

# Chapter 3

## Implications of Inverse Parametric Optimization in Model Predictive Control

Martin Gulan, Ngoc Anh Nguyen, Sorin Oлару, Pedro Rodriguez-Ayerbe and Boris Rohal'-Ilkiv

**Abstract** Recently, inverse parametric linear/quadratic programming problem was shown to be solvable via convex liftings approach [13]. This technique turns out to be relevant in explicit model predictive control (MPC) design in terms of reducing the prediction horizon to at most two steps. In view of practical applications, typically leading to problems that are not directly invertible, we show how to adapt the inverse optimality to specific, possibly convexly non-liftable partitions. Case study results moreover indicate that such an extension leads to controllers of lower complexity without loss of optimality. Numerical data are also presented for illustration.

### 3.1 Introduction

Interest in parametric optimization algorithms for linear and quadratic programs has been resurged during the last 15 years, following the observation that certain optimal control problems for constrained systems can be solved explicitly offline. In particular, a lot of attention has been attracted by the fact that solutions to the

---

M. Gulan (✉) · B. Rohal'-Ilkiv  
Institute of Automation, Measurement and Applied Informatics,  
Faculty of Mechanical Engineering, Slovak University of Technology,  
Námestie slobody 17, 812 31 Bratislava, Slovakia  
e-mail: martin.gulan@stuba.sk

B. Rohal'-Ilkiv  
e-mail: boris.rohal-ilkiv@stuba.sk

N.A. Nguyen · S. Oлару · P. Rodriguez-Ayerbe  
Laboratory of Signals and Systems, CentraleSupélec-CNRS-Université Paris-Sud, Université  
Paris-Saclay, 3, rue Joliot Curie, 911 90 Gif-sur-Yvette, France  
e-mail: Ngocanh.Nguyen@supelec.fr

S. Oлару  
e-mail: Sorin.Oлару@centralesupelec.fr

P. Rodriguez-Ayerbe  
e-mail: Pedro.Rodriguez@centralesupelec.fr

traditional model predictive control problems can be obtained in an explicit formulation by exploiting the concept of parametric convex programming (pCP) [3–5, 19, 22]. This allows to pre-compute the optimal solution to a linear/quadratic programming problem for a range of operating conditions of interest as a piecewise affine (PWA) function defined over a polyhedral partition of the parameter space. The implementation effort of these so-called explicit MPC (eMPC) methods [1] hence reduces to a simple function evaluation, since the controller itself maps into a lookup table of linear gains. This property naturally circumvents the main implementation drawback of the standard implicit MPC, which is the need for online optimization in the receding horizon fashion at each sampling instant. Clearly, within the real-time requirements, the solutions may become too expensive or simply infeasible, namely in the case of systems with fast-evolving dynamics. On the other hand, the efficacy of the offline solution is limited to small-dimensional systems. Therefore the eMPC algorithms necessarily encounter computational difficulties as the parameter space dimension or the prediction horizon increases. Moreover, the resulting PWA controller is often too complex; hence, if allowable, its structure may need to be treated adequately to satisfy the restrictions imposed by the implementation hardware.

In view of handling linear MPC problems often emerging from practical applications, the parametric convex programming has grown into a mature technique providing a variety of numerical algorithms available for effective solution. Recently, the concept of inverse optimality has been shown to provide a new perspective on the structural link between linear MPC and pCP problem formulation via the technique of inverse parametric convex programming (IpCP). As the name suggests, it is defined as an inverse optimality problem of parametric convex programming. It aims to build an alternative optimization problem characterized by an appropriate constraint set and a cost function such that its optimal solution coincides with the one of the original problem. More specifically, the goal of inverse parametric linear/quadratic programming is to construct a linear constraint set and a linear/quadratic cost function such that the optimal solution of this newly formulated problem is equivalent to a given PWA function defined over a given polyhedral partition. This topic has been recently investigated in [2, 7, 13], and turned out to be applicable in the complexity reduction of piecewise affine control law design. This chapter closely follows the results put forward in [13, 14]. Therein, a constructive procedure based upon convex liftings is presented, leading to an important implication in MPC design, that can be stated as follows: every continuous piecewise affine control law can be recovered via an MPC problem with a control horizon at most equal to two prediction steps. Central to the approach is the operation of convex lifting, and the related liftability condition, which plays a crucial role in the existence of an inverse optimal solution.

The main aim of this study is to address the difficulty arising in applications of IpCP in linear MPC, namely that the explicit solution of a linear MPC problem with respect to a quadratic cost function is not, in many cases, convexly liftable. With regard to this issue, in [14] it was shown that, in fact, any parameter space partition associated with a solution of a pQP can be sub-partitioned such that its convex liftability will be guaranteed. In control theory, however, such a subdivision necessarily increases the complexity of PWA control laws, which is commonly urged to

be kept as low as possible for the sake of efficient eMPC implementations. Therefore, herein we rather aim at devising an alternative, yet convenient approach that would target pQP-based MPC problems which are typically not directly invertible, without generating any gain in complexity of the inherited solution. To tackle this challenge, we recall the open question of minimal complexity of the inverse optimality formulation, discussed in [12]. It is known that the IpL/QP via convex liftings replaces the constraint set of linear MPC problem based on prediction along a finite horizon with mixed constraints on the control and state variables over two prediction stages. Nevertheless, the main advantage of a significant decrease in the dimension of optimization arguments tends to come at the price of a relatively large set of constraints. Since not all the constraints practically contribute to the optimal solution, an effort is being put into reducing the constraint set towards a minimal representation necessary for the inverse optimality formulation. We exploit this fact and present an algorithmic procedure tailored for the class of MPC problems of interest. At this point the contribution of the proposed extended IpLP approach is twofold as it simultaneously treats the issue of invertibility, preconditioned by convex liftability, and eliminates the redundant constraints from the formulation. The procedure, however, exactly retrieves only a part of the solution. To establish the equivalence of the remaining portion of controller structure we adopt the concept of clipping strategy proposed in [10]. This inverse optimality-based procedure is performance lossless, i.e., the retrieved PWA function inherits all the performance, closed-loop stability and feasibility guarantees of the pQP-based/online MPC solution.

As outlined earlier, the efficient implementation of eMPC, often using embedded platforms, is closely associated with the structure of the explicit controller. Since the PWA control law is defined over a set of polytopic regions, it is clear that if the number of regions becomes large, the storage capacity of the implementation hardware may be easily exceeded. To overcome these limitations, numerous methods for complexity reduction via simplifying eMPC optimizers have been proposed. This is usually achieved by devising less complex equivalent replacement functions of the original PWA feedback with no implications on optimality, or various sophisticated, albeit suboptimal, approximations. A brief, yet recent overview can be found, e.g., in [9, 11]. We recall this topic with regard to the third feature of the approach. As it will be evidenced later, the combination of extended IpLP with the use of a simple clipping filter [10] may find its use in case a low-complexity eMPC controller is sought. Nevertheless, this ability is not an objective of this study, rather an additional gain of the approach. Hence, when an online implementation is of interest and certain assumptions are fulfilled, the less complex, yet performance-wise optimal explicit representation can offer an important advantage. In particular, real-time implementation complexity is determined not only by the memory needed to store the controller, but also by the computational time necessary to find and evaluate the corresponding control law. This is of imminent importance when targeting embedded hardware or controlling fast systems. In this light, extended IpLP with clipping shows its potential use in practical MPC applications; in comparison with the generic convex liftings-based IpCP, which as such does not yield any computational benefits within online eMPC. On the contrary, the obtained explicit solution may in case of

directly non-invertible problems become too complex and thus intractable. Additionally, the proposed inverse optimization formulation may be as well cast and solved as an online MPC problem which would shed more light on the merits of its cost/constraints rearrangement.

