

Convex Liftings: Theory and Control Applications

Ngoc Anh Nguyen, Martin Gulan, Sorin Olaru, Pedro Rodriguez-Ayerbe

▶ **To cite this version:**

Ngoc Anh Nguyen, Martin Gulan, Sorin Olaru, Pedro Rodriguez-Ayerbe. Convex Liftings: Theory and Control Applications. [Research Report] CentraleSupelec. 2016. hal-01326804v1

HAL Id: hal-01326804

https://centralesupelec.hal.science/hal-01326804v1

Submitted on 5 Jun 2016 (v1), last revised 7 Aug 2017 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Convex Liftings: Theory and Control Applications

N. A. Nguyen[1], M. Gulan[2], S. Olaru[3], P. Rodriguez-Ayerbe[3]

***Abstract*—This paper presents the so-called *convex lifting* concept which will be proven to enable considerable implementation benefits for the class of piecewise affine controllers. Accordingly, two different algorithms to construct a convex lifting for a given polyhedral/polytopic partition will be presented. These two algorithms rely on either the vertex or the halfspace representations of the related polyhedra. Also, we introduce an algorithm to refine a polyhedral partition, which does not admit a convex lifting, into a convexly liftable one. Furthermore, two different schemes will be put forward to significantly reduce both the memory footprint and the runtime complexity which play a key role in implementation of piecewise affine controllers. Finally, these results will be illustrated via a numerical example.**

***Index Terms*—Convex lifting, model predictive control, explicit solutions, parametric programming.**

I. Motivation

Explicit model predictive control (MPC) has received significant attention in control community due to its relevance for rather small-dimensional systems [9]. However, even if controllers are explicitly obtained, there exist major problems in terms of implementation once the number of regions in the state-space partition becomes large. In particular, they require storing all the regions at the hardware level, making their implementation, namely on embedded computing platforms, difficult due to their limited memory storage and computational performance.

Different efficient implementation algorithms have been put forward [20], [21], however the requirement of substantial memory for storing the given partition is inevitable. Another contribution about efficient storage strategy has been presented in [6]. This proposal can avoid storing the state-space partition, however the point-location problem, determining which region the current state belongs to, becomes more demanding, see among the other point-location algorithms [7], [8], [35]. Therefore, it is necessary to investigate other implementation approaches for this class of controllers which can avoid storing the state-space partition and possibly facilitate the point-location problem. This work presents a convex lifting concept which allows for efficient piecewise affine (PWA) controllers' implementation.

For ease of presentation, let us start with some special cases of parametric linear programming problem where the optimal cost function presents an interesting property. To illustrate these, consider a linear MPC problem with respect to a linear cost function. Such a problem can be easily transformed into a parametric linear programming problem as follows:

$$u^*(x) = \arg\min_u C^T u \quad \text{subject to} \quad Gu \leq W + Ex. \tag{1}$$

It has been shown through Theorems IV-3 and IV-4 in [15] that $C^T u^*(x)$ is a convex, continuous, PWA function defined over a polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, where $\mathcal{I}_N = \{1, 2, \ldots, N\}$. Let us denote the optimal cost function and optimal solution of (1) as follows:

$$\begin{aligned} C^T u^*(x) &= a_i^T x + b_i \quad \text{for} \quad x \in \mathcal{X}_i, \\ u^*(x) &= H_i x + G_i \quad \text{for} \quad x \in \mathcal{X}_i. \end{aligned} \tag{2}$$

Since $C^T u^*(x)$ is also a convex function, this optimal cost function can be alternatively written in the following form:

$$C^T u^*(x) = \max_{j\in\mathcal{I}_N} (a_j^T x + b_j). \tag{3}$$

Accordingly, as advocated in [6], if the optimal solution to the parametric linear programming problem (1) is ***unique***, then implementation of the optimal control law $u^*(x)$ can be carried out according to Algorithm 1.

Algorithm 1 Efficient implementation of PWA controllers

1: Store (H_i, G_i) and (a_i, b_i)
2: At each sampling time, measure the state x
3: Find index $i \in \mathcal{I}_N$ such that:

$$a_i^T x + b_i = \max_{j\in\mathcal{I}_N} (a_j^T x + b_j).$$

4: Evaluate controller $u^*(x) = H_i x + G_i$.
5: Return to step 2.

Significant advantages of this implementation are to avoid storing the state-space partition and facilitate the point-location problem. However, as emphasized above, this implementation only holds if the optimal solution to (1) is unique, because in this case for any pair of different regions $(\mathcal{X}_i, \mathcal{X}_j)$, their optimal cost function satisfies $(a_i, b_i) \neq (a_j, b_j)$. Note also that in case the uniqueness of the optimal solution to (1) is fulfilled, the optimal cost function of the parametric linear programming problem (1) is, from a geometrical point of view, nothing other than a convex lifting associated with the state-space partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ (this will be formally proven later in Theorem II.7), since a convex lifting is defined as a convex, continuous, piecewise affine function defined over a polyhedral partition such that any pair of different regions are lifted onto two distinct hyperplanes. Otherwise, so far, in case optimal solution to (1) is not unique or the state-space partition resulted

[1]N. A. Nguyen is with Institute for Design and Control of Mechatronical Systems, Johannes Kepler University, Linz, Austria. Ngocanh.Nguyen.rs@gmail.com

[2]M. Gulan is with Institute of Automation, Measurement and Applied Informatics, Faculty of Mechanical Engineering, Slovak University of Technology, Bratislava, Slovakia. martin.gulan@stuba.sk

[3]S. Olaru, P. Rodriguez-Ayerbe are with Laboratory of Signals and Systems (L2S, UMR CNRS 8506), CentraleSupélec-CNRS-Université Paris Sud, Gif-sur-Yvette, France. Sorin.Olaru, Pedro.Rodriguez@centralsupelec.fr

from a linear MPC problem with respect to a quadratic cost function, this implementation is no longer applicable, even if the state-space partition admits a convex lifting. Motivated by this limitation, control theory related to the PWA control laws lacks:

- algorithms to verify whether the given state-space partition admits convex liftings,
- algorithms to construct convex liftings for the given state-space partition if they exist,
- algorithms to refine a partition, which does not admit a convex lifting (henceforth referred to as a convexly non-liftable partition), into one admitting a convex lifting (convexly liftable partition).

Accordingly, if the algorithms above are available, then Algorithm 1 will be directly applicable for implementation of PWA controllers obtained from both parametric linear and quadratic programming problems. This would bring a significant interest and allow the piecewise affine control laws to be implemented on embedded platforms with low computational performance and memory storage. In this paper, all these problems will be addressed.

II. Preliminaries

$\mathbb{R}, \mathbb{R}_+, \mathbb{N}_{>0}$ denote the field of real numbers, the set of non-negative real numbers and the positive integer set, respectively.

Given an arbitrary set $\mathcal{S}$, $\text{conv}(\mathcal{S})$ denotes the convex hull of $\mathcal{S}$; $\text{aff}(\mathcal{S})$ denotes the affine hull of $\mathcal{S}$. Also, $\dim(\mathcal{S})$ stands for the dimension of $\text{aff}(\mathcal{S})$. If $\mathcal{S}$ is a full-dimensional set, then $\text{int}(\mathcal{S})$ denotes its interior. Given a set $\mathcal{S} \subseteq \mathbb{R}^d$ and a subspace $\mathbb{S}$ of $\mathbb{R}^d$, then $\text{Proj}_{\mathbb{S}}\mathcal{S}$ denotes the orthogonal projection of $\mathcal{S}$ onto the space $\mathbb{S}$.

A polyhedron is defined as the intersection of finitely many closed halfspaces. A polytope is defined as a bounded polyhedron. Given a polyhedron $\mathcal{S}$, we use $\mathcal{V}(\mathcal{S})$ to denote the set of its vertices and $\mathcal{R}(\mathcal{S})$ denotes the set of its extreme rays. Further, if $\mathcal{S} \subseteq \mathbb{R}^d$ is a full-dimensional polyhedron, a face of $\mathcal{S}$ is the intersection of $\mathcal{S}$ and one of its supporting hyperplanes. $k-$face represents a face of dimension k. A $0-$face is called a vertex, an $1-$face is called an edge, a $(d-1)-$face is called a facet. Also, $\mathcal{F}(\mathcal{S})$ denotes the set of all facets of the polyhedron $\mathcal{S}$.

Given two sets S_1, S_2, we use $S_1 \backslash S_2$ to denote the following set: $S_1 \backslash S_2 := \{x : x \in S_1,\ x \notin S_2\}$.

Let us recall also some useful definitions.

Definition II.1 A collection of $N \in \mathbb{N}_{>0}$ full-dimensional polyhedra $\mathcal{X}_i \subset \mathbb{R}^d$, denoted by $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, is called a *polyhedral partition of a polyhedron* $\mathcal{X} \subseteq \mathbb{R}^d$ if:

1) $\mathcal{X} = \bigcup_{i\in\mathcal{I}_N} \mathcal{X}_i$,
2) $\text{int}(\mathcal{X}_i) \bigcap \text{int}(\mathcal{X}_j) = \emptyset$ with $i \neq j,\ (i,j) \in \mathcal{I}_N^2$,

$(\mathcal{X}_i, \mathcal{X}_j)$ are called neighbors or adjacent if $(i,j) \in \mathcal{I}_N^2$, $i \neq j$ and $\dim(\mathcal{X}_i \cap \mathcal{X}_j) = d-1$. Also, if $\mathcal{X}$ is a polytope then $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ is called *polytopic partition*.

In case $\mathcal{X}$ is not a polyhedron, $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ is still called a polyhedral/polytopic partition but of a *nonconvex polyhedral set.*

The definition of a *cell complex* was presented by Grünbaum in [16]. For simplicity, a *cell complex* should be hereafter understood as a polyhedral partition whose face-to-face property is fulfilled, i.e., for any pair of regions, the intersection of faces is either empty or a common face. Also, if $\mathcal{X}$ is a polyhedron, then a cell complex of $\mathcal{X}$ is understood as a polyhedral partition whose facet-to-facet property is satisfied, meaning any pair of neighboring regions share a common facet. For illustration, the polytopic partition in Fig.1 is a cell complex.

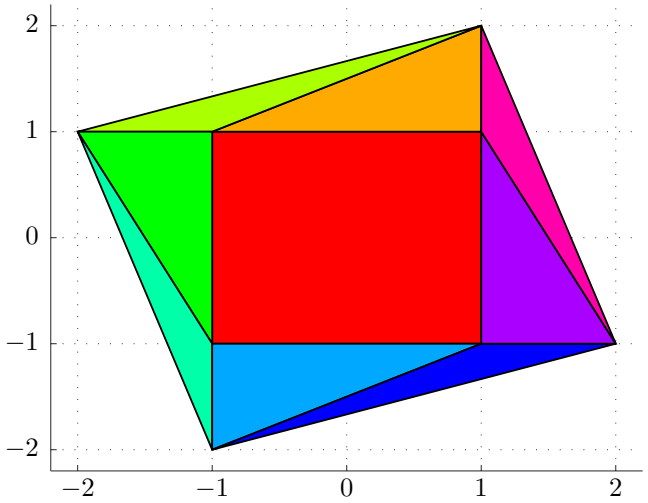


Fig. 1: A cell complex of a polytope in $\mathbb{R}^2$.

Definition II.2 For a given polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, a *piecewise affine lifting* is described by a function $z : \mathcal{X} \to \mathbb{R}$ with:

$$z(x) = a_i^T x + b_i \quad \text{for any } x \in \mathcal{X}_i, \tag{4}$$

and $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$, $\forall i \in \mathcal{I}_N$.

Definition II.3 Given a polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, a piecewise affine lifting

$$z(x) = a_i^T x + b_i \text{ for } x \in \mathcal{X}_i,$$

is called a *convex piecewise affine lifting* if the following conditions hold true:

- $z(x)$ is continuous over $\mathcal{X}$,
- for each $i \in \mathcal{I}_N$, $z(x) > a_j^T x + b_j$ for all $x \in \mathcal{X}_i \backslash \mathcal{X}_j$ and all $j \neq i$, $j \in \mathcal{I}_N$.

