logic signals and logic states is usually called a *mode*. Assuming that the thresholds are linear, the set of states and inputs corresponding to the same mode forms a (convex) polyhedron whose facets are a subset of the thresholds. The collection of thresholds is a collection of hyperplanes, and if all hyperplanes solely depend on states and inputs, is equivalent to a hyperplane arrangement [9]. Computing all cells (or polyhedra) of the hyperplane arrangement is equivalent to enumerating the set of feasible modes. For each mode, a hybrid system features an associated dynamic. Here, we restrict ourselves to discrete-time and PWA dynamics. Furthermore, as the modes are defined such that the logic signals are constant for a given mode, these logic signals are included in the affine expression of the PWA dynamics by using binary instead of logic variables.

The authors are with the Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), CH– 8092 Zurich, Switzerland, geyer,torrisi,morari@control.ee.ethz.ch

Describing complex hybrid systems directly through PWA models is a tedious and non-trivial task in general. To facilitate the modelling, the HYbrid Systems DEscription Language (HYSDEL) [13] has been developed, that allows the designer to describe a hybrid system on a textual basis. A hybrid system described in HYSDEL can be regarded as a composition of discrete hybrid automata (DHA) [13] which are a mathematical abstraction of the features provided by other computation oriented and domain specific hybrid system frameworks.

In [9], we have formulated an algorithm that efficiently enumerates the feasible modes of a composition of DHAs. This allows the model designer not only to determine the real complexity of the compound model, but it also enables him to transform the compound model into an equivalent PWA model. Every mode corresponds to a polyhedron with an associated affine dynamic. Consequently, the PWA model is given by a set of polyhedra that form – if the PWA system is well-posed – a polyhedral partition with associated affine dynamics.

However, different modes often correspond to the same affine dynamic. If some or all of the associated polyhedra form a convex union, those polyhedra should be merged in order to reduce the complexity of the model. A model minimal in the number of polyhedra not only allows for a more compact representation, but the online computation time is also reduced significantly when the model is used in connection with Model Predictive Control (MPC). Similarly, when computing the explicit MPC feedback law [2], [4], it is of major importance to have a representation with as few regions as possible as the complexity of the algorithms depends exponentially on the number of polyhedra. The same holds for the number of resulting regions that represent the feedback law.

If the number of polyhedra with the same affine dynamic is large, the number of possible polyhedral combinations for merging explodes. As most of these unions are not convex or not even connected and thus cannot be merged, trying all combinations using standard techniques based on *linear programming* (LP) [3] is prohibitive. Furthermore, our objective here is not only to reduce the number of polyhedra but rather to find the minimal and thus optimal number of disjoint polyhedra. The mode enumeration algorithm [9] not only derives the polyhedral partition, but also the corresponding set of markings of the associated hyperplane arrangement. Using the markings enables us to determine *a priori* – i.e. without solving any LP – if a given combination

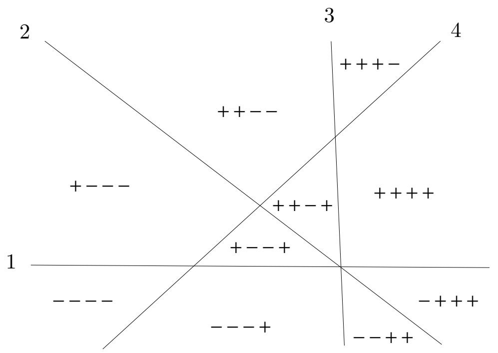


Fig. 1. Arrangement of four hyperplanes (lines) in $\mathcal{R} = \mathbb{R}^2$ with the markings $m \in M(\mathcal{R})$.

of polyhedra is convex. Exploiting this fact, we propose in this paper a branch and bound algorithm that yields the minimal number of polyhedra without solving LPs. Additional heuristics on the branching strategy allow us to reduce the computation time. Furthermore, the algorithm presented here can be extended to related cases lacking the hyperplane arrangement.