### 3.2 Notation and Definitions

Throughout this study,  $\mathbb{R}$ ,  $\mathbb{N}_{>0}$  are used to denote the set of real numbers, and the field of positive integers, respectively. For a compact notation, let us define the index set  $\mathcal{I}_N := \{i \in \mathbb{N}_{>0} \mid i \leq N\}$ , for a given  $N \in \mathbb{N}_{>0}$ . For a point  $x \in \mathbb{R}^d$ , by  $\mathbb{R}^{n_x}$  we denote the vector space containing the point  $x$ , hence in this case  $\mathbb{R}^{n_x} = \mathbb{R}^d$ . Next, given a finite set  $\mathcal{S} = \{x_1, x_2, \dots, x_n\}$  of  $n$  points,  $\text{Card}(\mathcal{S})$  denotes the cardinal number of the set  $\mathcal{S}$ . By  $\text{conv}(\mathcal{S})$  we denote its convex hull. Also, if  $\mathcal{S}$  is a finite set of rays, i.e.,  $\mathcal{S} = \{y_1, \dots, y_n\}$  then  $\text{cone}(\mathcal{S})$  represents the cone defined as follows:

$$\text{cone}(\mathcal{S}) = \{t_1 y_1 + \dots + t_n y_n : t_i \geq 0, \forall 1 \leq i \leq n\} \quad (3.1)$$

A polyhedron is defined as a convex intersection of finitely many closed half-spaces. A polytope is a bounded polyhedron. Given a full-dimensional polyhedron  $\mathcal{S}$  in  $\mathbb{R}^d$ , then  $\mathcal{V}(\mathcal{S})$  denotes the set of its vertices;  $\mathcal{R}(\mathcal{S})$  denotes the set of its rays. Also,  $\text{int}(\mathcal{S})$  denotes its interior.  $\text{Proj}_{\mathbb{S}} \mathcal{S}$  represents the orthogonal projection of the set  $\mathcal{S}$  onto a subspace  $\mathbb{S}$  of  $\mathbb{R}^d$ . Further, a face of  $\mathcal{S}$  is the intersection of  $\mathcal{S}$  and one of its supporting hyperplanes. A face of dimension  $d - 1$  is called a facet. The set of all facets of polyhedron  $\mathcal{S}$  is denoted as  $\mathcal{F}(\mathcal{S})$ . Given a function  $\kappa(x)$ ,  $\text{dom}(\kappa(x))$  denotes its domain. In addition, given two sets  $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^d$ , by  $\mathcal{S}_1 \oplus \mathcal{S}_2$  we denote their Minkowski sum defined as follows:

$$\mathcal{S}_1 \oplus \mathcal{S}_2 := \{y \in \mathbb{R}^d \mid \exists x_1 \in \mathcal{S}_1, x_2 \in \mathcal{S}_2 \text{ s.t. } y = x_1 + x_2\}.$$

Next, let us recall some necessary definitions.

**Definition 3.1** A collection of  $N \in \mathbb{N}_{>0}$  full-dimensional polyhedra  $\mathcal{X}_i \subset \mathbb{R}^d$ , denoted as  $\mathcal{X} = \{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$ , is called a polyhedral partition of a polyhedron  $\Omega \subseteq \mathbb{R}^d$  if:

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

In addition, we refer to polyhedra  $(\mathcal{X}_i, \mathcal{X}_j)$  as neighbours, or adjacent, if  $(i, j) \in \mathcal{I}_N^2$ ,  $i \neq j$  and  $\dim(\mathcal{X}_i \cap \mathcal{X}_j) = d - 1$ . If a polyhedral partition is a collection of polytopes, then it is called a polytopic partition.  $\mathcal{X}_i$  are referred to as regions of the partition  $\mathcal{X}$ .

In this chapter, a cell complex is understood as a polyhedral partition whose facet-to-facet property [20] is fulfilled, meaning that any two neighbouring regions share a common facet. Note that a complete, more general definition of cell complex can

be found in [6]. However, for simplicity, we restrict our attention to the property of interest.

**Definition 3.2** Given a polyhedral partition  $\mathcal{X} = \{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$  of a polyhedron  $\Omega \subseteq \mathbb{R}^d$ , a convex lifting is described by the function  $z : \Omega \rightarrow \mathbb{R}$  with:

$$z(x) = a_i^T x + b_i \quad \text{for any } x \in \mathcal{X}_i, \quad (3.2)$$

and  $a_i \in \mathbb{R}^d$ ,  $b_i \in \mathbb{R}$ ,  $\forall i \in \mathcal{I}_N$ ; such that the following conditions hold:

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

According to Lemma 1 in [15], a polyhedral partition admitting a convex lifting is required to be a cell complex. Conversely, a cell complex is not necessarily convexly liftable. The necessary and sufficient conditions for a cell complex to be convexly liftable<sup>1</sup> are referred to [14] and the references therein.

Moreover, a convex lifting can be in fact seen as a convex surface composed of the convex lower boundary of a polyhedron in an augmented space. This particular polyhedron can be described by the following definition:

**Definition 3.3** A given cell complex  $\mathcal{X} = \{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$  of a polyhedron  $\Omega \subseteq \mathbb{R}^d$  has an affinely equivalent polyhedron if there exists a polyhedron  $\tilde{\mathcal{X}} \subset \mathbb{R}^{d+1}$  such that for each  $i \in \mathcal{I}_N$ :

- (1)  $\exists F_i \in \mathcal{F}(\tilde{\mathcal{X}})$  satisfying:  $\text{Proj}_{\mathbb{R}^d} F_i = \mathcal{X}_i$ ,
- (2) if  $\underline{z}(x) = \min_z z \quad \text{s.t. } [x^T \ z]^T \in \tilde{\mathcal{X}}$ , then  $[x^T \ \underline{z}(x)]^T \in F_i$  for  $x \in \mathcal{X}_i$ .

The second condition in Definition 3.3 implies that the set of facets of  $\tilde{\mathcal{X}}$  at the lower values of  $z$  is exclusively considered. These lower facets  $F_i$  build a convex surface in  $\mathbb{R}^{d+1}$  known as the previously defined convex lifting, and their image via the orthogonal projection onto  $\mathbb{R}^d$  recovers the cell complex  $\mathcal{X}$ .

We remark that an additional set of definitions is provided within Sect. 3.5 as it closely relates to the problems treated therein.

### 3.3 Inverse Parametric Optimization via Convex Liftings

This section aims to recall the constructive solution to the inverse parametric linear/quadratic programming problem via convex liftings. For ease of presentation, let us start from a larger perspective—with the definition of a parametric convex<sup>2</sup> programming problem, which can be cast as follows:

<sup>1</sup>Note that the liftability condition does not restrict  $\mathcal{X}$  to be a partition of a polytope/polyhedron. This property, however, becomes important in the context of convexity of inverse parametric linear/quadratic programming problems.

<sup>2</sup>Convexity of the optimization problem (3.3) is ensured by convexity of  $f$ ,  $g_i$ , and  $h_j$  in  $\theta$  and  $x$ .

$$\begin{aligned}
& \min_{\theta} f(\theta, x) \\
& \text{s.t. } g_i(\theta, x) \leq 0, \quad \forall i \in \mathcal{I}_p, \\
& \quad h_j(\theta, x) = 0, \quad \forall j \in \mathcal{I}_q,
\end{aligned} \tag{3.3}$$

where  $\theta, x$  denote the decision variable and the parameter, respectively. The terms  $g_i(\theta, x), h_j(\theta, x)$  represent the left-hand sides of respective inequality and equality constraints for the optimization problem (3.3).

In view of parametric linear/quadratic programming, the equality constraints are omitted for simplicity. Moreover,  $g_i(\theta, x), \forall i \in \mathcal{I}_p$  define linear constraints on  $\theta$  and/or  $x$ , from the geometrical point of view representing a polyhedron in the augmented space of  $[\theta^T \ x^T]^T$ . Still,  $f(\theta, x)$  stands for a linear or quadratic cost function, which has the following description:

$$f(\theta, x) = \theta^T H \theta + (F^T x + Y)^T \theta, \tag{3.4}$$

with a positive semidefinite matrix  $H = H^T \succeq 0$  and appropriately dimensional matrices  $F, Y$ .