The second condition in the above definition implies that $z(x)$ is a convex function defined over $\mathcal{X}$. Moreover, the strict inequalities ensure that any pair of neighboring regions are lifted onto two distinct hyperplanes. For ease of presentation, a slight abuse of notation is used henceforth: a *convex lifting* is understood as a convex piecewise affine lifting.

From the above definition, if a polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ admits a convex lifting, then it has to be a cell complex. This observation is stated via the following proposition.

Proposition II.4 *A polyhedral partition of a polyhedron, which admits a convex lifting, is a cell complex.*

Proof: Suppose the given polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X}\subseteq\mathbb{R}^d$, which admits a convex lifting, is not a cell complex. Let $z(x)=a_i^Tx+b_i$ for $x\in\mathcal{X}_i$ denote this convex lifting of $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$. Then there exists a pair of neighboring regions, denoted by $\mathcal{X}_i,\mathcal{X}_j$, whose facet-to-facet property is not fulfilled.

Following the definition of convex liftings, the hyperplane denoted by $\mathcal{H}_0$, containing $\mathcal{X}_i\cap\mathcal{X}_j$, can be described by

$$\mathcal{H}_0=\left\{x\in\mathbb{R}^d : a_i^Tx+b_i=a_j^Tx+b_j\right\}.$$

Also, due to the violation of the facet-to-facet property, there exists a point, denoted by x_0, such that $x_0\in\mathcal{H}_0\cap\mathcal{X}_i$ but $x_0\notin\mathcal{X}_j$. $x_0\in\mathcal{H}_0$ implies

$$a_i^Tx_0+b_i=a_j^Tx_0+b_j. \tag{5}$$

On the other hand, $x_0\in\mathcal{X}_i, x_0\notin\mathcal{X}_j$ lead to

$$a_i^Tx_0+b_i>a_j^Tx_0+b_j. \tag{6}$$

The inclusions (5) and (6) are clearly contradictory. Therefore, the partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ has to be a cell complex. ■

According to Proposition II.4, a convex lifting is always defined over a cell complex. However, the cell complex characterization of $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ is a necessary condition for the existence of a convex lifting, but not a sufficient one.

Remark II.5 Note also that Proposition II.4 does not necessarily restrict $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ to polyhedral partition of a polyhedron. In other words, a polyhedral partition of a suitable set $\mathcal{X}\subseteq\mathbb{R}^d$, which admits a convex lifting, should also be a cell complex.

Definition II.6 A given cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ in $\mathbb{R}^d$ has an *affinely equivalent polyhedron* if there exists a polyhedron $\widetilde{\mathcal{X}}\subset\mathbb{R}^{d+1}$ such that for each $i\in\mathcal{I}_N$:

1) $\exists F_i\in\mathcal{F}(\widetilde{\mathcal{X}})$ satisfying: $\mathrm{Proj}_{\mathbb{R}^d}F_i=\mathcal{X}_i$,
2) if $\underline{z}(x)=\min\limits_z z$ s.t. $\begin{bmatrix}x^T & z\end{bmatrix}^T\in\widetilde{\mathcal{X}}$, then $\begin{bmatrix}x\\ \underline{z}(x)\end{bmatrix}\in F_i$ for $x\in\mathcal{X}_i$.

An illustration can be found in Fig.2 where a cell complex in $\mathbb{R}$ consists of the multicolored segments along the horizontal axis. One of its affinely equivalent polyhedra in $\mathbb{R}^2$ is the pink shaded region. Moreover, the lower facets of this polytope are an illustration of the facets F_i appearing in Definition II.6. Notice that given a polyhedron $\widetilde{\mathcal{X}}\subset\mathbb{R}^{d+1}$, if z denotes the last coordinate of $\widetilde{\mathcal{X}}$ such that $\begin{bmatrix}x^T & z\end{bmatrix}^T\in\widetilde{\mathcal{X}}$, then the optimal solution to the following parametric linear programming problem:

$$z^*(x)=\min_z z \text{ subject to } \begin{bmatrix}x^T & z\end{bmatrix}^T\in\widetilde{\mathcal{X}}. \tag{7}$$

is nothing other than a convex lifting for the cell complex associated with this optimal solution. This observation will be proven in the sequel.

First, consider the parametric linear programming problem (1) and its optimal solution (2). We will prove that if the optimal solution to (1) is unique, then the optimal cost function of (1) is a convex lifting for the polyhedral partition associated with the optimal solution (2).

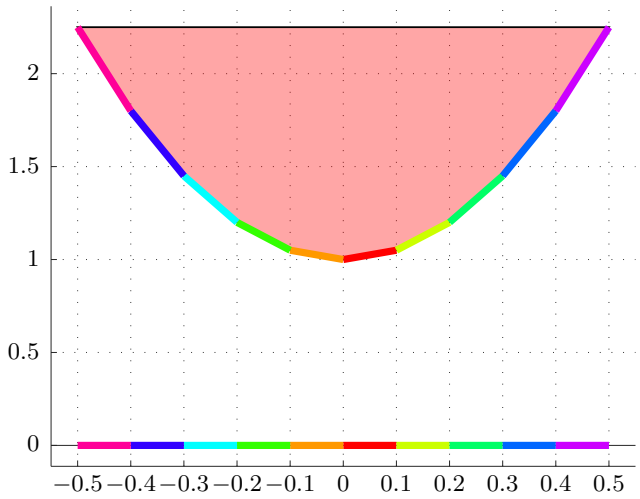


Fig. 2: An illustration of an affinely equivalent polyhedron.

Theorem II.7 *If the optimal solution to the parametric linear programming problem* (1) *is unique, then the optimal cost function $C^Tu^*(x)$ is a convex lifting for the polyhedral partition associated with $u^*(x)$.*

Proof: According to Theorems IV-3 and IV-4 in [15], $C^Tu^*(x)$ is a convex and continuous piecewise affine function defined over the polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, associated with the optimal solution $u^*(x)$. Accordingly, to prove $C^Tu^*(x)$ is a convex lifting for $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, it suffices to show that for any pair of different regions $(\mathcal{X}_i,\mathcal{X}_j)$,

$$(C^TH_i,C^TG_i)\neq(C^TH_j,C^TG_j).$$

Suppose the converse situation happens, i.e., there exist two different regions $(\mathcal{X}_i,\mathcal{X}_j)$, $i\neq j$, $(i,j)\in\mathcal{I}_N^2$ such that

$$(H_i,G_i)\neq(H_j,G_j),\ (C^TH_i,C^TG_i)=(C^TH_j,C^TG_j).$$

Consider $x_1\in\mathrm{int}(\mathcal{X}_i)$, $x_2\in\mathcal{X}_j$ and a scalar $\alpha\in[0,1]$. Due to the convexity of $C^Tu^*(x)$, we can see that

$$\begin{aligned}&C^Tu^*(\alpha x_1+(1-\alpha)x_2)\\ &\quad\leq\alpha C^T(H_ix_1+G_i)+(1-\alpha)C^T(H_jx_2+G_j).\end{aligned} \tag{8}$$

If we choose α close enough to 1 such that $\alpha x_1+(1-\alpha)x_2\in\mathcal{X}_i$, then

$$C^Tu^*(\alpha x_1+(1-\alpha)x_2)=C^T(H_i(\alpha x_1+(1-\alpha)x_2)+G_i). \tag{9}$$

Note also that according to the assumption $(C^TH_i,C^TG_i)=(C^TH_j,C^TG_j)$, it follows that

$$\begin{aligned}&\alpha C^T(H_ix_1+G_i)+(1-\alpha)C^T(H_jx_2+G_j)\\ &=C^TH_i(\alpha x_1+(1-\alpha)x_2)+C^TG_i.\end{aligned} \tag{10}$$

Also, since $H_ix_1+G_i, H_jx_2+G_j$ satisfy the constraint set in (1), so does $\alpha(H_ix_1+G_i)+(1-\alpha)(H_jx_2+G_j)$. According to (8), (9), (10), $\alpha(H_ix_1+G_i)+(1-\alpha)(H_jx_2+G_j)$ is also an

optimal solution to (1). Due to the uniqueness of the optimal solution to (1), we obtain the following:

$$H_i(\alpha x_1+(1-\alpha)x_2)+G_i = \alpha(H_i x_1+G_i)+(1-\alpha)(H_j x_2+G_j),$$

leading to

$$H_i x_2 + G_i = H_j x_2 + G_j. \tag{11}$$

It is worth emphasizing that (11) holds true for all $x_2 \in \mathcal{X}_j$. Since $(H_i, G_i) \neq (H_j, G_j)$, the set of $x \in \mathbb{R}^d$ satisfying $H_i x + G_i = H_j x + G_j$ represents a polyhedron of dimension lower than d, while $\mathcal{X}_j$ is a full-dimensional polyhedron in $\mathbb{R}^d$. This is clearly contradictory. Therefore, the initial hypothesis is not true. In other words, if the optimal solution to (1) is unique, the optimal cost function $C^T u^*(x)$ describes a convex lifting for the associated polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$. ■

Now, let us come back to prove that the optimal solution to (7) stands for a convex lifting of the associated polyhedral partition.

Lemma II.8 *Given the parametric linear programming problem* (7)*, if $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ denotes the polyhedral partition associated with $z^*(x)$, then $z^*(x)$ is a convex lifting for $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$.*

Proof: First, we will prove that the optimal solution to (7) is unique. Indeed, suppose there exist two optimal solutions to (7), denoted by $z_1^*(x)$ and $z_2^*(x)$, respectively. Without loss of generality, suppose $z_1^*(x), z_2^*(x)$ are defined on the same polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$. Consider a region $\mathcal{X}_i$ in this polyhedral partition and denote these optimal solutions over $\mathcal{X}_i$ as follows:

$$z_1^*(x) = (a_i^{(1)})^T x + b_i^{(1)} \quad \text{for} \quad x \in \mathcal{X}_i,$$
$$z_2^*(x) = (a_i^{(2)})^T x + b_i^{(2)} \quad \text{for} \quad x \in \mathcal{X}_i.$$

Since $z_1^*(x), z_2^*(x)$ represent optimal cost function of (7), we thus obtain:

$$(a_i^{(1)})^T x + b_i^{(1)} = (a_i^{(2)})^T x + b_i^{(2)}. \tag{12}$$

Note that (12) holds for all $x \in \mathcal{X}_i$, as a full-dimensional polyhedron. Accordingly, this case holds true only if $(a_i^{(1)}, b_i^{(1)}) = (a_i^{(2)}, b_i^{(2)})$. In other words, the optimal solution to (7) is unique. According to Theorem II.7, the optimal cost function of (7) stands for a convex lifting of $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$. ■

III. Constructions of convex liftings

A. Existing results on convex liftings

The definition of a convex lifting has been presented earlier. In control theory, so far, convex liftings have been used to solve the inverse parametric linear/quadratic programming problem [24]–[31].

Many necessary and sufficient conditions for the existence of convex liftings for cell complexes were investigated in different studies [2]–[4], [12], [13], [23], [32], [34]. It is shown in [32] that there exists a convex lifting for a cell complex in $\mathbb{R}^d$ if and only if one of the following holds:

- it admits a strictly positive $d-$stress;
- it is an additively weighted Dirichlet-Voronoi diagram;
- it is an additively weighted Delaunay decomposition;
- it is the section of a $(d+1)$-dimensional Dirichlet-Voronoi partition[1].

Interested readers are referred to [24] for details of the above notation. Note that the above results cover the general class of cell complexes in $\mathbb{R}^d$. Unfortunately, despite the mathematical completeness of the existing results, the verification of these conditions are expensive. Furthermore, they do not provide any hint for the construction of a convex lifting. On the other hand, applications in control theory need specific algorithms to verify the convex liftability of the cell complexes and construct their convex liftings whenever these exist. These elements are detailed in the coming subsections. We remark that the construction of convex liftings for some special cases, e.g., Voronoi diagrams and Delaunay triangulations and their recognition were already investigated in [4], [14], [18].