II. PRELIMINARIES

A. Cell Enumeration in Hyperplane Arrangement

Let $\mathcal{A}$ be a collection of n distinct hyperplanes $\{H_i\}_{i=1,\dots,n}$ in the d-dimensional Euclidian space $\mathbb{R}^d$, where each hyperplane is given by a linear equality $H_i = \{x \in \mathbb{R}^d \mid a_i x = b_i\}$. We say that the hyperplanes of $\mathcal{A}$ are in *general position*, if there exists no pair of parallel hyperplanes, and if any point of $\mathbb{R}^d$ belongs at most to d hyperplanes. Let SV : $\mathbb{R}^d \to \{-,+\}^n$ be the simplified sign vector defined as

$$\mathrm{SV}_i(x) = \begin{cases} - & \text{if } a_i x \le b_i, \\ + & \text{if } a_i x > b_i \end{cases} \quad \text{for } i \in \{1, 2, \dots, n\}\,. \quad (1)$$

Consider the set $\mathcal{P}_m = \{x \in \mathbb{R}^d \mid \mathrm{SV}(x) = m\}$ for a given sign vector m. This set is called a *cell* of the arrangement and is a polyhedron as it is defined by linear inequalities. We will refer to m as the *marking* of the polyhedron (or cell) $\mathcal{P}_m$ in the *hyperplane arrangement* $\mathcal{A}$ (see Fig. 1) [14]. Let $M(\mathcal{R})$ be the image of the function SV(x) for $x \in \mathcal{R} \subseteq \mathbb{R}^d$, namely the collection of all the possible markings of all the points in $\mathcal{R}$.

Let the '$*$' element extend the sign vector in the sense that it denotes the union of cells, where the associated hyperplane is not a facet of the associated polyhedron $\mathcal{P}_m$. As an example, consider in Fig. 1 the two polyhedra with the markings $m_1 = ----$ and $m_2 = +---$. Then, $m = *---$ is equivalent to $\{m_1, m_2\}$ and refers to $\mathcal{P}_{m_1} \cup \mathcal{P}_{m_2}$.

The cell enumeration problem in a hyperplane arrangement amounts to enumerate all the elements of the set $M(\mathcal{R})$. Let $\#M(\mathcal{R})$ be the number of cells identified by $M(\mathcal{R})$. Buck's formula [5] sets the upper bound $\#M \le \sum_{i=0}^{d} \binom{n}{i} = O(n^d)$ with the equality satisfied if the hyperplanes are in general position and $\mathcal{R} = \mathbb{R}^d$.

The cell enumeration problem admits an optimal solution with time and space complexity $O(n^d)$ [6]. An alternative approach based on reverse search was presented in [1], improved in [8] and implemented in [7], that runs in $O(n\ \mathrm{lp}(n,d)\ \#M)$ time and $O(n,d)$ space, where $\mathrm{lp}(n,d)$ denotes the complexity of solving a linear program (LP) with n constraints in d variables [8, Theorem 4.1]. Note that in many cases of interest, the hyperplanes are not in general position and $\#M$ can be considerably smaller than the theoretical upper bound.

From the definition of $\mathcal{P}_m$ and (1) follows directly that the collection of polyhedral sets $\{\mathcal{P}_m\}_{m \in M(\mathcal{R})}$ satisfies $\bigcup_{m \in M(\mathcal{R})} \mathcal{P}_m = \mathcal{R}$ and $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$, $\forall i, j \in M(\mathcal{R})$, $i \ne j$. A collection of polyhedral sets with these two properties is a *polyhedral partition* of the polyhedral set $\mathcal{R}$ [14].

B. Piecewise Affine Systems

Piecewise affine systems [11], [12] are defined by partitioning the state-input space into polyhedra and associating with each polyhedron an affine state-update and output function, i.e.

$$x(k+1) = A_{j(k)} x(k) + B_{j(k)} u(k) + f_{j(k)} \quad (2a)$$
$$y(k) = C_{j(k)} x(k) + D_{j(k)} u(k) + g_{j(k)} \quad (2b)$$
$$\text{with } j(k) \text{ such that } \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \in \mathcal{P}_{j(k)}, \quad (2c)$$

where $x(k) \in \mathcal{X}$, $u(k) \in \mathcal{U}$, $y(k) \in \mathcal{Y}$ denote at time k the real and binary states, inputs and outputs, respectively, the polyhedra $\mathcal{P}_{j(k)}$ define a set of polyhedra $\{\mathcal{P}_j\}_{j \in \mathcal{J}}$ on $\mathcal{X} \times \mathcal{U}$, and the real matrices $A_{j(k)}$, $B_{j(k)}$, $C_{j(k)}$, $D_{j(k)}$ and real vectors $f_{j(k)}$, $g_{j(k)}$ with $j(k) \in \mathcal{J}$, $\mathcal{J}$ finite, are constant and have suitable dimensions. In particular, $\mathcal{J}$ denotes the set of feasible modes. Throughout this paper, we assume that the PWA model is well-posed, i.e. the set of polyhedra $\{\mathcal{P}_j\}_{j \in \mathcal{J}}$ is a polyhedral partition of the state-input space $\mathcal{X} \times \mathcal{U}$.