It is well known from the works of [4, 19, 22] that the optimal solution to a parametric linear/quadratic programming problem is a piecewise affine function defined over a polyhedral partition. It is shown therein that the optimal solution to a pQP problem with respect to  $H \succ 0$  is continuous and unique. Otherwise, the continuity and the uniqueness properties may not be preserved for the case of a pLP. Fortunately, it is shown e.g., in [17] that an equivalently continuous optimal solution can always be selected. Therefore, the IpL/QP can focus on continuous PWA functions.

Given a continuous PWA function defined over a polyhedral partition, IpL/QP aims to find an appropriate optimization problem, characterized by a linear/quadratic cost function and a set of linear constraints such that the optimal solution to this optimization problem is equivalent to the given PWA function. Mathematically, let  $f_{\text{pwa}}(x) : \Omega \subseteq \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u}$  denote a continuous PWA function to be recovered, defined as follows:

$$f_{\text{pwa}}(x) = F_i x + G_i \text{ for } x \in \mathcal{X}_i. \tag{3.5}$$

IpL/QP aims to find a cost function  $J(x, z, u)$  and a set of four matrices  $H_x, H_u, H_z, K$  such that:

$$f_{\text{pwa}}(x) = \text{Proj}_{\mathbb{R}^{n_u}} \arg \min_{\begin{bmatrix} z^T & u^T \end{bmatrix}^T} J(x, z, u) \text{ s.t. } H_x x + H_z z + H_u u \leq K, \tag{3.6}$$

where  $z$  represents an auxiliary variable for the recovered optimization problem and will be shown to be a 1-dimensional variable.

**Assumption 3.1** As reported in [13, 15], this inverse optimality problem is valid under the following standing assumptions:

A1.1 The polyhedral partition  $\mathcal{X}$  is the partition of a polyhedron  $\Omega$ .

A1.2 The polyhedral partition  $\mathcal{X}$  is convexly liftable.

Note that Assumption A1.1 is of help in order to guarantee the convexity of the recovered optimization problem. On the other hand, Assumption A1.2 is not restrictive due to Theorem 2 and Lemma 2, also reported in [15]. Accordingly, if the given polyhedral partition does not satisfy Assumption A1.2, one can refine it such that the internal boundaries are maintained and the new partition is convexly liftable. Such subdivision will be discussed in more detail for the cases often encountered in MPC in Sect. 3.5.

As outlined in the introductory section, central to the constructive procedure of [13] is the operation of convex lifting for the given partition, recalled in Definition 3.2. An algorithm for the construction of convex liftings is referred to [14] and extended to a cell complex of a polyhedron via Theorem 1 in [15]. The following sets are also defined:

$$\begin{aligned}
 V_x &= \bigcup_{i \in \mathcal{I}_N} \mathcal{V}(\mathcal{X}_i), \quad R_x = \bigcup_{i \in \mathcal{I}_N} \mathcal{R}(\mathcal{X}_i), \\
 V_{[x^T \ z \ u^T]^T} &= \left\{ \left[ x^T \ z(x) \ f_{\text{pwa}}^T(x) \right]^T \mid x \in V_x \right\}, \\
 R_{[x^T \ z \ u^T]^T} &= \left\{ \left[ r^T \ \hat{z}(r) \ \hat{f}^T(r) \right]^T \mid r \in R_x, \begin{array}{l} \hat{z}(r) = a_i^T r \\ \hat{f}(r) = F_i r \end{array} \text{ if } r \in \mathcal{R}(\mathcal{X}_i) \right\}, \quad (3.7) \\
 \Pi_v &= \text{conv}(V_{[x^T \ z \ u^T]^T}), \quad \Pi_r = \text{cone}(R_{[x^T \ z \ u^T]^T}), \\
 \Pi &= \Pi_v \oplus \Pi_r.
 \end{aligned}$$

Following (3.7), recall that  $V_x, R_x$  denote the sets of vertices and rays of the cell complex  $\mathcal{X}$ , respectively. Also,  $V_{[x^T \ z \ u^T]^T}, R_{[x^T \ z \ u^T]^T}$  stand for the sets of augmented vertices and rays of  $\mathcal{X}$ , into  $\mathbb{R}^{n_x+n_u+1}$ . Note, however, that under convex liftings, the augmented terms for a vertex are different from the one for a ray as shown in (3.7). More clearly, consider a vertex  $v$  and a ray  $r$  of region  $\mathcal{X}_i$  in the cell complex  $\mathcal{X}$ . After embedding to the space of convex lifting, the augmented term corresponding to  $v$  is  $a_i^T v + b_i$ . However, under this embedding, the augmented term of  $r$  becomes  $a_i^T r$ . The resulting constraint set  $\Pi$  is computed via the Minkowski sum of  $\Pi_v$  and  $\Pi_r$ . This is basically due to Minkowski-Weyl theorem (see Corollary 7.1b in [18]).

Based on the above notation, we recall here the main result for IpL/QP problem via the following theorem.

**Theorem 3.1** *Given a continuous PWA function  $f_{\text{pwa}}(x)$ , defined over a cell complex  $\mathcal{X} = \{\mathcal{X}_i\}_{i \in \mathcal{I}_N}$  satisfying Assumptions A1.1, A1.2, then  $f_{\text{pwa}}(x)$  is the image via the orthogonal projection of the optimal solution to the following parametric linear programming problem:*

$$\min_{\begin{bmatrix} z \\ u^T \end{bmatrix}^T} z \quad \text{subject to} \quad \begin{bmatrix} x^T & z & u^T \end{bmatrix}^T \in \Pi. \quad (3.8)$$

For the proof of Theorem 3.1 we refer to Sect. 5 of [15]. Note, however, that as discussed in [12], not all constraints of  $\Pi$  are meaningful in (3.8). Many of them contribute to the construction of feasible region instead of building its optimal solution. Therefore, these constraints may need to be removed to simplify the optimization problem. Algorithms put forward in [12] can be effectively utilized to carry out this procedure.

We remark that this theorem is meaningful in the context of implicit MPC where it can help to overcome its computational limitations by implementing this alternative optimization approach and solving the LP<sup>3</sup> (3.8) at each sampling time. In this study we rather exploit this concept for efficient explicit MPC implementations. This point is discussed in Sect. 3.5 and clarified via a case study in Sect. 3.6.

### 3.4 Application to Linear MPC Problems

This section aims to recall an important result of IpL/QP related to the linear model predictive control. Consider a discrete-time, linear invariant system:

$$x(k+1) = Ax(k) + Bu(k), \quad (3.9)$$

where  $(A, B)$  is stabilizable. In MPC problems, a cost function is typically defined over a finite prediction horizon  $N \in \mathbb{N}_{>0}$  as follows:

$$J(U, x(k)) = \sum_{i=0}^{N-1} \ell_i(x_{k+i|k}, u_{k+i|k}) + V_N(x_{k+N|k}) \quad (3.10)$$

where  $x_{k+i|k} \in \mathbb{R}^{n_x}$  is the  $(k+i)$ -th prediction of the system state at time  $k$ ,  $u_{k+i|k}$  denotes the control action at time instant  $k+i$ , and  $U = [u_{k|k}^T, \dots, u_{k+N-1|k}^T]^T$ .

The  $\ell_i(x_{k+i|k}, u_{k+i|k})$  term of the objective represents a stage cost  $\forall i \in \mathcal{I}_{N-1} \cup \{0\}$  and  $V_N(x_{k+N|k})$  denotes the terminal penalty. Moreover, the state and control variables are typically required to satisfy constraints:

$$x_{k+i|k} \in \mathbb{X}, u_{k+i|k} \in \mathbb{U}, \forall i \in \mathcal{I}_{N-1} \cup \{0\}, x_{k+N|k} \in \mathbb{X}_f \quad (3.11)$$

with polyhedra  $\mathbb{X}, \mathbb{U}$  containing the origin in their interior. A suitable terminal constraint set  $\mathbb{X}_f \subset \mathbb{X}$  is considered to guarantee closed-loop stability.

A linear MPC problem thus aims to solve the following optimization problem:

$$\min_U J(U, x(k)) \quad \text{subject to} \quad (3.11) \quad (3.12)$$

where the stage costs and the terminal cost tend to take one of the following forms:

---

<sup>3</sup>The inverse optimality problem can be as well cast and solved implicitly/explicitly as a QP/pQP.