B. Construction of convex liftings based on the vertex representation

In this subsection, the main objective is to present an algorithm for the construction of a convex lifting for a given cell complex via linear/quadratic programming. This algorithm exploits the continuity and the convexity of a convex lifting for two neighboring regions. This construction is limited to polytopic partitions since it hinges on suitable constraints imposed at the vertices of this partition. Extension of this construction to polyhedral partitions of unbounded polyhedra can be found in [26].

Given a cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polytope $\mathcal{X} \subset \mathbb{R}^d$, $\widetilde{\mathcal{X}} \subset \mathbb{R}^{d+1}$ denotes an affinely equivalent polyhedron of $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$. For each region $\mathcal{X}_i$, $i \in \mathcal{I}_N$, the hyperplane, containing the lower facet of $\widetilde{\mathcal{X}}$ whose orthogonal projection onto $\mathbb{R}^d$ coincides with $\mathcal{X}_i$, has the following form:

$$\mathcal{H}_i = \left\{ \begin{bmatrix} x \\ z_i(x) \end{bmatrix} \in \mathbb{R}^{d+1} : z_i(x) = a_i^T x + b_i \right\}, \tag{13}$$

for suitable $a_i \in \mathbb{R}^d, b_i \in \mathbb{R}$.

Let $(i,j) \in \mathcal{I}_N^2$ be an index pair such that $(\mathcal{X}_i, \mathcal{X}_j)$ are neighbors. The *continuity conditions* between them are described as follows:

$$\forall x \in \mathcal{X}_i \cap \mathcal{X}_j,\ z_i(x) = z_j(x),\ i \neq j. \tag{14}$$

In addition, the *convexity conditions* between them can be handled as:

$$\forall x \in \mathcal{X}_i \backslash \mathcal{X}_j,\ z_i(x) > z_j(x). \tag{15}$$

The conditions (14) and (15) represent the fundamental properties of a convex lifting, therefore they can be used for its construction by considering (a_i, b_i) as variables. Algorithm 2 summarizes such a constructive procedure which allows for the computation of the gains (a_i, b_i), $\forall i \in \mathcal{I}_N$ of a convex lifting for the given polytopic partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$.

Now, we step by step prove that the feasibility of the optimization problem (18) serves as another necessary and

[1]Other related results can be found in Konstantin Rybnikov's thesis [32], equally in [2]–[4]. Note that an additively weighted Dirichlet-Voronoi diagram is in fact a generalization of a power diagram.

Algorithm 2 Construction of a convex lifting for a given cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polytope $\mathcal{X}\subset\mathbb{R}^d$.

Input: $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ and a given constant $c>0$.
Output: $(a_i,\, b_i)$, $\forall i\in\mathcal{I}_N$.

1: Register all pairs of neighboring regions in $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$.
2: For each pair of neighboring regions $(\mathcal{X}_i,\,\mathcal{X}_j)$, $(i,\,j)\in\mathcal{I}_N^2$:
- add continuity conditions $\forall v\in\mathcal{V}(\mathcal{X}_i\cap\mathcal{X}_j)$:
$$a_i^Tv+b_i=a_j^Tv+b_j; \tag{16}$$
- add convexity conditions $\forall u\in\mathcal{V}(\mathcal{X}_i), u\notin\mathcal{V}(\mathcal{X}_i\cap\mathcal{X}_j)$:
$$a_i^Tu+b_i\geq a_j^Tu+b_j+c. \tag{17}$$

3: Solve the following convex optimization problem by minimizing a chosen cost function, e.g.,
$$\min_{a_i,\,b_i}\ \sum_{i=1}^{N}(a_i^Ta_i+b_i^Tb_i)\ \text{ subject to }\ (16),(17). \tag{18}$$

sufficient condition for the convex liftability of the given polytopic partition of a polytope.

Proposition III.1 *If problem* (18) *is feasible, then function*
$$z(x)=a_i^Tx+b_i\ \ \textit{for}\ \ x\in\mathcal{X}_i$$
is a convex lifting over the given cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$.

Proof: If the optimization problem (18) is feasible, then the continuity conditions of function $z(x)$ and the convexity conditions of $z(x)$ are all fulfilled. Accordingly, for two neighboring regions $(\mathcal{X}_i,\mathcal{X}_j)$, it follows that:
$$a_i^Tv+b_i=a_j^Tv+b_j\ \text{ for all }\ v\in\mathcal{V}(\mathcal{X}_i\cap\mathcal{X}_j),$$
$$a_i^Tu+b_i\geq a_j^Tu+b_j+c\ \text{ for all }\ u\in\mathcal{V}(\mathcal{X}_i)\backslash\mathcal{V}(\mathcal{X}_j).$$
Therefore, any point $x=\sum_{v\in\mathcal{V}(\mathcal{X}_i\cap\mathcal{X}_j)}\alpha(v)v$ with $\sum_{v\in\mathcal{V}(\mathcal{X}_i\cap\mathcal{X}_j)}\alpha(v)=1$ and $\alpha(v)\in\mathbb{R}$, satisfies
$$a_i^Tx+b_i=a_j^Tx+b_j. \tag{19}$$
In other words, any point $x\in\text{aff}(\mathcal{X}_i\cap\mathcal{X}_j)$ satisfies (19). Furthermore, any point $x=\sum_{v\in\mathcal{V}(\mathcal{X}_i)}\alpha(v)v$ with $\sum_{v\in\mathcal{V}(\mathcal{X}_i)}\alpha(v)=1$, $\alpha(v)\in\mathbb{R}$,
$$a_i^Tx+b_i>a_j^Tx+b_j\ \text{ for }\ \alpha(v)\geq 0\ \forall v\in\mathcal{V}(\mathcal{X}_i)\backslash\mathcal{V}(\mathcal{X}_j)$$
$$\{\alpha(v):v\in\mathcal{V}(\mathcal{X}_i)\backslash\mathcal{V}(\mathcal{X}_j)\}\neq\{0:v\in\mathcal{V}(\mathcal{X}_i)\backslash\mathcal{V}(\mathcal{X}_j)\}. \tag{20}$$
Strictly speaking, any point x in the halfspace containing $\mathcal{X}_i$ but $x\notin\text{aff}(\mathcal{X}_i\cap\mathcal{X}_j)$, satisfies (20).

Similarly, the same relations (19) and (20) hold for the other pairs of neighboring regions. This leads to the continuity of $z(x)$ and for each $i\in\mathcal{I}_N$:
$$a_i^Tx+b_i>a_j^Tx+b_j\ \text{ for all }\ x\in\mathcal{X}_i\backslash\mathcal{X}_j,\ \forall j\neq i,\ j\in\mathcal{I}_N.$$
Therefore, function $z(x)=a_i^Tx+b_i$ for $x\in\mathcal{X}_i$ is a convex lifting defined over the cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, as defined in Definition II.3. ■

Note that the cost function in (18) is chosen to avoid the unboundedness of optimal solution. Other choices of this cost function are possible as long as the boundedness of optimal solution is guaranteed. Also, as seen in (17), the strict inequality (15) can be easily transformed into inequality constraints in an optimization problem by adding a positive constant c on the right-hand side of (17), thus $>$ can be replaced with $\geq$. Theoretically, if the given cell complex is convexly liftable, then any positive value of c does not have any effect on the feasibility of the optimization problem (18). This observation is proven in the sequel.

Proposition III.2 *Given a cell complex* $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ *of a polyhedron* $\mathcal{X}\subseteq\mathbb{R}^d$, *if* $z(x)=a_i^Tx+b_i$ *for* $x\in\mathcal{X}_i$ *is a convex lifting for this cell complex, then so is* $\tilde{z}(x)=(\alpha a_i)^Tx+(\alpha b_i)+\beta$ *for* $x\in\mathcal{X}_i$, *for any* $\alpha>0,\beta\in\mathbb{R}$.

Proof: In fact, if $z(x)$ represents a convex lifting for the given cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, according to the definition of a convex lifting, for each pair of neighboring regions $(\mathcal{X}_i,\mathcal{X}_j)$, it follows that:
$$\begin{aligned}&a_i^Tx+b_i=a_j^Tx+b_j\ \ \text{for}\ \ x\in\mathcal{X}_i\cap\mathcal{X}_j\\&a_i^Tx+b_i>a_j^Tx+b_j\ \ \text{for}\ \ x\in\mathcal{X}_i\backslash\mathcal{X}_j.\end{aligned} \tag{21}$$
Accordingly, for any $\alpha>0,\beta\in\mathbb{R}$, (21) amounts to:
$$\begin{aligned}&(\alpha a_i)^Tx+\alpha b_i+\beta=(\alpha a_j)^Tx+\alpha b_j+\beta\ \ \text{for}\ \ x\in\mathcal{X}_i\cap\mathcal{X}_j,\\&(\alpha a_i)^Tx+\alpha b_i+\beta>(\alpha a_j)^Tx+\alpha b_j+\beta\ \ \text{for}\ \ x\in\mathcal{X}_i\backslash\mathcal{X}_j.\end{aligned} \tag{22}$$
Inclusion (22) means that $\tilde{z}(x)=(\alpha a_i)^Tx+(\alpha b_i)+\beta$ for $x\in\mathcal{X}_i$ is also a convex lifting for the given cell complex for any $\alpha>0,\beta\in\mathbb{R}$. ■

We now prove that the feasibility of the optimization problem (18) serves as a necessary and sufficient condition for the convex liftability of the given polytopic partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of polytopes.

Theorem III.3 *The given polytopic partition* $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ *of a polytope* $\mathcal{X}$, *is convexly liftable if and only if the optimization problem* (18) *is feasible for any constant* $c>0$.

Proof: $\longleftarrow$ This inclusion directly follows according to Proposition III.1.
$\longrightarrow$ If the given polytopic partition, denoted by $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, is convexly liftable, then there exists a constant $\tilde{c}>0$ and a function $z(x)=a_i^Tx+b_i$ for $x\in\mathcal{X}_i$ such that for any pair of neighboring regions $(\mathcal{X}_i,\mathcal{X}_j)$, the following holds:
$$\begin{aligned}&a_i^Tv+b_i=a_j^Tv+b_j\ \ \text{for}\ \ v\in\mathcal{V}(\mathcal{X}_i\cap\mathcal{X}_j)\\&a_i^Tu+b_i\geq a_j^Tu+b_j+\tilde{c}\ \ \text{for}\ \ u\in\mathcal{V}(\mathcal{X}_i)\backslash\mathcal{V}(\mathcal{X}_j).\end{aligned} \tag{23}$$
According to Proposition III.2, if we choose $\alpha=c/\tilde{c}>0,\beta=0$, (23) is equivalent to:
$$\begin{aligned}&(\alpha a_i)^Tv+(\alpha b_i)=(\alpha a_j)^Tv+(\alpha b_j)\ \ \text{for}\ \ v\in\mathcal{V}(\mathcal{X}_i\cap\mathcal{X}_j)\\&(\alpha a_i)^Tu+(\alpha b_i)\geq(\alpha a_j)^Tu+(\alpha b_j)+c\\&\qquad\qquad\qquad\text{for}\ \ u\in\mathcal{V}(\mathcal{X}_i)\backslash\mathcal{V}(\mathcal{X}_j).\end{aligned}$$

In other words, $(\alpha a_i, \alpha b_i)$ for all $i \in \mathcal{I}_N$ also make the constraint set (16) and (17) feasible. Therefore, the optimization problem (18) is feasible with any given constant $c > 0$. ■

Remark III.4 Note that Theorem III.3 holds true not only for polytopic partitions of polytopes but also for cell complexes of nonconvex polyhedral sets in $\mathbb{R}^d$.

Remark III.5 According to Proposition II.4, if a polyhedral partition is convexly liftable, then it should be a cell complex. Therefore, the optimization problem (18) is infeasible for the polytopic partitions of polytopes whose facet-to-facet property is not fulfilled.

To illustrate Algorithm 2, a cell complex of a polytope is presented in Fig.3. One of its convex liftings is also presented therein. Further, Fig.4 depicts a cell complex of a nonconvex set which is the underlying partition. One of its convex liftings is also illustrated above.