C. Mode Enumeration Algorithm

Given a hybrid system modelled as a composition of DHAs in HYSDEL, the mode enumeration algorithm [9] derives the set of feasible modes $\mathcal{J}$, the set of polyhedra $\{\mathcal{P}_j\}_{j \in \mathcal{J}}$ defined on the state-input space $\mathcal{X} \times \mathcal{U}$ and the corresponding affine dynamics $\{\mathcal{S}_j\}_{j \in \mathcal{J}}$, where $\mathcal{S}_j = \{A_j, B_j, f_j, C_j, D_j, g_j\}$. Details about compositions of DHAs and the mode enumeration algorithm are not relevant here and can be found in [9]. However, most important is the fact, that the algorithm also yields the hyperplane arrangement $\mathcal{A}$ and the set of markings $M(\mathcal{R})$ which are both defined on $\mathcal{R}$, with $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{U}$. These markings will form the basis for the optimal complexity reduction in the next section. Next, we define global and local hyperplane arrangements.

If all the thresholds are defined only on states and inputs, the hyperplane arrangement and the markings are defined obviously on the state-input space, too, and consequently $\mathcal{R} = \mathcal{X} \times \mathcal{U}$. We refer to such a hyperplane arrangement as a *global* (or globally valid) hyperplane arrangement.

Furthermore, because of the equivalence of DHAs and PWA models [11], [13], the modes $j \in \mathcal{J}$ are formally equivalent to the markings $m \in M(\mathcal{R})$, and we may write $\{\mathcal{P}_j\}_{j\in\mathcal{J}} = \{\mathcal{P}_m\}_{m\in M(\mathcal{R})}$ and $\{\mathcal{S}_j\}_{j\in\mathcal{J}} = \{\mathcal{S}_m\}_{m\in M(\mathcal{R})}$.

If, however, some of the thresholds also depend on auxiliary variables that on their part depend on other thresholds, the mode enumeration algorithm yields a *collection* of hyperplane arrangements that are sequentially defined within each other. Hence, $\mathcal{R}$ is not the whole state-input space but rather a polytopic subset of it. We say that the hyperplane arrangement is *local* (or locally valid). This is generally the case for compositions of DHAs that sequentially depend on each other, where each DHA defines a hyperplane arrangement within a cell of the hyperplane arrangement of the preceding DHA. Consequently, the (global) modes of the PWA model do not correspond to the markings of the (local) hyperplane arrangements. For further details – particularly on compositions of DHAs containing loops – the reader is referred to [9] and references therein.

III. Optimal Merging

For a given PWA representation the aim of the optimal merging algorithm is to find an equivalent representation that is minimal in the number of polyhedra by merging as many polyhedra with the same affine dynamic as possible. For clarity of exposition, we associate with each affine dynamic a different color, and we collect the polyhedra with the same color. Then, for a given color, we want to solve the following problem.

Problem 1 (Optimal Merging Problem): Given an initial set of polyhedra $\{\mathcal{P}_i\}_{i=1,\dots,p}$ with the same color, the optimal merging problem amounts to derive a new set of polyhedra $\{\mathcal{Q}_i\}_{i=1,\dots,q}$ with the following properties: (i) the union of the new polyhedra is equal to the union of the original ones, i.e. $(\bigcup_{i=1}^{q}\mathcal{Q}_i) = (\bigcup_{i=1}^{p}\mathcal{P}_i)$, (ii) the new polyhedra are mutually disjoint, i.e. $\mathcal{Q}_i \neq \mathcal{Q}_j$ for all $i,j \in \{1,\dots,q\}$, $i \neq j$, (iii) the new polyhedra are formed as unions of the old ones, i.e. for each $\mathcal{Q}_j$, $j \in \{1,\dots,q\}$, there exists an index set $I \subseteq \{1,\dots,p\}$, such that $\mathcal{Q}_j = \bigcup_{i\in I}\mathcal{P}_i$, and (iv) q is minimal, i.e. there exists no set $\{\mathcal{Q}_i\}_{i=1,\dots,q}$ with a smaller number of polyhedra.

Hence the problem can be considered as an optimal set partitioning problem. This task is non-trivial, as the union of polyhedra with the same color is in general non-convex, and because we do not allow for overlapping polyhedra.

In this section, we assume that besides the PWA representation a corresponding *global* hyperplane arrangement $\mathcal{A}$ is available together with the markings $M(\mathcal{R})$. In the next section, we will relax this assumption and extend the merging algorithm to general PWA system.