- (1) 2-norm:  $\ell_i(x_{k+i|k}, u_{k+i|k}) = x_{k+i|k}^T Q x_{k+i|k} + u_{k+i|k}^T R u_{k+i|k}$ ,  $V_N(x_{k+N|k}) = x_{k+N|k}^T P x_{k+N|k}$ , where  $Q = Q^T$  is a positive semidefinite matrix and  $R, P$  are symmetric, positive definite matrices,
- (2)  $1/\infty$ -norm:  $\ell_i(x_{k+i|k}, u_{k+i|k}) = \|Q x_{k+i|k}\|_p + \|R u_{k+i|k}\|_p$ ,  $V_N(x_{k+N|k}) = \|P x_{k+N|k}\|_p$ , where  $p = 1/\infty$  and  $Q, R, P$  are of appropriate dimension.

As shown in [3, 4], the optimal solutions of such linear MPC problems of modest size can be found explicitly by parametric linear/quadratic programming, as follows:

$$U^* = \arg \min_U J(U, x(k)) \quad \text{s.t.} \quad GU \leq W + Ex(k), \quad (3.13)$$

where the control input sequence  $U$  is regarded as the decision variable, the current state  $x(k)$  represents the parameter, and  $G, W, E$  are appropriate matrices describing the constraints (3.11). In the implementation, the receding horizon MPC feedback becomes  $u_k = \text{Proj}_{\mathbb{R}^{n_u}} U^*$ . This explicit solution to a linear MPC problem has a piecewise affine structure, and therefore it inherits also the properties of an inverse optimality problem recalled earlier. Based on this property, the main result in this section is summarized via the following theorem.

**Theorem 3.2** *The continuous explicit solution of a generic linear MPC problem with respect to a linear/quadratic cost function is equivalently obtained through a linear MPC problem with a linear cost function and the control horizon at most equal to two prediction steps.*

Interested reader is further referred to [13] for the proof. Therein, numerical examples can also be found, illustrating the application of the generic inverse optimality solution to problems that are directly invertible. This allows one to recover the optimal explicit continuous feedback inherited from the linear MPC problem via an inverse pL/QP problem, while the structure of the PWA control law and the underlying parameter partition are maintained. The following section is going further and focuses on more realistic MPC scenarios where the convex liftability of the state-space partition is not a priori guaranteed, hence requiring to appropriately revisit the IpCP procedure in order to find an inverse problem formulation and its solution.

### 3.5 A Novel Approach for a Class of Primarily Non-invertible Control Laws Obtained from MPC Problems

As outlined in the introduction, we aim at recovering linear MPC problems whose explicit solution obtained by parametric convex programming leads to a convexly non-liftable parameter partition, which therefore may not be used within the generic IpCP scheme proposed in [13] where the authors build upon the assumption that the state-space partition to be recovered is a convexly liftable cell complex (cf. Definition 3.2). Here, we are, however, interested in situations when the partition inherited

from the solution of an MPC problem is found to be a convexly non-liftable polyhedral partition or cell complex, eventually. As mentioned recently in [14], the parameter space partition associated with the optimal solution to a parametric quadratic programming problem, is in many cases not convexly liftable. We remark that this tends to be the case for most of practical set-ups involving input constraints, usually yielding explicit controllers with many saturated regions.<sup>4</sup> One may observe that the larger their number is, the smaller the likelihood of constructing a convex lifting gets. In such a case, a straightforward scheme to approach this problem would be to rearrange the possibly disconnected and non-convex union of polyhedra sharing the same saturated control law, in a way to allow for convex liftability of the entire partition. Despite the existence of several region merging techniques, none of them can in fact guarantee this property. Even if we manage to remove the saturated regions completely, as per separation-based complexity reduction of [11], the collection of unsaturated polyhedra left to be lifted may very often be non-convex, which prevents us from constructing its affinely equivalent polyhedron while recovering the explicit controller without giving rise to some additional regions.

This problem may be as well treated by an appropriate sub-partitioning of a given convexly non-liftable polyhedral partition, as outlined recently in [14]. Therein, it is shown that any parameter space polyhedral partition associated with a solution of a pQP problem can be sub-partitioned such that its internal boundaries are preserved and the new cell complex is convexly liftable. Despite the new hyperplane arrangement allows the convex liftability to be retrieved, it naturally leads to a refining that can easily increase the complexity of a PWA feedback to a large extent. Implementation of such an eMPC controller may thus become a costly, if not intractable, alternative to the original one.

In view of the aforementioned practical aspects, let us now briefly formulate an extended inverse optimality problem to be tackled henceforth.

**Problem 3.1** Given a, possibly, convexly non-liftable polyhedral partition/cell complex<sup>5</sup>  $\mathcal{R} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_R}$  of a polyhedron  $\Omega \subseteq \mathbb{R}^{n_x}$ , associated with a continuous piecewise affine function  $\kappa(x) = f_{\text{pwa}}(x) : \Omega \rightarrow \mathbb{R}^{n_u}$ , find a linear cost function  $J(x, z, u)$ , and matrices  $H_x, H_u, H_z, K$  such that the inverse optimal solution

$$\begin{cases} \tilde{\kappa}(x) = \text{Proj}_{\mathbb{R}^{n_u}} \arg \min_{[z \ u]^T} J(x, z, u), \\ \text{s.t. } H_x x + H_z z + H_u u \leq K. \end{cases} \quad (3.14)$$

returns an equivalent replacement of  $\kappa(x)$  when passed through a suitable filter  $\phi(\cdot)$ .

**Assumption 3.2** At this point, some assumptions need to be stated to make the present approach reasonable from both the construction and practical viewpoint:

<sup>4</sup>The definition of a un/saturated region is given in Sect. 3.5.1.

<sup>5</sup>Within this section and henceforth we slightly abuse the notation  $\mathcal{R}$  to refer to state-space partitions inherited from MPC problems.

- A2.1 The parametric linear programming (pLP) problems are exclusively considered as possible candidates for the inverse optimality solutions.
- A2.2  $\mathcal{R}$  is the partition of a polytope  $\Omega$ .
- A2.3  $\kappa(x)$  is the explicit optimal solution to a pQP-based linear MPC problem.
- A2.4  $\kappa(x)$  is partially saturated with the corresponding polytopic regions causing  $\mathcal{R}$  to be convexly non-liftable.

Assumption A2.1 provides a manageable framework for the constructive inverse optimality procedures by imposing linearity of the candidate pCP problems. Although a quadratic cost function may be used as well, in the scope of this study an IpLP is of interest for brevity, yet with a straightforward extension towards IpQP. Meanwhile, Assumption A2.2 merely requires the feasible set to be a polytope, which is usual in the context of MPC. However, it is not restrictive, and the procedure can be easily extended to the case where the feasible set is a polyhedron. The other two assumptions stem from a more practical reasoning outlined at the beginning of this section. By Assumption A2.3 we focus our attention rather on explicit solutions obtained by parametric quadratic programming. Considering pQP in this context not only relates to the nature of most linear MPC problems, but it is mainly due to the typically inherent curse of direct non-liftability of the associated solutions, defined over convexly non-liftable state-space partitions.<sup>6</sup> This is in contrast to pLP problems which are directly invertible without any refinement required [14], and also for this fact are not of interest here. The last assumption stems from numerous observations encountered in practical MPC problems where the explicit receding horizon feedbacks usually contain many regions for which the associated optimal control action is either constantly on the upper limit or constantly on the lower limit. This reasonable idea was also central to the works of [10, 11] aiming for complexity reduction of explicit MPC feedback laws. Recall that, in general,  $\mathcal{R}$  inherited from a pQP may be convexly non-liftable since it is mostly a strict polyhedral partition, i.e., facet-to-facet property does not hold [20]; however, it may as well be a cell complex that is not convexly liftable simply due to the nature of its hyperplane arrangement (see e.g., the example in [15]). The second part of Assumption A2.4 moreover hints that the liftability issue is typically induced by the existence and structural properties of the collections of aforementioned saturated polytopic regions. This becomes more and more likely as the cardinality of such collections increases, at the price of the remaining collection of unsaturated regions, which is itself thus in a vast majority of cases found to be convexly liftable. In fact, as it is evidenced, e.g., in [10, 11], a typical explicit MPC feedback law contains a significantly smaller number of unsaturated regions as compared to the number of saturated ones. In addition, if the convex liftability of the collection of unsaturated regions, as invoked via Assumption A2.4, practically happened not to hold, it is still possible to use sub-partitioning to preserve it. With the usual structural proportions mentioned above, it would however become still very cheap compared to a complete partition refining—actually, as a matter of fact, such a possible slight increase in controller complexity would be most likely

---

<sup>6</sup>This structural property is related to the piecewise quadratic cost function. This is to be compared with the piecewise linear and convex cost function of a pLP-based MPC problem.

eliminated by a notable complexity reduction indicated by the approach proposed in the following subsection. Needless to say, no such an issue has been encountered within tested practical MPC scenarios, neither is it present in the case study put forward in Sect. 3.6.