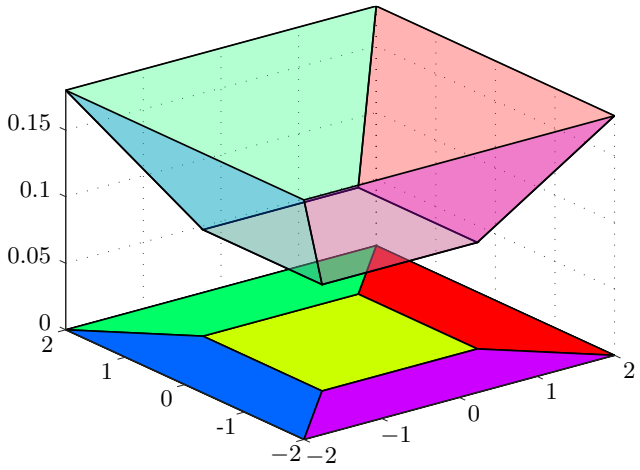


Fig. 3: A cell complex of a polytope and its convex lifting resulted from Algorithm 2.

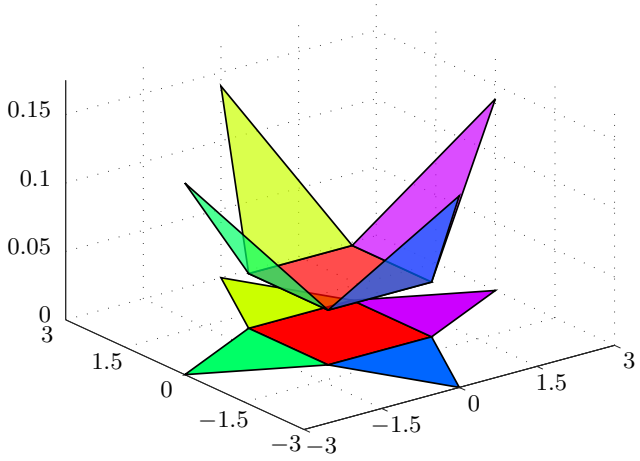


Fig. 4: A cell complex of a nonconvex set and its convex lifting resulted from Algorithm 2.

C. Construction of convex liftings based on the halfspace representation

Recall that Algorithm 2 relies on the vertex representation of related polytopes. It is worth recalling that the pivoting algorithm, by Avis and Fukuda in [5], can carry out the vertex enumeration in time $\mathcal{O}(ndv)$, where d denotes the dimension of the given polytope, v represents the number of vertices of this polytope and n denotes the number of facets of this polytope. Note however that the vertex enumeration in many cases is not necessary, particularly in control theory where the implementation of state-space partition mostly hinges on the halfspace representation. Moreover, the construction of convex liftings based on the vertex representation is limited to polytopic partitions, therefore, this construction for partitions of unbounded polyhedra may cause computational complications, see [26]. Motivated by these limitations, this subsection presents an approach to construct convex liftings based on the halfspace representation.

Given a convexly liftable cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, as denoted in Definition II.3, we use

$$z(x) = a_i^T x + b_i \quad \text{for} \quad x \in \mathcal{X}_i, \tag{24}$$

to denote a convex lifting for $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$. As a convex lifting, $z(x)$ has to fulfill the continuity and convexity conditions. More precisely, for any pair of neighboring regions $(\mathcal{X}_i, \mathcal{X}_j)$, the corresponding affine functions $a_i^T x + b_i$ and $a_j^T x + b_j$ have to satisfy:

$$a_i^T x + b_i \geq a_j^T x + b_j \quad \text{for all} \quad x \in \mathcal{X}_i, \tag{25a}$$
$$a_j^T x + b_j \geq a_i^T x + b_i \quad \text{for all} \quad x \in \mathcal{X}_j. \tag{25b}$$

It can be easily observed that according to (25a) and (25b):

$$a_i^T x + b_i = a_j^T x + b_j \quad \text{for all} \quad x \in \mathcal{X}_i \cap \mathcal{X}_j, \tag{26}$$

implies the continuity of $z(x)$ at any point $x \in \mathcal{X}_i \cap \mathcal{X}_j$. Furthermore, given the halfspace representation of region $\mathcal{X}_i$, i.e., $\mathcal{X}_i = \left\{x \in \mathbb{R}^d : H_i x \leq K_i\right\}$, (25a) holding for all $x \in \mathcal{X}_i$, leads to:

$$\begin{aligned} \mathcal{X}_i &= \left\{x \in \mathbb{R}^d : H_i x \leq K_i\right\} \\ &\subseteq P = \left\{x \in \mathbb{R}^d : (a_j - a_i)^T x \leq b_i - b_j\right\}. \end{aligned} \tag{27}$$

According to the extended Farkas lemma [33], (27) leads to the existence of a suitable vector λ_{ij} such that:

$$\lambda_{ij} \geq 0,\ \lambda_{ij} H_i = (a_j - a_i)^T,\ \lambda_{ij} K_i \leq b_i - b_j. \tag{28}$$

Similarly, given the halfspace representation of region $\mathcal{X}_j$, i.e., $\mathcal{X}_j = \left\{x \in \mathbb{R}^d : H_j x \leq K_j\right\}$, (25b) leads to the existence of a suitable vector λ_{ji} such that

$$\lambda_{ji} \geq 0,\ \lambda_{ji} H_j = (a_i - a_j)^T,\ \lambda_{ji} K_j \leq b_j - b_i. \tag{29}$$

It is worth emphasizing that the constraints in (25) cannot guarantee that the affine functions corresponding to regions $\mathcal{X}_i$ and $\mathcal{X}_j$ are distinct, i.e., $(a_i, b_i) \neq (a_j, b_j)$. Therefore, in order to ensure this property of a convex lifting, one needs to impose additional constraints. A simple way to avoid nonlinear constraints is to require:

$$a_i^T x_0 + b_i \geq a_j^T x_0 + b_j + c, \tag{30}$$

for a given scalar constant $c > 0$ and $x_0 \in \text{int}(\mathcal{X}_i)$. Constraint (30) is meaningful to guarantee $(a_i, b_i) \neq (a_j, b_j)$. In fact, if the converse situation happens, constraint (30) will be infeasible. Also, x_0 can be freely chosen as long as it lies in the interior of $\mathcal{X}_i$; the Chebyshev center is also a possible candidate. Recall that Chebyshev center of a polyhedron $\mathcal{X}$ is the center of the largest inscribed ball of $\mathcal{X}$. More precisely, finding Chebyshev center x_c of polyhedron $\mathcal{X}$ amounts to solving the following problem

$$\max_{x_c, r} r \quad \text{s.t.}$$
$$x_c \in \mathcal{X},\ \left\{x \in \mathbb{R}^d : (x - x_c)^T (x - x_c) \leq r\right\} \subseteq \mathcal{X}.$$

Note also that this problem can be easily transformed into a linear programming problem, see [11].

For completeness, a procedure to construct convex liftings based on the halfspace representation for a given convexly liftable cell complex is summarized in Algorithm 3.

Algorithm 3 Construction of a convex lifting for a given convexly liftable cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$

Input: $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, the halfspace representation of $\mathcal{X}_i = \left\{x \in \mathbb{R}^d : H_i x \leq K_i\right\}$ and a scalar constant $c > 0$.

Output: gains a_i, b_i.

1: Find Chebyshev center for each region $\mathcal{X}_i$, denoted by x_i.

2: Register all pairs of neighboring regions in $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$.

3: For each pair of neighboring regions $(\mathcal{X}_i, \mathcal{X}_j)$, $(i, j) \in \mathcal{I}_N^2$:

- add constraints:

$$\lambda_{ij} \geq 0,\ \lambda_{ij} H_i = (a_j - a_i)^T,\ \lambda_{ij} K_i \leq b_i - b_j; \quad (31)$$

- add constraints:

$$\lambda_{ji} \geq 0,\ \lambda_{ji} H_j = (a_i - a_j)^T,\ \lambda_{ji} K_j \leq b_j - b_i; \quad (32)$$

- add constraint:

$$a_i^T x_i + b_i \geq a_j^T x_i + b_j + c. \quad (33)$$

4: Solve the following convex optimization problem by minimizing a chosen cost function, e.g.,

$$\min_{a_i, b_i, \lambda_{ij}, \lambda_{ji}} \sum_{i=1}^{N} (a_i^T a_i + b_i^T b_i) \quad \text{subject to} \quad (31), (32), (33). \quad (34)$$

Remark III.6 Note that Chebyshev center of a polyhedron may not always be unique or may lie at infinity. As emphasized above, other candidate of this point is possible as long as it lies in the interior of $\mathcal{X}_i$.

The following results present important formal properties of the construction resulting from Algorithm 3.

Proposition III.7 *If the optimization problem* (34) *is feasible, then the function $z(x) = a_i^T x + b_i$ for $x \in \mathcal{X}_i$ represents a convex lifting for cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$.*

Proof: If the optimization problem (34) is feasible, then the constraints (31), (32) and (33) are all feasible. According to the extended Farkas lemma [33], constraint (31) leads to:

$$a_i^T x + b_i \geq a_j^T x + b_j \quad \text{for all} \quad x \in \mathcal{X}_i. \quad (35)$$

Similarly, it follows from constraint (32) that

$$a_j^T x + b_j \geq a_i^T x + b_i \quad \text{for all} \quad x \in \mathcal{X}_j. \quad (36)$$

According to (35) and (36), the continuity of $z(x)$ at the common boundary of $\mathcal{X}_i$ and $\mathcal{X}_j$ is verified by

$$a_i^T x + b_i = a_j^T x + b_j \quad \text{for all} \quad x \in \mathcal{X}_i \cap \mathcal{X}_j.$$

This leads to the following inclusions for the vertices and the extreme rays of $\mathcal{X}_i \cap \mathcal{X}_j$:

$$\begin{aligned} a_i^T v + b_i = a_j^T v + b_j \quad &\text{for all} \quad v \in \mathcal{V}(\mathcal{X}_i \cap \mathcal{X}_j), \\ a_i^T r = a_j^T r \quad &\text{for all} \quad r \in \mathcal{R}(\mathcal{X}_i \cap \mathcal{X}_j). \end{aligned} \quad (37)$$

Moreover, constraint (33) implies

$$a_i^T x_i + b_i \geq a_j^T x_i + b_j + c > a_j^T x_i + b_j. \quad (38)$$

From (37) and (38), any point x, described in the following form:

$$x = \gamma x_i + \sum_{v \in \mathcal{V}(\mathcal{X}_i \cap \mathcal{X}_j)} \alpha(v) v + \sum_{r \in \mathcal{R}(\mathcal{X}_i \cap \mathcal{X}_j)} \mu(r) r$$

with $\alpha(v), \mu(r) \in \mathbb{R}, \gamma + \sum_{v \in \mathcal{V}(\mathcal{X}_i \cap \mathcal{X}_j)} \alpha(v) = 1$ satisfies:

$$a_i^T x + b_i > a_j^T x + b_j \quad \text{for all} \quad \gamma > 0, \quad (39a)$$
$$a_i^T x + b_i = a_j^T x + b_j \quad \text{for} \quad \gamma = 0. \quad (39b)$$

In other words, any point x, in the halfspace containing $\mathcal{X}_i$ but not in $\text{aff}(\mathcal{X}_i \cap \mathcal{X}_j)$, satisfies (39a). Otherwise, any point $x \in \text{aff}(\mathcal{X}_i \cap \mathcal{X}_j)$, satisfies (39b).

The same inclusion holds for the other pairs of neighboring regions, leading to the fact that:

$$a_i^T x + b_i > a_j^T x + b_j \quad \text{for all} \quad x \in \mathcal{X}_i \backslash \mathcal{X}_j \quad \text{and} \quad j \neq i. \quad (40)$$

Therefore, function $z(x) = a_i^T x + b_i$ for $x \in \mathcal{X}_i$ represents a convex lifting for $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ according to Definition II.3. ■

Similar to Subsection III-B, any value of the given scalar c in Algorithm 3 does not affect the feasibility of the optimization problem (34) as long as $c > 0$.