A. Preliminaries

The proof of the following lemma follows directly from the definition of the markings.

Lemma 1 (Separating Hyperplane): Given the hyperplane arrangement $\{H_i\}_{i=1,\dots,n}$ consisting of n distinct hyperplanes, the set of markings $M(\mathcal{R})$, and the two polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$ with the corresponding markings $m_1, m_2 \in M(\mathcal{R})$ which differ in the j-th component, H_j is a separating hyperplane for $\mathcal{P}_1$ and $\mathcal{P}_2$ [14].

Definition 1 (Envelope, [3] p. 144): Given two polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$, the envelope $\text{env}(\mathcal{P}_1,\mathcal{P}_2)$ of the two polyhedra is defined as the intersection of halfspaces that contain both polyhedra, where the halfspaces are given by the facets of the two polyhedra.

Lemma 2 (Envelope): Given the hyperplane arrangement $\{H_i\}_{i=1,\dots,n}$ consisting of n distinct hyperplanes, the set of markings $M(\mathcal{R})$, and the two polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$ with the corresponding markings $m_1, m_2 \in M(\mathcal{R})$, where $m_1(i) = m_2(i)$ for $i \in I$ and $m_1(i) \neq m_2(i)$ for $i \in I'$ with $I' = \{1,\dots,n\}\setminus I$, we construct the marking m as follows. $m(i) = m_1(i)$ for $i \in I$ and $m(i) =$'$*$' for $i \in I'$. Then the envelope $\text{env}(\mathcal{P}_1,\mathcal{P}_2)$ of the two polyhedra is given by the marking m.

Proof: As all the facets of $\mathcal{P}_1$ and $\mathcal{P}_2$ are subsets of the hyperplanes in the arrangement, and as the hyperplanes with indices I' are separating hyperplanes for $\mathcal{P}_1$ and $\mathcal{P}_2$ according to Lemma 1, the proof follows from the definition of the envelope. ■

The proof can be easily generalized to envelopes of more than two polyhedra.

Theorem 1 (Convexity, [3] Theorem 3): Given the two polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$, their union $\mathcal{P}_1 \cup \mathcal{P}_2$ is convex if and only if $\mathcal{P}_1 \cup \mathcal{P}_2 = \text{env}(\mathcal{P}_1,\mathcal{P}_2)$.

The following lemma allows us to determine the convexity of two polyhedra by only evaluating their corresponding markings. This lemma constitutes the basis for the optimal merging algorithm.

Lemma 3 (Convexity): Given the collection of markings $M(\mathcal{R})$, the union of the two polyhedra $\mathcal{P}_1$ and $\mathcal{P}_2$ with the markings $m_1, m_2 \in M(\mathcal{R})$, $m_1 \neq m_2$, is convex, if and only if the markings differ in exactly one component.

Proof: As we have Theorem 1 at our disposal, we only need to prove, that $\mathcal{P}_1 \cup \mathcal{P}_2 = \text{env}(\mathcal{P}_1,\mathcal{P}_2)$ if and only if m_1 and m_2 differ in exactly one component. The "$\Leftarrow$" part follows directly from Lemma 2. The "$\Rightarrow$" part follows by contradiction. Recall, that $\mathcal{P}_1 \cup \mathcal{P}_2 \subseteq \text{env}(\mathcal{P}_1,\mathcal{P}_2)$, and assume that $\mathcal{P}_1 \cup \mathcal{P}_2 \neq \text{env}(\mathcal{P}_1,\mathcal{P}_2)$, i.e. there are points $x \in \text{env}(\mathcal{P}_1,\mathcal{P}_2) \setminus (\mathcal{P}_1 \cup \mathcal{P}_2)$. Then there exists at least one hyperplane that is separating x from $\mathcal{P}_1$ or x from $\mathcal{P}_2$ besides the one that is separating $\mathcal{P}_1$ from $\mathcal{P}_2$. Thus m_1 and m_2 differ in at least two components. ■

The concept of markings in a hyperplane arrangement allows us to evaluate the convexity of polyhedra by applying Lemma 3 to its associated set of markings. The algorithm refrains from solving LPs – in fact, it extracts the information from the markings that in turn summarize the result of the LPs solved to compute the cells of the hyperplane arrangement. Although we will use branch and bound techniques to assure optimality, the computation time to solve the optimal merging problem is rather small making the algorithm applicable to problems of meaningful size.