To review the objective, by exploiting the procedure of inverse parametric optimization via convex liftings [13], we aim to reconstruct an appropriate linear programming problem (3.14) with respect to a given piecewise affine function  $\kappa(x)$  defined over a given, possibly convexly non-liftable polyhedral partition/cell complex of the parameter space,  $\mathcal{R}$ , such that the optimal solution of this reconstructed problem is equivalent to the given PWA function. The equivalence requires that the boundary between two different regions of the parameter space partition corresponding to two different affine functions is preserved, whereas a proper rearrangement of the subset of the domain of definition, over which the PWA function is saturated, is allowable. This property is effectively exploited within this study. An example is the new convexly liftable cell complex  $\tilde{\mathcal{R}}$  corresponding to a structurally modified, yet optimal PWA solution function  $\phi(\tilde{\kappa}(x))$ , as shown in Sect. 3.5.

The following subsections present the main contribution of this study—the extension to convex liftings-based inverse parametric convex programming procedure, namely IpLP, with the main focus on a wide and challenging class of primarily non-invertible linear MPC problems having the explicit optimal solution in the form as per Assumption 3.2.

### 3.5.1 Additional Notation and Definitions

Let us now append some necessary notation, related closely to the topic treated in this section. We first state a set of definitions found to be useful and hence adopted from [10] to keep the notation compact. To make the further explanation more clear, we restrict ourselves to single-input systems (cf. Remark 3.2).

**Definition 3.4** Let  $\bar{\kappa}$  and  $\underline{\kappa}$  denote, respectively, the maximum and minimum values which the PWA function  $\kappa(x) := f_i^T x + g_i$  attains over  $\text{dom}(\kappa(x))$ . Denote by  $\mathcal{I}_{\max}$  ( $\mathcal{I}_{\min}$ ) the index set of regions where  $\kappa(x)$  is saturated at the maximum (minimum), and let  $\mathcal{I}_{\text{sat}} = \mathcal{I}_{\max} \cup \mathcal{I}_{\min}$ . We call a region  $\mathcal{R}_i$  the saturated region if it is either saturated at the minimum or at the maximum, i.e., if  $i \in \mathcal{I}_{\text{sat}}$ . Otherwise the region is called unsaturated. The index set of unsaturated regions is denoted by  $\mathcal{I}_{\text{unsat}}$ .

**Definition 3.5** Given a continuous PWA function  $\kappa(x)$ , defined over the parameter space partition  $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^R$ , we call the PWA function  $\tilde{\kappa}(x) := \tilde{f}_j^T x + \tilde{g}_j$  a suitable augmentation of  $\kappa(x)$  if the following properties hold:

- P1:  $\tilde{\kappa}(x)$  is defined over partition  $\tilde{\mathcal{R}} = \{\tilde{\mathcal{R}}_j\}_{j=1}^{\tilde{R}}$  such that  $\bigcup_{j \in \mathcal{I}_{\tilde{\mathcal{R}}}} \tilde{\mathcal{R}}_j = \bigcup_{i \in \mathcal{I}_{\mathcal{R}}} \mathcal{R}_i$ ,  
i.e.,  $\text{dom}(\tilde{\kappa}(x)) = \text{dom}(\kappa(x))$ .
- P2:  $\tilde{\kappa}(x) = \kappa(x)$ ,  $\forall x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ .

P3:  $\tilde{\kappa}(x) \geq \bar{\kappa}, \forall x \in \mathcal{R}_{\mathcal{I}_{\max}}$ .

P4:  $\tilde{\kappa}(x) \leq \underline{\kappa}, \forall x \in \mathcal{R}_{\mathcal{I}_{\min}}$ .

In addition, we introduce several polyhedral operations, which were shown to be convenient in the context of IpCP complexity analysis [12], and are utilized here to clarify the algorithmic notation.

Given a full row rank matrix  $M \in \mathbb{R}^{r \times (d+1)}$ , by  $\mathcal{P}(M)$  we denote the polyhedron

$$\mathcal{P}(M) = \{x \in \mathbb{R}^d \mid M(\cdot, 1:d)x \leq M(\cdot, d+1)\},$$

where  $M(\cdot, i)$  denotes the  $i$ th column of the matrix  $M$  and  $M(\cdot, i:j)$  represents the matrix composed by the  $i$ th– $j$ th columns of  $M$ . Conversely, given a polyhedron  $\mathcal{P}$ , by  $\mathcal{P}^{-1}(\mathcal{P})$  we denote the minimal representation (in terms of dimension) of a matrix  $M$  satisfying  $\mathcal{P} = \{x \in \mathbb{R}^d \mid M(\cdot, 1:d)x \leq M(\cdot, d+1)\}$ . Note that  $\mathcal{P}^{-1}(\mathcal{P})$  is not unique due to the following observation:  $\mathcal{P}(M) = \mathcal{P}(\alpha M)$ ,  $\alpha > 0$ .

For ease of presentation let us also define an operator for removal of redundant constraints. Given two sets of constraints corresponding to two polyhedra  $P_M, P_N$ ,  $M \in \mathcal{P}^{-1}(P_M) \subset \mathbb{R}^{r_M \times (d+1)}$ ,  $N \in \mathcal{P}^{-1}(P_N) \subset \mathbb{R}^{r_N \times (d+1)}$ , by  $\text{RmRdd}(M, N)$  we denote the set of constraints characterizing  $P_M$  which are not redundant in the representation of  $P_N$ . We remark that there exist numerous algorithms for removing redundant constraints. Herein, we recall the one presented in [16] through a mathematical description as follows:  $\text{RmRdd}(M, N) = K \in \mathbb{R}^{r_K \times (d+1)}$  s.t.

- $K$  is a sub-block of  $M$ ,
- for any  $i \in \mathcal{I}_{r_K}$ ,  $\max_{x \mid x \in P_N} K(i, 1:d)x > K(i, d+1)$ .

In addition, for  $P_M \subseteq P_N$ ,  $M \in \mathcal{P}^{-1}(P_M)$ ,  $N \in \mathcal{P}^{-1}(P_N)$ ,  $\text{RmRdd}(N, M) = \emptyset$ .

### 3.5.2 Extended Convex Liftings-Based Inverse pLP with Clipping

In this subsection we show how to recover solutions to a given class of MPC problems via another reformulated IpLP-based MPC problem with the prediction horizon equal to two prediction steps while preserving optimality.

Given a convexly non-liftable polyhedral partition/cell complex  $\mathcal{R} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_R}$ , the proposed procedure exploits the possibly non-convex collection of unsaturated regions,  $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ , which is assumed to be convexly liftable (recall Assumption A2.4 and related comments). The first step is hence to find the gains  $(a_i, b_i)$ ,  $\forall i \in \mathcal{I}_{\text{unsat}}$  of its associated convex lifting. This can be carried out efficiently by using the constructive procedure (Algorithm 3.1 in [15]) first proposed in [14], yielding  $z(x) = a_i^T x + b_i$ , with  $x \in \bigcup_{i \in \mathcal{I}_{\text{unsat}}} \mathcal{R}_i$ . Further, instead of constructing an affinely equivalent polyhedron, we build the vertex set  $V_{[x^T z u^T]^T}^{\text{unsat}}$  and compute its dual, half-space representation  $\Pi_{[x^T z u^T]^T}^{\text{unsat}} = \text{conv}(V_{[x^T z u^T]^T}^{\text{unsat}})$  as per Theorem 5.3 in [13], slightly modified for

our case. This polytope is to be used to construct a constraint set for Problem 3.1 by extending the given convex lifting and control law over the saturated regions.