Theorem III.8 *The given cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, is convexly liftable if and only if the optimization problem* (34) *is feasible for any constant $c > 0$.*

Proof: $\longleftarrow$ This inclusion directly follows according to Proposition III.7.

$\longrightarrow$ If the given cell complex $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ is convexly liftable, then there exists a function $z(x) = a_i^T x + b_i$ for $x \in \mathcal{X}_i$ such that for any pair of neighboring regions $(\mathcal{X}_i, \mathcal{X}_j)$, the following holds:

$$\begin{aligned} a_i^T x + b_i \geq a_j^T x + b_j \quad &\text{for} \quad x \in \mathcal{X}_i \\ a_j^T x + b_j \geq a_i^T x + b_i \quad &\text{for} \quad x \in \mathcal{X}_j. \end{aligned} \quad (41)$$

According to the extended Farkas lemma, inclusion (41) leads to the existence of two suitable vectors $\lambda_{ij}, \lambda_{ji}$ such that:

$$\lambda_{ij} \geq 0,\ \lambda_{ij}H_i + (a_i - a_j)^T = 0,\ \lambda_{ij}K_i \leq b_i - b_j, \quad (42a)$$

$$\lambda_{ji} \geq 0,\ \lambda_{ji}H_j + (a_j - a_i)^T = 0,\ \lambda_{ji}K_j \leq b_j - b_i, \quad (42b)$$

where $\mathcal{X}_i = \left\{x \in \mathbb{R}^d : H_i x \leq K_i\right\}$, $\mathcal{X}_j = \left\{x \in \mathbb{R}^d : H_j x \leq K_j\right\}$.

Also, since $z(x)$ is a convex lifting for $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, there exists a constant $c_{ij} > 0$ for each pair of neighboring regions $(\mathcal{X}_i, \mathcal{X}_j)$ such that

$$a_i^T x_i + b_i \geq a_j^T x_i + b_j + c_{ij}, \quad (43)$$

where x_i represents Chebyshev center of $\mathcal{X}_i$. Let $\tilde{c}$ be the minimal value of c_{ij} for the pairs of neighboring regions $(\mathcal{X}_i, \mathcal{X}_j)$, i.e.,

$$\tilde{c} = \min_{(i,j)\in\mathcal{I}_N^2 \,|\, \dim(\mathcal{X}_i \cap \mathcal{X}_j) = d-1} c_{ij}.$$

Accordingly, for any pair of neighboring regions $(\mathcal{X}_i, \mathcal{X}_j)$, we obtain:

$$a_i^T x_i + b_i \geq a_j^T x_i + b_j + \tilde{c}.$$

Choose $\delta = c/\tilde{c} > 0$, it follows that

$$(\delta a_i)^T x_i + (\delta b_i) \geq (\delta a_j)^T x_i + (\delta b_j) + c. \quad (44)$$

According to (42a), (42b) and (44), it can be deduced that $(\delta a_i, \delta b_i)$ for all $i \in \mathcal{I}_N$ make the constraints (31), (32) and (33) feasible, since $\delta\lambda_{ij}, \delta\lambda_{ji} \geq 0$. In other words, the optimization problem (34) is feasible with any given constant $c > 0$. ∎

Remark III.9 Note that according to Theorem III.8, the feasibility of the optimization problem (34) serves as another necessary and sufficient condition for the convex liftability of the polyhedral partitions of polyhedra.

Remark III.10 As proved in Proposition II.4, a polyhedral partition admitting a convex lifting should be a cell complex. Accordingly, for any polyhedral partitions of polyhedra whose facet-to-facet property is not fulfilled, the optimization problem (34) is infeasible.

D. Convexly non-liftable partitions

This subsection addresses polyhedral partitions whose convex liftability is not fulfilled. This is usually the case in control theory, in particular for polyhedral partitions obtained from linear model predictive control problems with respect to quadratic cost functions. It is worth emphasizing that rearranging a given polyhedral partition is allowed. However, any modification of the initial boundaries of the given polyhedral partition is not permitted due to the fact that it destroys the original structure of PWA controller. This may lead to the case where two different affine control laws are defined over the same region of state space. Therefore, the problem is formulated as follows: by preserving the internal boundaries, is it possible to refine a given polyhedral partition in order to recover the convex liftability property?

It will be proven that there exists at least one subdivision which can retrieve the convex liftability for a given polyhedral partition. The proof will show that the so-called *hyperplane arrangement* technique, defined as the decomposition of a space by a set of hyperplanes, can carry out this subdivision.

Theorem III.11 *Given a convexly non-liftable polyhedral partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polyhedron $\mathcal{X} \subseteq \mathbb{R}^d$, there exists at least one subdivision, preserving the internal boundaries of this partition, such that the new cell complex is convexly liftable.*

Proof: Let $\mathcal{H}(\mathcal{X}_i)$ be the set of supporting hyperplanes of $\mathcal{X}_i$ at its facets; also define $\mathcal{H}(\mathcal{X}) = \bigcup_{i\in\mathcal{I}_N} \mathcal{H}(\mathcal{X}_i)$. We will show that the decomposition of $\mathcal{X}$ by $\mathcal{H}(\mathcal{X})$ leads to a new cell complex which is convexly liftable. As presented above, such a decomposition is denoted as *hyperplane arrangement*. The convex liftability of such a decomposition can be proven by returning to the concept of stresses. (Details about stresses can be found in [24]).

In fact, considering any $(d-2)-$face F_0 lying in the interior of $\mathcal{X}$, this $(d-2)-$face F_0 is the intersection of finitely many hyperplanes in $\mathcal{H}(\mathcal{X})$. If $\mathcal{F}^{(d-1)}(F_0)$ denotes the set of all $(d-1)-$faces in the star of F_0, then for each $F_i^{(d-1)} \in \mathcal{F}^{(d-1)}(F_0)$, there exists a unique $F_j^{(d-1)} \neq F_i^{(d-1)}$ and $F_j^{(d-1)} \in \mathcal{F}^{(d-1)}(F_0)$ such that $F_i^{(d-1)}, F_j^{(d-1)}$ lie in a common hyperplane of $\mathcal{H}(\mathcal{X})$ and they have a common facet F_0. Accordingly, it can be seen that the inward unit normal vectors to the faces $F_i^{(d-1)}, F_j^{(d-1)}$ at their common facet F_0, denoted by $n(F_0, F_i^{(d-1)})$, $n(F_0, F_j^{(d-1)})$, respectively, satisfy:

$$n(F_0, F_i^{(d-1)}) = -n(F_0, F_j^{(d-1)}).$$

Thus, a pair of coefficients of strictly positive stresses $s(F_i^{(d-1)}), s(F_j^{(d-1)})$ exists (e.g. $s(F_i^{(d-1)}) = s(F_j^{(d-1)}) = 1$) such that:

$$s(F_i^{(d-1)})n(F_0, F_i^{(d-1)}) + s(F_j^{(d-1)})n(F_0, F_j^{(d-1)}) = 0.$$

Applying the same argument for all elements of $\mathcal{F}^{(d-1)}(F_0)$, one can obtain a strictly positive $d-$stress such that F_0 is in equilibrium. ∎

Remark III.12 Note that Theorem III.11 states the existence of a suitable refinement, whereas the proof points to a specific technique for the refinement. In a broader perspective, for a given polyhedral partition which does not admit a convex lifting, there exist multiple practical solutions for suitable refinements into a convexly liftable cell complex, hyperplane arrangement is only one of them. An alternative, *fitting* planar cell complexes into Voronoi diagrams, can be found in [1].

Return to the hyperplane arrangement technique, an algorithm to carry out this decomposition is presented in Algorithm 4 for a given polyhedral partition.

To illustrate Algorithm 4, consider the cell complex in Fig.5, the result is depicted in Fig.6. Again, the convex liftability of this cell complex can be verified by the feasibility of problem (18) or (34).

Algorithm 4 An algorithm to carry out the hyperplane arrangement technique for a given polyhedral partition.

Input: Convexly non-liftable partition $\Omega = \{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ in $\mathbb{R}^d$.

Output: Convexly liftable cell complex $\widetilde{\Omega} = \left\{\widetilde{\mathcal{X}}_i\right\}_{i\in\mathcal{I}_{\widetilde{N}}}$.

$M = [\,]$
For $i = 1 : N$
  $\mathcal{X}_i = \{x : H_i x \leq K_i\}$, $M = [M; H_i\ K_i]$
End
Remove redundant rows of matrix M.
For $i = 1 : \text{size}(M, 1)$
  $\widetilde{\Omega} = \emptyset$
  For $j = 1 : \text{length}(\Omega)$
    $\mathcal{X}_j = \{x : H_j x \leq K_j\}$
    $\mathcal{Y}^{(1)} = \left\{x : \begin{bmatrix} H_j \\ M(i, 1:d)\end{bmatrix} x \leq \begin{bmatrix} K_j \\ M(i, d+1)\end{bmatrix}\right\}$
    $\mathcal{Y}^{(2)} = \left\{x : \begin{bmatrix} H_j \\ -M(i, 1:d)\end{bmatrix} x \leq \begin{bmatrix} K_j \\ -M(i, d+1)\end{bmatrix}\right\}$
    If $\dim(\mathcal{Y}^{(1)}) = d$ & $\dim(\mathcal{Y}^{(2)}) < d$ **then**

$$\widetilde{\Omega} \leftarrow \widetilde{\Omega} \cup \left\{\mathcal{Y}^{(1)}\right\}$$

    Elseif $\dim(\mathcal{Y}^{(1)}) < d$ & $\dim(\mathcal{Y}^{(2)}) = d$ **then**

$$\widetilde{\Omega} \leftarrow \widetilde{\Omega} \cup \left\{\mathcal{Y}^{(2)}\right\}$$

    Elseif $\dim(\mathcal{Y}^{(1)}) = d$ & $\dim(\mathcal{Y}^{(2)}) = d$ **then**

$$\widetilde{\Omega} \leftarrow \widetilde{\Omega} \cup \left\{\mathcal{Y}^{(1)}, \mathcal{Y}^{(2)}\right\}$$

    End
  End
  $\Omega \leftarrow \widetilde{\Omega}$
End

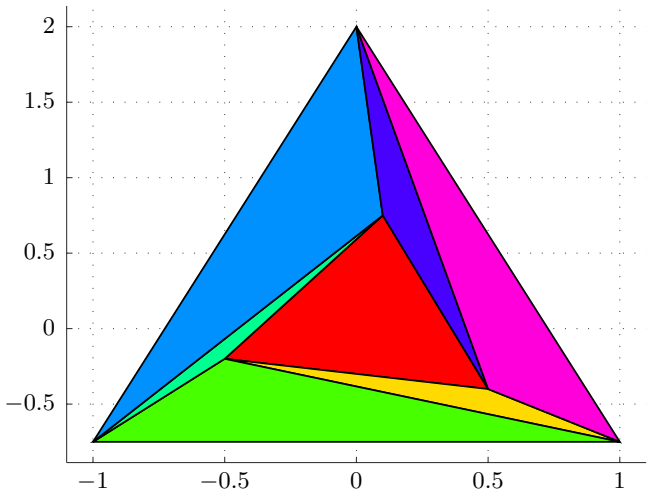


Fig. 5: A convexly non-liftable cell complex in $\mathbb{R}^2$.