B. Precomputation

Definition 2 (Connectivity): Two polyhedra are called *neighboring* polyhedra if they share a common facet. A set of polyhedra $\{\mathcal{P}_i\}_{i\in I}$ is *connected* if for each $\mathcal{P}_i$, $i \in I$, there exists a $\mathcal{P}_j$, $i \neq j$, $j \in I$ such that $\mathcal{P}_i$ and $\mathcal{P}_j$ are neighboring polyhedra.

Obviously, a necessary condition for the convexity of a union of a set of polyhedra is that the set of polyhedra is connected. The connectivity can be easily determined using the markings. In order to reduce the computation time, we exploit this fact by further partitioning the set of polyhedra with the same color into connected subsets.

C. Branch and Bound Algorithm

Let the set M_w denote the markings of a connected subset with the same color. We refer to the corresponding polyhedra as *white* polyhedra. As the color of the remaining polyhedra is not relevant at this stage, we assume that the remaining markings $M_b' = M(\mathcal{R})\backslash M_w$ correspond to *black* polyhedra. The basic concept of the algorithm is to derive a minimal representation of the white polyhedra by dividing their envelope sequentially into polyhedra using the hyperplanes of the hyperplane arrangement.

Let the envelope of the white polyhedra with markings M_w be denoted by $\mathcal{P}_m$. It is given by the marking m, which is constructed as in Lemma 2. Slightly abusing the notation we will write $m = \text{env}(M_w)$. As all the white polyhedra are contained in their envelope, we can formulate an equivalent problem with reduced complexity that considers only the black polyhedra contained in this envelope, i.e. $M_b = \{m_b \in M_b' \mid \mathcal{P}_{m_b} \subseteq \mathcal{P}_m\}$, where $\mathcal{P}_{m_b}$ denotes the polyhedron with marking m_b.

Let I denote the index set of hyperplanes in $\mathcal{A}$ that are separating hyperplanes for polyhedra in the envelope $\mathcal{P}_m$. According to Lemma 1, I is simply the collection of indices i with $m(i) =$'$*$'. Then, we can choose any hyperplane H_i, $i \in I$, to divide $\mathcal{P}_m$ into two polyhedra. H_i also divides the sets of white and black markings respectively into two subsets. We denote the subset of M_w that holds those markings whose i-th element is a '$-$' with $M_w|_{m(i)=-}$, i.e. $M_w|_{m(i)=-} \triangleq \{m \in M_w \mid m(i) = \text{'}-\text{'}\}$. $M_w|_{m(i)=+}$ and the partition of M_b are defined accordingly. Clearly, the unions of each pair of subset equal the original sets M_w and M_b, respectively.

Next, the algorithm branches on the i-th hyperplane by calling itself twice – first with the arguments M_w and M_b restricted to possessing a '$-$' as i-th element, and then correspondingly with the arguments restricted to a '+'. Both function calls return sets of markings M_m corresponding to merged white polyhedra. This is repeated for all the remaining hyperplanes with indices $i \in I$.

A merging branch terminates if one of the following two cases occurs. First, if the set of markings corresponding to black polyhedra is empty, i.e. $M_b = \emptyset$. This implies, that at this point the envelope contains only white polyhedra. Hence, the envelope represents the union of the set of white polyhedra with markings in M_w, and it is convex by construction. We will refer to this convex set as a *merged white* polyhedron. Second, if the set of markings corresponding to white polyhedra is empty, i.e. $M_w = \emptyset$, as this implies, that no more white polyhedra are available for merging.

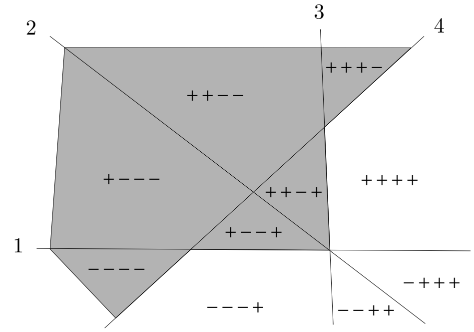


Fig. 2. Example for optimal merging with four hyperplanes in $\mathcal{R} = \mathbb{R}^2$ and the corresponding markings. The polyhedra corresponding to M_w are white and the polyhedra corresponding to M_b' are grey shaded, respectively.

The algorithm uses standard bound techniques to cut off suboptimal branches by using the two variables z and $\bar{z}$. z denotes the current number of merged white polyhedra and $\bar{z}$ is the local upper bound on z, where initially $z = 0$ and $\bar{z} = \infty$. Branching is only performed if $z < \bar{z}$, as branches with $z \geqslant \bar{z}$ are either equivalent to or worse than the current optimum.