As outlined in the introduction, for the further course of this study we exploit the question of minimal complexity of the inverse optimality formulation, discussed in [12]. In particular, not all the constraints—defining half-spaces of the set  $\Pi_{[x^T z u^T]^T}$  within the generic IpCP problem formulation are meaningful from the optimization point of view, in the sense that they are not active at the optimum. Therefore, it is sufficient to exclusively consider only the active constraints, which are defined by corresponding supporting hyperplanes containing certain facets of  $\Pi_{[x^T z u^T]^T}$ . We adopt this idea to preserve the active constraints which are thus not redundant in the half-space representation of  $\Pi_{[x^T z u^T]^T}^{\text{unsat}}$ . Such a relaxation naturally modifies this set in both parameter and argument space; hence it needs to be associated with an appropriate restriction in the parameter space, which bounds the domain of validity for the optimal solution. This procedure is summarized in Algorithm 3.1.

---

### Algorithm 3.1 Construction of constraint set for the extended IpCP

---

**INPUT:** Convexly non-liftable polyhedral partition/cell complex  $\mathcal{R} = \{\mathcal{R}_i\}_{i=1}^R$  due to  $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ ,

$$M_{[x^T z u^T]^T}^{\text{unsat}} = \mathcal{P}^{-1}(\Pi_{[x^T z u^T]^T}^{\text{unsat}}), \begin{bmatrix} z \\ u \end{bmatrix}_{\text{pwa}} = \arg \min_{[z u^T]^T} z \quad \text{s.t. } [x^T z u^T]^T \in \Pi_{[x^T z u^T]^T}^{\text{unsat}}$$

**OUTPUT:**  $\Pi_{\min}$ , constraint set  $\tilde{\Pi}$  associated with  $\tilde{\mathcal{R}}$  (which is to be obtained from (3.15))

- 1:  $M_{\min} = []$ .
  - 2: **for**  $i = 1 : \text{Card}(\mathcal{I}_{\text{unsat}})$
  - 3: Find the polyhedron  $P_0$  described by the active constraints at  $[x^T z u^T]^T$  for  $x \in \mathcal{R}_{\mathcal{I}_{\text{unsat}}, i}$ .
  - 4:  $M_0 = \mathcal{P}^{-1}(P_0)$ ,  $M_{\min} = \begin{bmatrix} M_0 \\ M_{\min} \end{bmatrix}$ .
  - 5: **end for**
  - 6:  $\Pi_{\min} = \mathcal{P}(M_{\min})$ ,  $\Pi_x = \text{Proj}_{\mathbb{R}^{n_x}} \Pi_{\min}$ .
  - 7:  $M_x = \mathcal{P}^{-1}(\Pi_x)$ ,  $M_f = \mathcal{P}^{-1}(\text{conv}(\mathcal{R}))$ .
  - 8:  $\overline{M}_f = \text{RmRdd}(M_f, M_x)$ .
  - 9:  $\overline{M} = [\overline{M}_f(\cdot, 1 : n_x) \ 0_{m \times (n_x + 1)} \ \overline{M}_f(\cdot, n_x + 1)]$ . Note:  $\overline{M}_f \in \mathbb{R}^{m \times (n_x + 1)}$
  - 10:  $M_{\min} = \begin{bmatrix} M_{\min} \\ \overline{M} \end{bmatrix}$ ,  $\tilde{\Pi} = \mathcal{P}(M_{\min})$ .
- 

*Remark 3.1* Note that the explicit solution  $[z u^T]^T$  required as input for Algorithm 3.1 is known a priori with respect to a given continuous PWA function and constructed convex lifting associated with  $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ , without the need to solve the minimization problem. Step 8 is present to avoid the redundancy phenomena while adding supplementary constraints given by  $\overline{M}_f$ . In addition, we remark that for the MPC control laws designed with the feasible set  $\Omega = \text{conv}(\mathcal{R})$  to be positively invariant, these constraints (Steps 6-9) can be further omitted as long as the initial state  $x_0 \in \Omega$ . This may reduce the complexity of the formulation in implicit MPC implementations.

This algorithmic procedure first retrieves the minimal half-space representation of the set  $\Pi_{[x^T z u^T]^T}^{\text{unsat}}$ , potentially useful for the IpCP problem formulation, and given by an unbounded polyhedron  $\Pi_{\min}$  (Steps 1–6). Feasibility of this set is further provided

through a restriction imposed on its parameter subspace (Step 7). Additional redundant constraints that may emerge at this point are effectively removed via Step 8. The resulting constraint set is thus given by  $\tilde{\Pi}$  (Step 10). Now, recall that, geometrically, each active constraint corresponds to a hyperplane of dimension  $n_x + n_u$ . Therefore, we are interested in the constraints corresponding to the supporting hyperplanes of  $\tilde{\Pi}$  which contain the  $n_x$ -faces whose orthogonal projection onto  $\mathbb{R}^{n_x}$  retrieves the parameter space partition  $\mathcal{R}$  associated with the optimal PWA function  $\kappa(x)$ . This is indeed not quite the case here, since  $\text{Proj}_{\mathbb{R}^{n_x+n_u}} \tilde{\Pi}$  (or  $\text{Proj}_{\mathbb{R}^{n_x+n_u}} \tilde{\Pi}$ —affinely equivalent polyhedron to  $\tilde{\mathcal{R}}$ ) is formed as a convex extension of  $\text{Proj}_{\mathbb{R}^{n_x+n_u}} \prod_{[x^T \ z \ u^T]^T}^{\text{unsat}}$  ( $\text{Proj}_{\mathbb{R}^{n_x+n_u}} \prod_{[x^T \ z \ u^T]^T}^{\text{unsat}}$ , respectively) over  $\bigcup_{i \in \mathcal{I}_{\text{sat}}} \mathcal{R}_i$ , bounded by the restriction in the parameter space  $\mathbb{R}^{n_x}$  imposed by the feasible set  $\Omega = \text{dom}(\kappa(x))$ . It follows that the explicit optimal solution ( $\tilde{\kappa}(x)$ ,  $\tilde{\mathcal{R}}$ ) to thusly cast “inverse” problem is not equivalent anymore, in the sense that  $\tilde{\kappa}(x)$  does not coincide with the original PWA feedback  $\kappa(x)$ , namely in its saturated portions defined over  $\mathcal{R}_{\mathcal{I}_{\text{sat}}}$ . The cell complex  $\tilde{\mathcal{R}}$  in this way inherits a collection of regions with similar (or equivalent, if  $\bigcup_{i \in \mathcal{I}_{\text{unsat}}} \mathcal{R}_i$  is convex) partitioning as  $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ .

The following theorem summarizes the minimal formulation of an extended IpLP problem resulting directly from Algorithm 3.1.

**Theorem 3.3** *The image via orthogonal projection onto  $\mathbb{R}^{n_u}$  of the optimal solution to the following optimization problem*

$$\min_{[z \ u^T]^T} z \quad \text{s.t.} \quad [x^T \ z \ u^T]^T \in \tilde{\Pi} \quad (3.15)$$

where  $\tilde{\Pi}$  is obtained from Algorithm 3.1, is a suitable augmentation of the given continuous PWA function  $\kappa(x) : \Omega \rightarrow \mathbb{R}^{n_u}$ , denoted as  $\tilde{\kappa}(x)$  and associated with a convexly liftable cell complex  $\tilde{\mathcal{R}} = \{\tilde{\mathcal{R}}_j\}_{j \in \mathcal{I}_{\tilde{\mathcal{R}}}}$ .