IV. Applications of Convex Liftings in Control

This section aims to apply convex liftings to facilitate the implementation of PWA control laws. Note that earlier studies in this subject can be found in [17], [24]. In control theory, since the performance of physical systems is always limited,

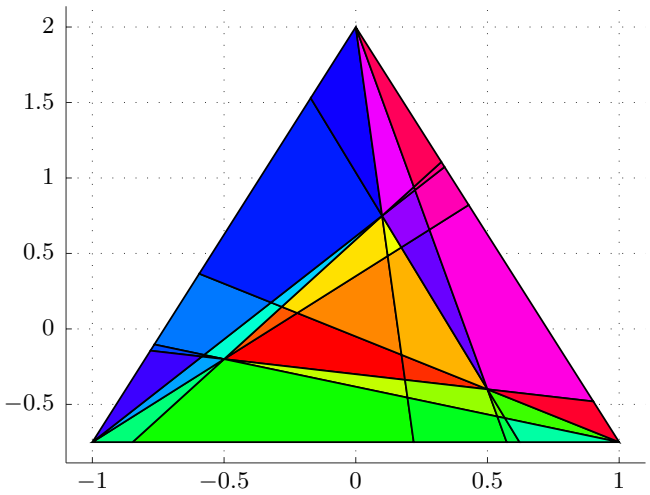


Fig. 6: Cell complex resulted from Algorithm 4.

the control signal is usually bounded. Therefore, without loss of generality, the constraints on current control variable denoted by $u \in \mathbb{R}^{d_u}$, is assumed to be in the following form:

$$u_{\min} \leq u \leq u_{\max}.$$

For ease of presentation, we use $u^{(i)}$ to denote the i^{th} component of vector u. By an *unsaturated region*, we denote a region whose associated control law is not of componentwise saturation, i.e., $u^{(i)}_{\min} < u^{(i)} < u^{(i)}_{\max}$ for at least one $i \in \mathcal{I}_{d_u}$. Furthermore, a *saturated region* implies a region corresponding to a componentwise saturated control law, i.e., either $u^{(i)} = u^{(i)}_{\min}$ or $u^{(i)} = u^{(i)}_{\max}$ for all $i \in \mathcal{I}_{d_u}$. Accordingly, given a state-space partition, the *unsaturated partition* consists of the unsaturated regions. Such a partition may not be a partition of a polyhedron but of a nonconvex set. The developments of this section are motivated by two following observations:

- the complexity of state-space partitions is mainly due to the saturation, thus the boundaries between saturated regions of the same controller can be appropriately modified;
- in most practical MPC problems, the unsaturated partition is convexly liftable.

Note that the existence of unsaturated partition is the premise of works on complexity reduction in explicit MPC controllers in [20], [21].

For ease of presentation, the following assumptions are necessary for the next developments.

Assumption IV.1 The control input is a scalar variable, i.e., $\dim(u) = d_u = 1$.

Assumption IV.2 The unsaturated partition is a convexly liftable polytopic partition.

Assumption IV.3 The state space $\mathcal{X}$ is a polytope.

Assumption IV.4 The given PWA control law is continuous.

Assumption IV.1 is not restrictive, since the development presented in the sequel can be easily extended to multivariable

cases. Note also that even if the unsaturated partition is not convexly liftable, one can use Algorithm 4 to split it into a convexly liftable cell complex. This is meaningful to avoid a complete hyperplane rearrangement of the initial state-space partition. Therefore, Assumption IV.2 loses no generality of the proposed schemes. Also, Assumption IV.3 restricts our attention to polytopic partitions of the state space. This is not restrictive, since the construction can be easily extended to polyhedral partitions. Finally, we are exclusively interested in implementation of the continuous PWA controllers as presented in Assumption IV.4.

Given a PWA controller

$$u(x) = H_i x + G_i \quad \text{for} \quad x \in \mathcal{X}_i, \tag{45}$$

defined over a polytopic partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ of a polytope $\mathcal{X} \subset \mathbb{R}^d$ satisfying Assumptions IV.1, IV.2 and IV.4, let $\mathcal{I}^{\text{uns}} \subset \mathcal{I}_N$ denote the index set such that $\{\mathcal{X}_i\}_{i\in\mathcal{I}^{\text{uns}}}$ represents the unsaturated partition of $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ and $u(x)$. Also, we use $\ell^{\text{uns}}(x)$ to denote a convex lifting for $\{\mathcal{X}_i\}_{i\in\mathcal{I}^{\text{uns}}}$, i.e.,

$$\ell^{\text{uns}}(x) = (a_i^{\text{uns}})^T x + b_i^{\text{uns}} \quad \text{for} \quad x \in \mathcal{X}_i,\, i \in \mathcal{I}^{\text{uns}}. \tag{46}$$

Now, in order to use Algorithm 1, we need to construct a convex lifting, denoted by $\ell(x)$, which is defined over the whole $\mathcal{X}$ and coincides with $\ell^{\text{uns}}(x)$ over $\{\mathcal{X}_i\}_{i\in\mathcal{I}^{\text{uns}}}$. To perform this task, two different constructions will be presented in the sequel; the first one aims to rearrange the saturated regions to reduce the number of regions and in the meantime find a suitable convex lifting over the rearranged paritition, whereas the second one incorporates suitable clippings of control signal with convex liftings.

A. Construction based on convex liftings for the vertices of the state space $\mathcal{X}$

The first construction aims to compute an appropriate height h corresponding to the vertices of $\mathcal{X}$. This height has to satisfy that the augmented vertices $\left[v^T\ \ell^{\text{uns}}(v)\right]^T$ for $v \in \bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)$ and $\left[v^T\ h\right]^T$ for $v \in \mathcal{V}(\mathcal{X})\backslash\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)$ create a convex lifting $\ell(x)$ over $\mathcal{X}$, as mentioned previously. This construction is presented in Algorithm 5.

The following lemma represents the most important property of $\ell(x)$ resulted from Algorithm 5.

Lemma IV.5 *$\ell(x)$ resulted from Algorithm 5 satisfies:*

$$\ell(x) = \ell^{\text{uns}}(x) \text{ for all } x \in \bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{X}_i.$$

Proof: Consider any point $\left[x^T\ z\right]^T \in \Pi$, defined in (48). This point can be described as a convex combination of the points in P_1, P_2 as follows:

$$\begin{aligned}
&\alpha(v), \beta(v) \geq 0,\\
&\sum_{v\in\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \alpha(v) + \sum_{v\in\mathcal{V}(\mathcal{X})\backslash\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \beta(v) = 1,\\
&\begin{bmatrix} x \\ z \end{bmatrix} = \sum_{v\in\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \alpha(v) \begin{bmatrix} v \\ \ell^{\text{uns}}(v) \end{bmatrix}\\
&\qquad\qquad + \sum_{v\in\mathcal{V}(\mathcal{X})\backslash\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \beta(v) \begin{bmatrix} v \\ h \end{bmatrix}.
\end{aligned}$$

Denote also $\underline{z}(x) = \max_{j\in\mathcal{I}^{\text{uns}}} (a_j^{\text{uns}})^T x + b_j^{\text{uns}}$ for $x \in \mathcal{X}$. Clearly, $\underline{z}(x) = \ell^{\text{uns}}(x)$ for $x \in \bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{X}_i$ and is known to be a convex function over $\mathcal{X}$. According to (47), it follows that:

$$\begin{aligned}
&\sum_{v\in\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \alpha(v)\ell^{\text{uns}}(v) + \sum_{v\in\mathcal{V}(\mathcal{X})\backslash\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \beta(v)h\\
&\geq \sum_{v\in\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \alpha(v)\underline{z}(v)\\
&\qquad + \sum_{v\in\mathcal{V}(\mathcal{X})\backslash\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \beta(v)(c + \underline{z}(v))\\
&\geq \underline{z}(x) + \sum_{v\in\mathcal{V}(\mathcal{X})\backslash\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)} \beta(v)c\\
&\geq \underline{z}(x).
\end{aligned}$$

If $x \in \mathcal{X}_i$ for $i \in \mathcal{I}^{\text{uns}}$, then the equality only happens when $\beta(v) = 0$ for $v \in \mathcal{V}(\mathcal{X})\backslash\bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i)$. In other words, when

---

Algorithm 5 Construct a convex lifting over $\mathcal{X} \subset \mathbb{R}^d$, coincident with $\ell^{\text{uns}}(x)$ over the unsaturated partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}^{\text{uns}}}$.

Input: $\{\mathcal{X}_i\}_{i\in\mathcal{I}^{\text{uns}}}$, $\ell^{\text{uns}}(x)$ defined in (46), $\mathcal{X}$ and a given constant $c > 0$.

Output: h, $\ell(x)$.

1: Solve the problem:

$$\min_h h \quad \text{s.t.} \quad h \geq c + (a_i^{\text{uns}})^T v + b_i^{\text{uns}},\ \forall i \in \mathcal{I}^{\text{uns}},\\ \text{for all} \quad v \in \mathcal{V}(\mathcal{X}), v \notin \bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i) \tag{47}$$

2: Construct the polytope

$$\begin{aligned}
P_1 &= \left\{ \begin{bmatrix} v \\ \ell^{\text{uns}}(v) \end{bmatrix} : v \in \bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i) \right\} \subset \mathbb{R}^{d+1},\\
P_2 &= \left\{ \begin{bmatrix} v \\ h \end{bmatrix} : v \in \mathcal{V}(\mathcal{X})\backslash \bigcup_{i\in\mathcal{I}^{\text{uns}}} \mathcal{V}(\mathcal{X}_i) \right\} \subset \mathbb{R}^{d+1},\\
\Pi &= \text{conv}\,\{P_1 \cup P_2\}
\end{aligned} \tag{48}$$

3: Solve the following parametric linear programming problem:

$$z^*(x) = \arg\min_z z \text{ subject to } \left[x^T\ z\right]^T \in \Pi. \tag{49}$$

4: $\ell(x) = z^*(x)$.

$x \in \bigcup_{i\in\mathcal{I}^{\rm uns}} \mathcal{X}_i$ the minimal cost function of (49) satisfies $\ell(x) = z^*(x) = \underline{z}(x) = \ell^{\rm uns}(x)$. ■

Let $\{\mathcal{Y}_j\}_{j\in\mathcal{I}_M}$ denote the state-space partition associated with $\ell(x)$ obtained from Algorithm 5. The following corollary presents another property of $\ell(x)$ resulted from Algorithm 5.

Corollary IV.6 *$\ell(x)$ resulted from Algorithm 5, represents a convex lifting for the polytopic partition $\{\mathcal{Y}_j\}_{j\in\mathcal{I}_M}$ of the state space $\mathcal{X}$.*

Proof: The proof follows as a direct consequence of Lemma II.8. ■

According to Lemma IV.5, for each region $\mathcal{X}_i$ of the unsaturated partition $\{\mathcal{X}_i\}_{i\in\mathcal{I}^{\rm uns}}$, there exists a region $\mathcal{Y}_j$ of $\{\mathcal{Y}_j\}_{j\in\mathcal{I}_M}$ such that $\mathcal{Y}_j = \mathcal{X}_i$. If $\mathcal{I}^{\max} \subset \mathcal{I}_N$ ($\mathcal{I}^{\min} \subset \mathcal{I}_N$) denotes the set of indices such that each region $\mathcal{X}_j$, $j \in \mathcal{I}^{\max}$ ($j \in \mathcal{I}^{\min}$) is associated with saturated controller $u(x) = u_{\max}$ ($u(x) = u_{\min}$) for $x \in \mathcal{X}_j$. Accordingly, $\mathcal{I}_N = \mathcal{I}^{\rm uns} \cup \mathcal{I}^{\max} \cup \mathcal{I}^{\min}$. Define the following controller, denoted by $\widetilde{f}_{\rm pwa}(x)$, associated with $\{\mathcal{Y}_j\}_{j\in\mathcal{I}_M}$:

$$\widetilde{f}_{\rm pwa}(x) = \begin{cases} u(x) \text{ if } x \in \mathcal{Y}_j \text{ s.t. } \exists i \in \mathcal{I}^{\rm uns}, \mathcal{Y}_j = \mathcal{X}_i \\ u_{\max} \text{ if } x \in \mathcal{Y}_j \text{ s.t. } \mathcal{Y}_j \subset \bigcup\limits_{i\in\mathcal{I}^{\max}} \mathcal{X}_i \\ u_{\min} \text{ if } x \in \mathcal{Y}_j \text{ s.t. } \mathcal{Y}_j \subset \bigcup\limits_{i\in\mathcal{I}^{\min}} \mathcal{X}_i \end{cases}$$

Note that the new PWA control law $\widetilde{f}_{\rm pwa}(x)$ is equivalent to the given one $u(x)$ in the sense that $\widetilde{f}_{\rm pwa}(x) = u(x)$ for all $x \in \mathcal{X}$. Therefore, it suffices to implement $\widetilde{f}_{\rm pwa}(x)$ as in Algorithm 1.