The branch and bound algorithm for deriving the minimal number of merged polyhedra is summarized in the following. With $\#M$ we denote the cardinality of the set M.

Algorithm 1:

function M_m = Merge(M_w, M_b', z, $\bar{z}$)
$m = \text{env}(M_w)$
$M_b = \{m_b \in M_b' \mid \mathcal{P}_{m_b} \subseteq \mathcal{P}_m\}$
if $M_w = \emptyset$ **then** $M_m = \emptyset$
elseif $M_b = \emptyset$ **then** $M_m = m$
else
  $I = \{i \mid m(i) = \text{'}*\text{'}\}$
  $M_m = \emptyset$
  for $i \in I$
    if $z < \bar{z}$ **then**
      M_{m_1} = Merge ($M_w|_{m(i)=-}$, $M_b|_{m(i)=-}$, $z, \bar{z}$)
      M_{m_2} = Merge ($M_w|_{m(i)=+}$, $M_b|_{m(i)=+}$, $z + \#M_{m_1}, \bar{z}$)
      if $M_m = \emptyset$ **or** $\#M_{m_1} + \#M_{m_2} < \#M_m$ **then**
        $M_m = M_{m_1} \cup M_{m_2}$
        $\bar{z} = \min(\bar{z},\, z + \#M_m)$
return M_m

Example 1: As an example with four hyperplanes in a two-dimensional space consider Fig. 2. The envelope of the white polyhedra is given by the positive halfspace of H_4 and the marking $m = {*}{*}{*}{+}$. Thus, only the black polyhedra with markings $M_b = \{{+}{-}{-}{+}, {+}{+}{-}{+}\}$ are considered, and branching is only performed on the hyperplanes in $I = \{1, 2, 3\}$. Branching on H_1 leads in one step to the two merged (white) polyhedra with $M_m = \{{-}{*}{*}{+}, {+}{+}{+}{+}\}$. This is already the optimal solution. Nevertheless, the algorithm also branches on the two remaining hyperplanes in I and finds two additional solutions that are equivalent to the first one in terms of the number of polyhedra.

Lemma 4: Algorithm 1 solves the optimal merging problem stated in Problem 1.

Proof: The proof follows in a constructive way from the algorithm. More specifically, when branching on the i-th hyperplane H_i, the set of white markings is divided into the two sets $M_w|_{m(i)=-}$ and $M_w|_{m(i)=+}$ according to the two halfspaces defined by H_i. This operation assures that the merged polyhedra are mutually disjoint. In particular, as no white polyhedra are discarded during the operation and since $M_w = (M_w|_{m(i)=-}) \cup (M_w|_{m(i)=+})$, the union of the merged polyhedra equals the union of the white polyhedra. The minimality of the number of merged polyhedra is ensured by branching on all hyperplanes. ■

Remark 1: According to the property (iii) of Problem 1, the merged polyhedra are unions of the original white ones. This implies, that the facets of the merged polyhedra are a subset of the facets of the white polyhedra. Algorithm 1 derives a solution which is minimal in the number of *merged* polyhedra. However, by introducing additional facets, the number of polyhedra might be further reduced. Thus, in general, the merged polyhedra constitute only a suboptimal solution to the (more general) covering problem which is not addressed here. Nevertheless, even though such cases can be constructed, they are very rare and have not been encountered in applications so far.

D. Derivation of Global Hyperplane Arrangement

Algorithm 1 in the proposed form is only applicable to problems with a globally valid hyperplane arrangement. In this section, we propose two extensions that will allow us to employ the algorithm also for problems with local hyperplane arrangements, or even more general, for problems that altogether lack a hyperplane arrangement.

As mentioned before, local hyperplane arrangements are defined in a polyhedron $\mathcal{R}$ which is a subset of the state-input space, i.e. $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{U}$. For a given $\mathcal{R}$, the hyperplane arrangement is readily available together with the markings. Thus, optimal merging can be performed for each subset $\mathcal{R}$. The overall solution is then the union of the local solutions. Even though the results are locally optimal, the overall solution is in general *suboptimal*. As an example, consider two local hyperplane arrangements which each encompass one white polyhedron and a number of black polyhedra, and assume that the union of these two white polyhedra is convex. Using Algorithm 1 twice (for each local hyperplane arrangement) fails to merge the two white polyhedra, and is thus clearly suboptimal. Nevertheless, this approach is meaningful if we are interested only in reducing the number of polyhedra but not necessarily in finding the minimal number, and have limited time and computational power at our disposal.