*Proof* Follows directly from Algorithm 3.1 and from Definition 3.5 of  $\tilde{\kappa}(x)$ .  $\square$

As implied by Theorem 3.3, solution of LP (3.15) renders a suitable augmentation  $\tilde{\kappa}(x)$  of the original piecewise affine function  $\kappa(x)$ . The augmented function, however, cannot be readily applied as an explicit receding horizon MPC feedback since, in general,  $\tilde{\kappa}(x) \neq \kappa(x)$  for  $x \in \mathcal{R}_{\mathcal{I}_{\text{sat}}}$ . Therefore, to achieve the equivalence of the two, we adopt the concept of a simple clipping filter proposed in [10], and recalled here as follows:

**Theorem 3.4** ([10]) *Consider a saturated continuous PWA function  $\kappa(x)$  and its suitable augmentation  $\tilde{\kappa}(x)$  obtained per Theorem 3.3. Let*

$$\phi(\tilde{\kappa}(x)) := \max \{ \underline{\kappa}, \min \{ \tilde{\kappa}(x), \bar{\kappa} \} \} \quad (3.16)$$

*Then the equivalence  $\phi(\tilde{\kappa}(x)) = \kappa(x)$  is established for all  $x \in \text{dom}(\kappa(x))$ .*

*Proof* Straightforward with respect to Definition 3.5; can be found in [10].  $\square$

*Remark 3.2* As outlined in Sect. 3.5.1, the above derived procedure is clearly valid for single-input systems. However, the dimension of  $\kappa(x)$  in Problem 3.1 is not necessarily equal to 1, in fact, it can be extended for the case when  $\kappa(x)$  is a vector-valued function with  $n_u > 1$ . In that case, Definitions 3.4, 3.5, and Theorem 3.4 need to be slightly modified in an adequate manner to account for this fact; for more details see [10].

Finally, in Algorithm 3.2 we summarize the proposed constructive IpLP-based procedure towards recovering a suitable augmentation of a given continuous piecewise affine function, which when passed through a clipping filter represents the solution to Problem 3.1.

---

### Algorithm 3.2 Extended IpLP with clipping

---

**INPUT:** Saturated continuous PWA function  $\kappa(x)$  defined over a, possibly, convexly non-liftable polyhedral partition/cell complex  $\mathcal{R} = \{\mathcal{R}_i\}_{i \in \mathcal{I}_R}$  of a polytope  $\Omega \subset \mathbb{R}^{n_x}$ .

**OUTPUT:** Suitable augmentation  $\tilde{\kappa}(x)$  and equivalent replacement  $\phi(\tilde{\kappa}(x))$ , defined over a convexly liftable cell complex  $\tilde{\mathcal{R}} = \{\tilde{\mathcal{R}}_j\}_{j \in \mathcal{I}_{\tilde{R}}}$ .

- 1: Obtain the index set  $\mathcal{I}_{\text{unsat}}$  containing indices of unsaturated regions.
- 2: Compute the gains of convex lifting for the possibly non-convex, yet convexly liftable, collection of unsaturated regions  $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$  via Algorithm 3.1 (Steps 1 to 3) in [15].
- 3: Construct the set  $\Pi_{[x^T z u^T]^T}^{\text{unsat}}$  defined over  $\mathcal{R}_{\mathcal{I}_{\text{unsat}}}$ , as stated in Theorem 5.3 in [13].
- 4: Form a new constraint set  $\tilde{\Pi}$  associated with  $\tilde{\mathcal{R}}$ , as described in Algorithm 3.1.
- 5: Formulate and solve an extended IpLP problem with constraints on  $x, z, u$  given by the polytope  $\tilde{\Pi}$  to obtain:

$$\begin{bmatrix} \tilde{z}^* \\ \tilde{u}^* \end{bmatrix} = \arg \min_{[z u^T]^T} z \quad \text{s.t. } [x^T z u^T]^T \in \tilde{\Pi}.$$

- 6: Extract the appropriate sub-component of the above optimal vector:

$$\tilde{u}^* = \text{Proj}_{\mathbb{R}^{n_u}} \begin{bmatrix} \tilde{z}^* \\ \tilde{u}^* \end{bmatrix} = \tilde{\kappa}(x)$$

to obtain a suitable augmentation  $\tilde{\kappa}(x)$  of the given PWA function  $\kappa(x)$ , defined over a rearranged, convexly liftable cell complex  $\tilde{\mathcal{R}}$ .

- 7: Design a clipping filter  $\phi(\cdot)$  as per Theorem 3.4 to maintain equivalence between  $\tilde{\kappa}(x)$  and  $\kappa(x)$ .
- 

## 3.6 A Case Study with Implication in Low-Complexity eMPC

The main aim of this section is to illustrate the proposed constructive procedure via a case study MPC problem. It is shown that by executing all the steps of Algorithm 3.2, i.e., formulating and solving an extended IpLP problem with clipping, we obtain a performance-lossless replacement of the receding horizon MPC feedback associated with the original control problem. In addition, we perform an analysis to assess the computational complexity of the scheme as well as the structural complexity of the recovered eMPC controller.



The example is taken from an active vibration control task [21], where the objective is to attenuate vibrations of a flexible cantilever beam, described by the following experimentally identified state-space model:

$$\begin{aligned} x(k+1) &= \begin{bmatrix} 0.867 & 1.119 \\ -0.214 & 0.870 \end{bmatrix} x(k) + \begin{bmatrix} 9.336\text{E-}4 \\ 5.309\text{E-}4 \end{bmatrix} u(k) \\ y(k) &= \begin{bmatrix} -0.553 & -0.705 \end{bmatrix} x(k) \end{aligned} \quad (3.17)$$

where the output of the model is the beam tip deflection in mm, and the input is the direct actuator voltage in V. Control inputs required to steer the system (3.17) into equilibrium can be computed by solving the following optimization problem:

$$\begin{aligned} \min_{u_0, \dots, u_{N-1}} \quad & x_N^T P x_N + \sum_{k=0}^{N-1} [x_k^T Q x_k + R u_k^2] \\ \text{s.t.} \quad & x_0 = x(k) \\ & x_{k+1} = A x_k + B u_k, \quad k = 0, \dots, N-1 \\ & u_k \in \mathbb{U}, \quad k = 0, \dots, N-1 \\ & x_N \in \mathbb{X}_f \end{aligned} \quad (3.18)$$

subject to constraints  $|u_k| \leq 120$  V. The MPC problem (3.18) was formulated with  $Q = C^T C$ ,  $R = 1\text{E-}4$ , and  $P$  and  $\mathbb{X}_f$  designed such that closed-loop stability is guaranteed, i.e., by setting  $P$  to the solution of DARE and using a positive control invariant terminal set  $\mathbb{X}_f$ . Next, it was recast and solved as a pQP. The explicit MPC feedback  $\kappa(x)$  for different horizon lengths  $N$  was obtained using MPT Toolbox [8]. Each resulting PWA solution  $u_0^*(x) = \kappa(x)$  was subsequently post-processed using the extended IpLP procedure described by Algorithm 3.2.

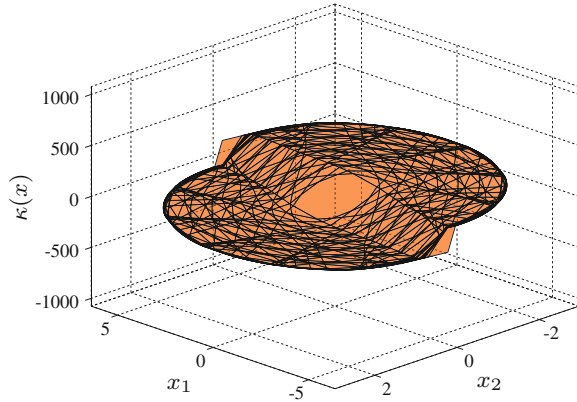
Table 3.1 reports the results obtained by the proposed extended IpLP procedure, marked by (·b), compared to the standard explicit solution via parametric quadratic programming, denoted as (·a), used for the solution of (3.18) for different lengths of prediction horizon  $N \in \{10, 20, 30, 40, 50\}$ . The complexity of the IpLP-based scheme is expressed in terms of the number of regions of the resulting eMPC controller, and the number of constraints (half-spaces defining the constraint set  $\tilde{\Pi}$ ) in LP (3.15). As an example, the standard MPC problem needs 90 constraints with the horizon of length 40 to obtain the PWA control law shown in Fig. 3.1, while the formulation of IpLP with clipping requires 641 constraints (551 non-redundant “general” constraints and additional 90 constraints to conserve the solution’s structure) and exactly two prediction steps to yield the same result, see Fig. 3.2. Moreover, one may also notice the complexity reduction in the number of polytopic regions of the underlying state-space partition. In this particular case, the recovered partition consists of 814 regions instead of 3397 regions of the original partition, which corresponds to the reduction by a factor of more than four.