Remark IV.7 In case the given PWA control law $u(x)$ is of multiple inputs, then implementation of this controller according to the construction of convex liftings as in Algorithm 5 should be carried out componentwise. Roughly speaking, the implementation of $u(x)$ can be wrapped up as follows:

- construct a convex lifting $(\ell^{\rm uns})^{(i)}(x)$ for the unsaturated partition, denoted by $\{\mathcal{X}_j\}_{j\in\mathcal{I}^{(i)}}$, of the state-space partition $\{\mathcal{X}_j\}_{j\in\mathcal{I}_N}$ ($\mathcal{I}^{(i)} \subseteq \mathcal{I}_N$) associated with the $i^{\rm th}$ component $u^{(i)}(x)$ of the given PWA controller $u(x)$;
- construct an extended convex lifting, denoted by $\ell^{(i)}(x)$ defined over $\mathcal{X}$ for $(\ell^{\rm uns})^{(i)}(x)$ as in Algorithm 5;
- rearrange each component $u^{(i)}(x)$ of the given PWA controller $u(x)$ according to $\ell^{(i)}(x)$; denote this rearranged component by $\tilde{u}^{(i)}(x)$
- implement each rearranged component $\tilde{u}^{(i)}(x)$ as in Algorithm 1.

Note also that in this multiple-input case, the unsaturated partitions $\{\mathcal{X}_j\}_{j\in\mathcal{I}^{(i)}}$ of $\{\mathcal{X}_j\}_{j\in\mathcal{I}_N}$ associated with the components $u^{(i)}(x)$ of $u(x)$, may not be identical.

B. Construction based on convex lifting and clipping

Although the construction of $\ell(x)$ in Algorithm 5 shows an advancement in terms of efficient storage, the number of affine functions composing $\ell(x)$ is still relatively large. We will present a more efficient construction which can considerably reduce the number of affine functions. This construction employs convex liftings and the concept of clipping presented in [20].

As mentioned in the proof of Lemma IV.5, we can choose such a convex lifting $\ell(x)$ as follows:

$$\ell(x) = \max_{j\in\mathcal{I}^{\rm uns}} (a_j^{\rm uns})^T x + b_j^{\rm uns} \quad \text{for} \quad x \in \mathcal{X}. \tag{50}$$

Obviously, this construction ensures that $\ell(x) = \ell^{\rm uns}(x)$ for all $x \in \bigcup_{i\in\mathcal{I}^{\rm uns}} \mathcal{X}_i$. Let $\{\mathcal{Y}_i\}_{i\in\mathcal{I}_M}$ denote the polytopic partition of $\mathcal{X}$ associated with the convex lifting $\ell(x)$ defined in (50). For ease of presentation, denote $\ell(x)$ as follows:

$$\ell(x) = \widetilde{a}_i^T x + \widetilde{b}_i \quad \text{for} \quad x \in \mathcal{Y}_i. \tag{51}$$

According to the new state-space partition $\{\mathcal{Y}_i\}_{i\in\mathcal{I}_M}$, a (non-equivalent) rearrangement of the given control law $u(x)$, denoted by $\widetilde{f}_{\rm pwa}(x)$, is put forward as follows:

$$\begin{aligned} \widetilde{f}_{\rm pwa}(x) = \widetilde{H}_i x + \widetilde{G}_i = H_j x + G_j \quad &\text{for} \quad x \in \mathcal{Y}_i \\ &\text{such that} \quad j \in \mathcal{I}^{\rm uns},\ \mathcal{X}_j \subseteq \mathcal{Y}_i. \end{aligned} \tag{52}$$

The following corollary represents a property of $\widetilde{f}_{\rm pwa}(x)$ constructed in (52).

Corollary IV.8 *If $u(x)$ defined in* (45) *is continuous, then $\widetilde{f}_{\rm pwa}(x)$ is continuous over $\bigcup_{i\in\mathcal{I}^{\rm uns}} \mathcal{X}_i$.*

Proof: It can be observed that $\widetilde{f}_{\rm pwa}(x) = u(x)$ over $\bigcup_{i\in\mathcal{I}^{\rm uns}} \mathcal{X}_i$. Obviously, the proof directly follows. ■

Note also that the continuity of $\widetilde{f}_{\rm pwa}(x)$ may not be guaranteed over $\mathcal{X}\backslash\bigcup_{i\in\mathcal{I}^{\rm uns}} \mathcal{X}_i$, as this property is not accounted for in the definition of $\widetilde{f}_{\rm pwa}(x)$ in (52). However, in implementation, $\widetilde{f}_{\rm pwa}(x)$ will be saturated over this region such that the given constraints are respected. The implementation is summarized in Algorithm 6.

Algorithm 6 Efficient implementation of PWA controllers based on convex liftings and clippings

1: Store $\ell(x)$ defined in (50), denoted as in (51) and the PWA controller $\widetilde{f}_{\rm pwa}(x)$ defined as in (52).
2: At each sampling time, obtain the current state x.
3: Find index $i \in \mathcal{I}_M$ such that:

$$\widetilde{a}_i^T x + \widetilde{b}_i = \max_{j\in\mathcal{I}_M} (\widetilde{a}_j^T x + \widetilde{b}_j).$$

4: Evaluate control law

$$u^*(x) = \begin{cases} \widetilde{H}_i x + \widetilde{G}_i & \text{if} \quad u_{\min} \le \widetilde{H}_i x + \widetilde{G}_i \le u_{\max} \\ u_{\max} & \text{if} \quad \widetilde{H}_i x + \widetilde{G}_i > u_{\max} \\ u_{\min} & \text{if} \quad \widetilde{H}_i x + \widetilde{G}_i < u_{\min}. \end{cases}$$

5: Return to step 2.

Note that by the saturation at Step 4, $u^*(x)$ is equivalent to the given PWA controller, i.e., $u^*(x) = u(x)$, see [17], [20].

Remark IV.9 Algorithm 6 can be easily extended to the multiple-input case. More precisely, unlike the implementation in Subsection IV-A, this implementation, based on convex

liftings and clippings, only requires the construction of a single convex lifting for the unsaturated partition of $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$ associated with $u(x)$ instead of a convex lifting for each component of $u(x)$. Accordingly, the given multiple-input $u(x)$ can be implemented as in Algorithm 6 where Step 4 has to be modified to carry out the componentwise saturation. Namely, if d_u denotes the dimension of $u(x)$, then

$$u^*(x) = \left[(u^*)^{(1)}(x)\dots(u^*)^{(d_u)}(x)\right]^T,$$

where $(u^*)^{(j)}(x)$ for $j \in \mathcal{I}_{d_u}$ are defined as follows:

$$(u^*)^{(j)}(x) = \begin{cases} \widetilde{H}_i^{(j)}x + \widetilde{G}_i^{(j)} \text{ if } u_{\min}^{(j)} \le \widetilde{H}_i^{(j)}x + \widetilde{G}_i^{(j)} \le u_{\max}^{(j)} \\ u_{\max}^{(j)} \text{ if } \widetilde{H}_i^{(j)}x + \widetilde{G}_i^{(j)} > u_{\max}^{(j)} \\ u_{\min}^{(j)} \text{ if } \widetilde{H}_i^{(j)}x + \widetilde{G}_i^{(j)} < u_{\min}^{(j)}, \end{cases}$$

and $\widetilde{H}_i^{(j)}, \widetilde{G}_i^{(j)}, u_{\max}^{(j)}, u_{\min}^{(j)}$ again denote the j^{th} row of matrices $\widetilde{H}_i, \widetilde{G}_i, u_{\max}, u_{\min}$.

C. Complexity analysis

In the following, we assess memory and runtime complexity of the proposed approaches in the context of implementation of PWA controllers. As a reference, the total memory consumed by the original PWA control law together with its underlying partition, $\{\mathcal{X}_i\}_{i\in\mathcal{I}_N}$, is $\sum_{i=1}^{N}(c_i+d_u)(d+1)$ real numbers, where c_i denotes the number of halfspaces defining the i-th region. On the other hand, storing the simpler controller obtained via Section IV-A and its associated control law requires $M(1+d_u)(d+1)$ real numbers, where M is the number of affine terms of the convex lifting, resulted from Algorithm 5. Finally, the memory footprint of the implementation proposed in Section IV-B amounts to $|\mathcal{I}^{\text{uns}}|(1+d_u)(d+1)+2d_u$ real numbers, where the second term denotes a negligible memory needed to encode the clipping function in Algorithm 6.

In addition, we quantify the necessary on-line computational effort. Specifically, the standard implementation of the original PWA control laws consists of the point location problem, i.e., finding index i of the region $\mathcal{X}_i$ that contains x, and evaluation of the corresponding control law. In the worst case, this amounts to $\sum_{i=1}^{N} c_i(2d+1) + 2d_u d$ floating point operations (FLOPs). Note that the proposed implementations based on convex lifting may perform this task in a very efficient way, in particular the one described by Algorithm 6 exploiting clipping, without the need to carry out expensive point location. In total, it requires a constant number of $2|\mathcal{I}^{\text{uns}}|d+2d_ud+2d_u$ FLOPs, which is a significant reduction in runtime complexity, even if $|\mathcal{I}^{\text{uns}}| = N$ was the case (typically $|\mathcal{I}^{\text{uns}}| \ll \sum_{i=1}^{N} c_i$).

It should be noted that all the above figures do not consider evaluating and storing the full optimizer as only its first element is required for implementation of PWA controllers in a receding horizon fashion.

In terms of complexity reduction in explicit MPC, one may compare the proposed convex lifting approach, e.g., with the clipping-based implementation of [20] as they both exploit the concept of clipping. The latter, however, relies on replacing some of the saturated regions by extensions of the unsaturated ones, whereas the achievable reduction may range from none to the case when the new partition has $|\mathcal{I}^{\text{uns}}|$ regions. Another technique of [21] in turn requires to only store the unsaturated regions by employing a separating function. Clearly, both of the aforementioned approaches necessitate storing a modified state-space partition, and hence performing the point location at each sampling instant. Alternatively, a region-free implementation of explicit MPC was proposed in [10], and recently extended in [22]. Its nature, however, renders it applicable for MPC problems with rather larger parametric space and short prediction horizons.

V. Illustrative example

To illustrate the above proposed schemes, consider the double integrator system:

$$x_{k+1} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0.125 \\ 0.5 \end{bmatrix} u_k. \tag{53}$$

We design a PWA controller based on linear model predictive control which minimizes the following quadratic cost function:

$$\sum_{i=0}^{N_{\text{p}}-1} \left(x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k}\right) + x_{k+N_{\text{p}}|k}^T P x_{k+N_{\text{p}}|k},$$

where $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 10$, P is set to be the solution of discrete-time Riccati equation and the prediction horizon N_{p} equal to 10. This problem is subject to the following constraints:

$$\begin{aligned} &-2 \le u_{k+i|k} \le 2 \quad \text{for} \quad 0 \le i \le N_{\text{p}}-1, \\ &x_{k+N_{\text{p}}|k} \in \mathcal{X}_{\text{f}}, \end{aligned}$$

where $\mathcal{X}_{\text{f}}$ denotes the terminal constraint set as the maximal positively invariant set associated with the local control law $u = \begin{bmatrix} -0.2554 & -0.7590 \end{bmatrix} x$. The above problem is explicitly solved using MPT 3.0 [19]. The resulting PWA controller is presented in Fig.7 with the red and green controllers denoting where the control action attains the minimal and maximal values, respectively. In particular, the corresponding state-space partition consists of 73 unsaturated regions, 160 regions where $u(x) = -2$ and 160 regions where $u(x) = 2$. Storing all the 393 regions in this case amounts to 4758 real numbers, with additional 1179 real numbers needed to encode the PWA control law. Assuming double precision arithmetics, the total memory footprint of the original MPC controller is 48 kilobytes. The worst-case computational effort required for its online evaluation is 7934 FLOPs.