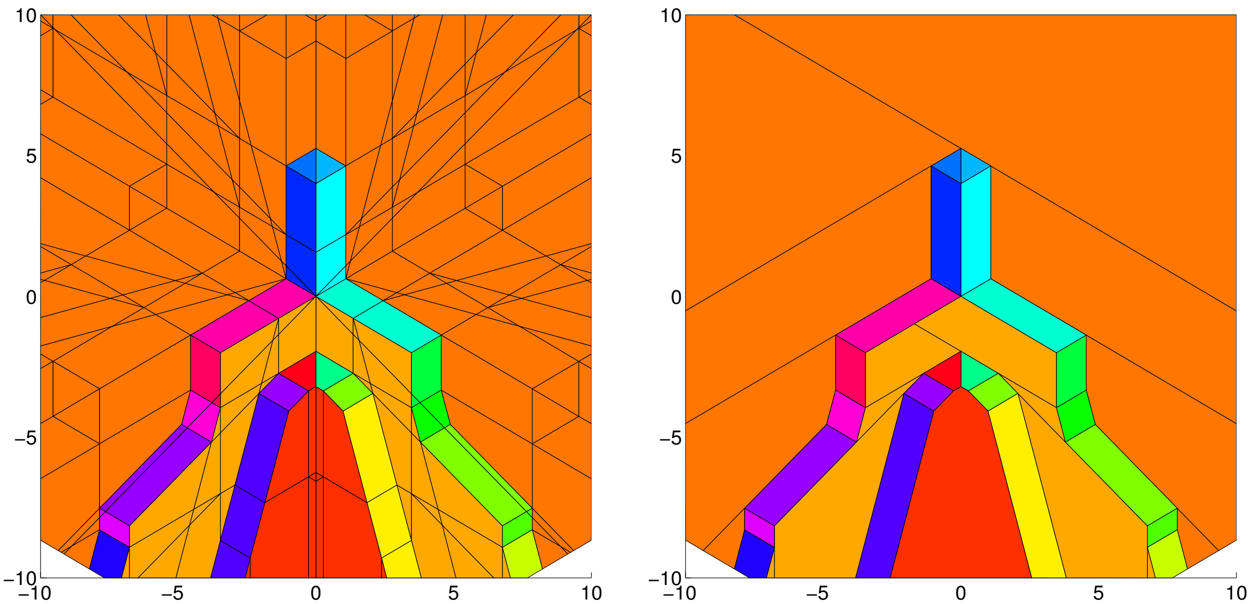
If the aim is to derive the *optimal* solution, we need to compute the global hyperplane arrangement by extending the facets of the polyhedra. Due to the lack of space, we give here only a brief outline of such an algorithm which consists of three major steps. First, we collect the facets of all polyhedra. By removing duplicates, we obtain the hyperplane arrangement. Next, we determine the relative position of each polyhedron with respect to each hyperplane. This yields a preliminary set of markings where we use an additional symbol to denote polyhedra whose interior intersects with a hyperplane. The algorithm resolves these markings in a last step by dividing the corresponding polyhedra into two. As this operation involves solving LPs and increases the number of polyhedra significantly, such an algorithm is computational tractable only for problems with a limited complexity. However, a number of enhancements, namely the exploitation of parallel hyperplanes and the removal of redundant hyperplanes reduces the computation time remarkably. We will refer to this approach as Algorithm 2.

IV. EXAMPLE

In this final section we show how the complexity reduction algorithm can be extended to cases for which the hyperplane arrangement and thus the markings are missing. These problems include the computation of the feedback law of MPC controllers for linear or hybrid systems. Apart from the case of hybrid systems with quadratic performance indices in the cost function, the resulting feedback laws are again PWA expressions defined on polyhedral partitions. When implementing the controller online, we require a feedback law whose corresponding number of polyhedra is as low as possible in order to reduce the memory requirements and the computational burden for the controller hardware. Thus we are interested in deriving an equivalent PWA feedback law minimal in the number of polyhedra by merging polyhedra associated with the same feedback law. In order to tackle this problem, we first need to compute the global hyperplane arrangement by extending the facets defining the polyhedra.