Fig.8 in turn depicts a convex lifting obtained from Algorithms 2 and 5. The associated state-space partition is shown merely for illustration and it needs not to be constructed. This region-free MPC implementation hence requires to only store the convex lifting and the corresponding PWA feedback, which in total amounts to 6.7 kB, implying a reduction by a factor of 7.2. Evaluating the optimal control action requires 559 FLOPs, which is 14.2 times less than in the case of the original explicit solution.

Finally, a convex lifting constructed per Algorithm 2 and equation (50) is depicted in Fig.9. The total memory footprint

of the resulting region-free controller is 3.5 kB, i.e., the memory consumption is reduced by a factor of 13.7. Accordingly, the online evaluation effort reduces to mere 298 FLOPs, which is 26.6 times faster than the worst-case runtime of the original controller.

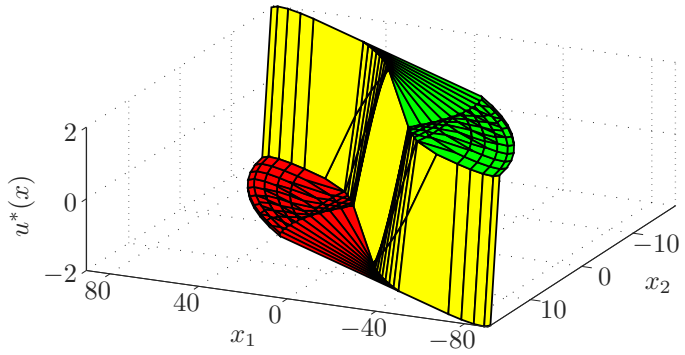


Fig. 7: The original PWA controller.

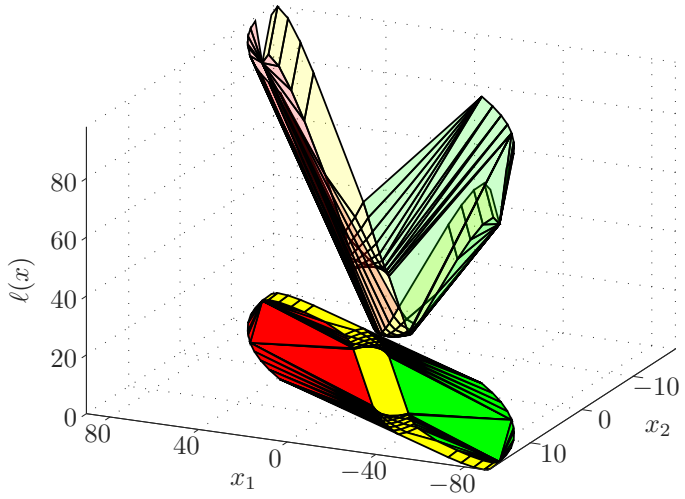


Fig. 8: A convex lifting obtained from Algorithms 2 and 5 and its associated state-space partition.

To assess how the controller complexity in the considered MPC example scales with problem size, in particular with the prediction horizon, we report the related memory consumption data in Table I. The proposed approaches are also compared with the clipping and separation based MPC implementations of [20] and [21], respectively. One may observe the significant complexity reduction of explicit solutions achieved via the convex lifting based techniques (denoted by CL), in particular the latter one, described in Section IV-B. The online evaluation effort is omitted here for brevity, however, it scales better in favor of the convex lifting based approach as it does not perform the traditional point location (c.f. Section IV-C). This clearly allows for controller deployment even on low-end embedded microcontroller platforms with limited storage capacity

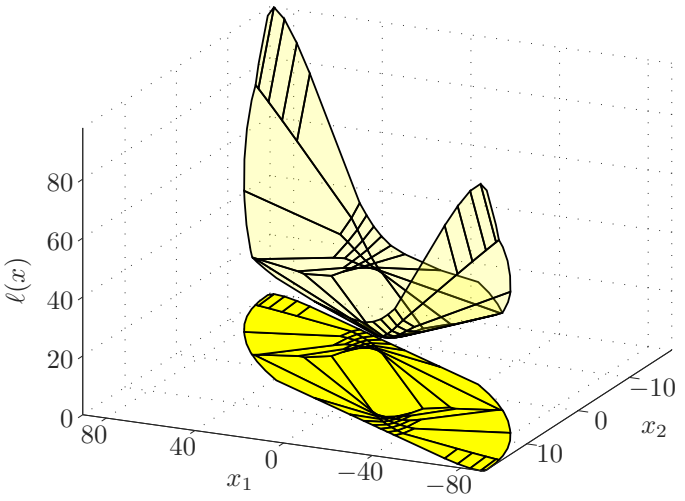


Fig. 9: A convex lifting obtained from Algorithms 2 and equation (50) and its associated state-space partition.

and computational power. Similarly, the offline computational time is substantially lower for the proposed approach (1-2 s to obtain $\ell^{\text{uns}}(x)$ for $N_{\text{p}} = 50$). We remark that the approach is accordingly applicable for problems involving a higher number of parameters or optimization variables.

TABLE I: Memory consumption in kilobytes for different implementations of the MPC example

	$N_p = 20$	30	40	50
Original explicit solution	133.4	265.1	445.7	681.7
Clipping-based approach [20]	49.6	95.3	242.8	401.9
Separation-based approach [21]	12.1	14.5[a]	17.0[a]	21.1[a]
CL-based approach per Sec. IV-A	10.8	14.6	18.5	24.1
CL-based approach per Sec. IV-B	4.8	5.7	6.7	8.3

[a] These figures were obtained from theoretical formulas since the computation failed in these cases.

VI. Conclusion

This paper presented the concept of convex liftings and its application in control theory. Accordingly, two different algorithms to construct convex liftings have been put forward. This concept was also shown to be useful for efficient implementation of PWA controllers via two proposed schemes. These allow for significant reduction of both their storage requirements and runtime complexity. Finally, a numerical example was considered to illustrate advantages of the proposed technique compared with existing methods.

References

[1] G. Aloupis, H. Pérez-Rosés, G. Pineda-Villavicencio, P. Taslakian, and D. Trinchet-Almaguer, "Fitting voronoi diagrams to planar tesselations," in *Combinatorial Algorithms*. Springer, 2013, pp. 349–361.

[2] F. Aurenhammer, "Criterion for the affine equivalence of cell complexes in r^d and convex polyhedra in r^{d+1}," *Discrete and Computational Geometry*, vol. 2, pp. 49–64, 1987.

[3] ——, "Power diagrams: properties, algorithms and applications," *SIAM Journal on Computing*, vol. 16, no. 1, pp. 78–96, 1987.

[4] ——, "Voronoi diagrams: A survey of a fundamental data structure," *ACM Comput. Surveys*, vol. 23, pp. 345–405, 1991.

[5] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," *Discrete & Computational Geometry*, vol. 8, no. 1, pp. 295–313, 1992.

[6] M. Baotic, F. Borrelli, A. Bemporad, and M. Morari, "Efficient on-line computation of constrained optimal control," *SIAM Journal on Control and Optimization*, vol. 47, no. 5, pp. 2470–2489, 2008.

[7] F. Bayat, T. A. Johansen, and A. A. Jalali, "Using hash tables to manage the time-storage complexity in a point location problem: Application to explicit model predictive control," *Automatica*, vol. 47, no. 3, pp. 571–577, 2011.

[8] ——, "Flexible piecewise function evaluation methods based on truncated binary search trees and lattice representation in explicit mpc," *Control Systems Technology, IEEE Transactions on*, vol. 20, no. 3, pp. 632–640, 2012.

[9] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[10] F. Borrelli, M. Baotić, J. Pekar, and G. Stewart, "On the computation of linear model predictive control laws," *Automatica*, vol. 46, no. 6, pp. 1035–1041, 2010.

[11] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[12] H. Crapo and W. Whiteley, "Plane self stresses and projected polyhedra 1: the basic pattern," *Structural Topology*, vol. 19, pp. 55–73, 1993.

[13] ——, "Spaces of stresses, projections and parallel drawings for spherical polyhedra," *Contributions to Algebra and Geometry*, vol. 35, no. So. 2, pp. 259–281, 1994.

[14] H. Edelsbrunner and R. Seidel, "Voronoi diagrams and arrangements," *Discrete Comput. Geom.*, 1986.

[15] T. Gal, *Postoptimal analyses, parametric programming and related topics*. Walter de Gruyter, 1995.

[16] B. Grünbaum, *Convex polytopes*. Wiley Interscience, 1967.

[17] M. Gulan, N. A. Nguyen, S. Olaru, P. Rodriguez-Ayerbe, and B. Rohal'-Ilkiv, "Implications of inverse parametric optimization in model predictive control." in *Developments in Model-Based Optimization and Control*, S. Olaru, A. Grancharova, and F. L. Pereira, Eds. Springer, 2015.

[18] D. Hartvigsen, "Recognizing voronoi diagrams with linear programming," *ORSA J. Comput.*, 1992.

[19] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proc. of the European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/ mpt.

[20] M. Kvasnica and M. Fikar, "Clipping-based complexity reduction in explicit MPC," *Automatic Control, IEEE Transactions on*, vol. 57, no. 7, pp. 1878–1883, 2012.

[21] M. Kvasnica, J. Hledík, I. Rauová, and M. Fikar, "Complexity reduction of explicit model predictive control via separation," *Automatica*, vol. 49, no. 6, pp. 1776–1781, 2013.

[22] M. Kvasnica, B. Takács, J. Holaza, and S. Di Cairano, "On region-free explicit model predictive control," in *IEEE 54th Annual Conference on Decision and Control*, Osaka, Japan, 2015, pp. 3669–3674.

[23] J. C. Maxwell, "On reciprocal diagrams and diagrams of forces," *Philosophical Magazine*, vol. ser. 4. 27, pp. 250–261, l864.

[24] N. A. Nguyen, "Explicit robust constrained control for linear systems: analysis, implementation and design based on optimization," Ph.D. dissertation, CentraleSupélec, Université Paris-Saclay, France, 11/2015.

[25] N. A. Nguyen, S. Olaru, and P. Rodriguez-Ayerbe, "Any discontinuous PWA function is optimal solution to a parametric linear programming problem," in *54th IEEE Conference on Decision and Control, Osaka, Japan*, 2015.

[26] ——, "Inverse parametric linear/quadratic programming problem for continuous PWA functions defined on polyhedral partitions of polyhedra," in *54th IEEE Conference on Decision and Control, Osaka, Japan*, 2015.

[27] ——, "On the complexity of the convex liftings-based solution to inverse parametric convex programming problems," in *European Control Conference, Linz, Austria*, 2015.

[28] ——, "Recognition of additively weighted voronoi diagrams and weighted delaunay decompositions," in *European Control Conference, Linz, Austria*, 2015.

[29] N. A. Nguyen, S. Olaru, P. Rodriguez-Ayerbe, M. Hovd, and I. Necoara, "Constructive solution to inverse parametric linear/quadratic programming problems via convex liftings." [Online]. Available: https://hal-supelec.archives-ouvertes.fr/hal-01207234/document

[30] ——, "Inverse parametric convex programming problems via convex liftings," in *19th IFAC World Congress, Cape Town, South Africa*, 2014.

[31] ——, "On the lifting problems and their connections with piecewise affine control law design," in *European Control Conference, Strasbourg, France*, 2014.

[32] K. Rybnikov, "Polyhedral partitions and stresses," Ph.D. dissertation, Queen University, Kingston, Ontario, Canada, 1999.

[33] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[34] A. Schulz, "Lifting planar graphs to realize integral 3-polytopes and topics in pseudo-triangulations," Ph.D. dissertation, Fachbereich Mathematik und Informatik der Freien Universitat Berlin, 2008.

[35] P. Tøndel, T. A. Johansen, and A. Bemporad, "Evaluation of piecewise affine control via binary search tree," *Automatica*, vol. 39, no. 5, pp. 945–950, 2003.