As an example, consider a simple PWA model with two real states and two modes for which the authors in [2] have formulated a constrained infinite time optimal control problem. The resulting polyhedral partition of the state space is shown in Fig. 3(a), where the affine control laws are indicated by different colors. Note that there exist 19 different control laws and 252 polyhedra. Algorithm 2 derives a hyperplane arrangement with 135 hyperplanes containing 5200 polyhedra. The optimal merging approach leads to 39 polyhedra which are shown in Fig. 3(b). Compared to the initial 252 polyhedra, this is a reduction of 84 percent. The total computation time amounts to 4.5 min running MATLAB 6.5 on a Pentium IV 2.8 GHz machine, where deriving the hyperplane arrangement requires 2.5 min and merging takes roughly 2 min.

As the rather large computation time indicates, the optimal complexity reduction algorithm as proposed here is only applicable to PWA feedback laws with a limited number of polyhedra defined in a low-dimensional space. The main problem is the major increase in the number



(a) The original 252 polyhedra.

(b) The merged 39 polyhedra.

Fig. 3. Polyhedral partitions of the PWA feedback law, where each color relates to a different affine feedback law (19 laws in total).

of polyhedra when deriving the hyperplane arrangement (here from 252 to 5200 polyhedra). However, we would like to stress that Algorithm 1 is capable of deriving successfully the minimal representation of thousands of polyhedra. In this example, it reduces 5200 polyhedra to 39 within approximately 2 min.

Another example concerning a PWA model resulting from the mode enumeration algorithm can be found in [10].

V. Conclusions

Based on the results of the mode enumeration algorithm [9] that transforms a hybrid system into a PWA system, and exploiting the associated markings of the hyperplane arrangement allows us to build an equivalent PWA system minimal in the number of polyhedra of the partition by using branch and bound techniques. As we refrain from solving LPs, the algorithm is both optimal and computationally tractable. By computing the (global) hyperplane arrangement, the applicability of the algorithm can be extended to derive minimal PWA representations of general PWA models and control laws. Future research will be devoted to develop fast and suboptimal algorithms.

The MATLAB code is available upon request from the authors. An extended version of this paper with additional examples, remarks and definitions can be found in [10].

Acknowledgements

This work was partially supported by the EU project *Control and Computation*. The authors would like to thank Prof. Komei Fukuda for sharing his tools and Mato Baotić for fruitful discussions.

References

[1] D. Avis and K. Fukuda. Reverse search for enumeration. *Discr. App. Math.*, 65:21–46, 1996.

[2] M. Baotić, F.J. Christophersen, and M. Morari. Infinite time optimal control of hybrid systems with a linear performance index. In *Proc. 42th IEEE Conf. on Decision and Control*, pages 3191–3196, 2003.

[3] A. Bemporad, K. Fukuda, and F.D. Torrisi. Convexity recognition of the union of polyhedra. *Comp. Geometry: Theory and Applications*, 18:141–154, April 2001.

[4] F. Borrelli. *Constrained Optimal Control of Linear and Hybrid Systems*, volume 290 of *LNCIS*. Springer, 2003.

[5] R.C. Buck. Partition of space. *American Math. Monthly*, 50:541–544, 1943.

[6] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.

[7] J.A. Ferrez and K. Fukuda. Implementations of lp-based reverse search algorithms for the zonotope construction and the fixed-rank convex quadratic maximization in binary variables using the ZRAM and the cddlib libraries. Technical report, Mcgill, `http://www.cs.mcgill.ca/~fukuda/download/mink/RS_TOPE020713.tar.gz`, July 2002.

[8] J.A. Ferrez, K. Fukuda, and Th.M. Liebling. Cuts, zonotopes and arrangements. Technical report, EPF Lausanne, Switzerland, November 2001.

[9] T. Geyer, F.D. Torrisi, and M. Morari. Efficient mode enumeration of compositional hybrid systems. In A. Pnueli and O. Maler, editors, *Hybrid Systems: Computation and Control*, volume 2623 of *LNCS*, pages 216–232. Springer-Verlag, 2003.

[10] T. Geyer, F.D. Torrisi, and M. Morari. Optimal complexity reduction of piecewise affine models based on hyperplane arrangements. Technical Report AUT04-01, Automatic Control Laboratory ETH Zurich, `http://control.ee.ethz.ch`, 2004.

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

[12] E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. Automat. Contr.*, 26(2):346–358, April 1981.

[13] F.D. Torrisi and A. Bemporad. Hysdel — a tool for generating computational hybrid models for analysis and synthesis problems. Technical Report AUT02-03, Automatic Control Laboratory ETH Zurich, `http://control.ee.ethz.ch`, 2002. To appear in IEEE Control Systems Technology.

[14] G. M. Ziegler. *Lectures on Polytopes*. Springer, 1994.