The total memory footprint of the original function  $\kappa$  for the investigated scenario with 3397 regions is 408 kB. The recovered function  $\tilde{\kappa}$ , on the other hand, requires

**Table 3.1** Selected data comparing equivalent (in terms of the resulting feedback law) formulations of the case study MPC problem for different lengths of prediction horizon  $N$ 

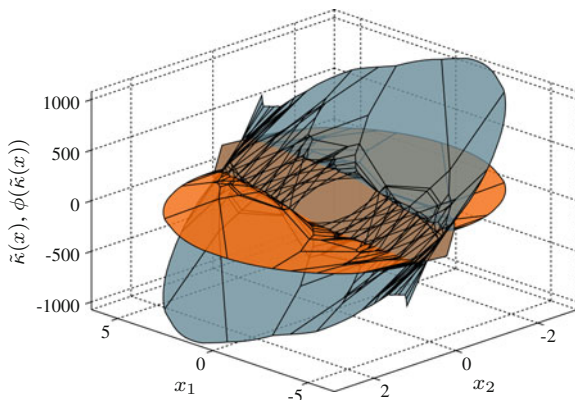
Formulation	# of regions				# of constraints <sup>†</sup>		
	$N$	$R$	$R_{\text{unsat}}$	$\tilde{R}$		$\aleph(\Pi)$	$\aleph(\mathcal{A})$
(1a) Standard MPC problem	10	257	75	–	30	–	–
(1b) Extended IpLP w. clipping	2	–	–	127	–	118	30
(2a)	20	913	159	–	50	–	–
(2b)	2	–	–	320	–	232	50
(3a)	30	1955	259	–	70	–	–
(3b)	2	–	–	557	–	384	70
(4a)	40	3397	367	–	90	–	–
(4b)	2	–	–	811	–	551	90
(5a)	50	5231	367	–	110	–	–
(5b)	2	–	–	1063	–	774	110

<sup>†</sup> $\aleph(\cdot)$  denotes the number of half-spaces defining a polyhedral set. For the case of formulation (-b),  $\Pi = \Pi_{\min}$  and  $\mathcal{A} = \mathcal{P}(\bar{M}_f)$  (see Algorithm 3.1)

**Fig. 3.1** Surface plot of the explicit MPC control law we aim to recover (3397 regions,  $N = 40$ )

105 kB, with additional 16 bytes to store the clipping filter  $\phi(\cdot)$ . This indicates reduction of memory consumption by a factor of 3.8. The worst-case computational effort needed to evaluate  $\kappa$  is 54332 FLOPs, which can be by employing  $\tilde{\kappa}$  reduced, adequately, to 14278 FLOPs (14272 FLOPs to perform point location by sequential search and 6 FLOPs to evaluate  $\phi(\tilde{\kappa}(x))$ ).

To review the complexity also from another practical point of view, Table 3.2 reports the CPU times spent on main algorithmic routines of the proposed extended IpCP approach and its generic pCP counterpart. The total time of the former may be split among four consecutive parts, in fact, represented by Steps 2 to 5 of Algorithm 3.2. Clearly, the last two steps represent the computationally most demanding procedures, which also scale more significantly as  $N$  increases. However, when



**Fig. 3.2** Surface plot of the recovered eMPC controller (811 regions,  $N = 2$ ). Result of the clipping procedure is shown in *orange colour* (illustration only)

**Table 3.2** Remark on computation times of the main algorithmic features for the respective formulation of the case study MPC example

Formulation	Task execution time (s) <sup>†</sup>				
	$N = 10$	$N = 20$	$N = 30$	$N = 40$	$N = 50$
Standard MPC problem <i>pQP</i>	6.27	24.21	91.62	171.87	356.19
Extended IpLP with clipping <i>convex lifting</i> (Algorithm 3.2, step 2)	0.11	0.28	0.50	0.62	1.75
<i>Facet enumeration</i> (Algorithm 3.2, step 3)	0.05	0.07	0.12	0.17	0.23
<i>Constraint set</i> (Algorithm 3.2, step 4)	2.83	14.98	38.31	74.29	135.31
<i>pLP</i> (Algorithm 3.2, step 5)	1.67	7.33	24.46	90.81	150.48

<sup>†</sup>All the computations were performed on a 2.2 GHz Core i5 CPU with 4GB of RAM

summed up, the algorithmic features of extended IpCP, including the core pLP itself, are still shown to require less offline time than the original explicit solution (pQP).

*Remark 3.3* Note that the two approaches may only be compared in terms of memory consumed by storage of the polyhedral partition along with the associated PWA function, or eventually the time required for online evaluation of the controller—both reduced here w.r.t. the data in Table 3.1. However, this is not the case for the offline pre-processing time spent on solving the respective pCPs since the entire IpCP procedure assumes the existence of the original explicit controller. Yet, such a study can shed more light on computational effort of the proposed approach with respect to the generic one, and in this way allow to analyse its viability.

Finally, we present a brief illustration of the aforementioned results. In particular, Fig. 3.1 shows the explicit optimal solution to the receding horizon MPC problem (3.18) for  $N = 40$ . The optimal feedback takes form of a continuous PWA function  $\kappa(x)$  defined over a state-space partition  $\mathcal{R}$ . This cell complex is not convexly liftable due to the hyperplane arrangement of the two large collections of

polytopes representing the controller regions over which  $\kappa(x)$  is saturated at  $\bar{\kappa}$  or  $\underline{\kappa}$ . This solution thus presents a candidate for the extended IpLP scheme. Following the algorithmic procedure presented in Sect. 3.5.1 we obtain the solution of an extended inverse MPC problem with  $N = 2$ , which is the PWA function  $\tilde{\kappa}(x)$  depicted in Fig. 3.2. It is defined over a similar recovered partition  $\tilde{\mathcal{R}}$ , which is a convexly liftable cell complex. The optimality of  $\tilde{\kappa}(x)$  over its entire domain of validity is achieved by passing it through a simple clipping filter  $\phi(\cdot)$ .

The resulting low-complexity feedback law can be readily and conveniently used within online/offline explicit MPC implementations while preserving all the performance, closed-loop stability and feasibility properties. We note that the proposed approach may be analogously exploited in implicit MPC schemes, at each sampling instant solving the equivalent “horizon 2” problem with the constraint set  $\Pi_{\min}$  (cf. Remark 3.1); hence revealing the full potential efficiency of the convex liftings-based inverse optimality approach in practical model predictive control applications.

### 3.7 Conclusion

The chapter presents an approach exploiting the inverse parametric convex programming based upon the concept of convex lifting. As shown recently, for any continuous PWA function defined over a polyhedral partition, one may find an appropriate equivalent function by solving another parametric linear/quadratic programming problem with a supplementary variable of dimension equal to 1. Applying this idea to the model predictive control framework allowed to further state that any linear MPC formulation has an equivalent inverse reformulation with two steps of the prediction horizon. This study, in particular, deals with the invertibility of the PWA functions resulting from pQP optimization problems, which is in fact seldom guaranteed due to its structural properties. Despite being already proven that this issue can always be treated by a suitable sub-partitioning of the corresponding parameter space partition, herein the emphasis has been put mainly on devising a technique which would not increase the controller’s explicit representation in terms of the polytopic regions. On the contrary, an algorithmic extension to the inverse parametric linear programming is presented, which in addition to the primary objective—to allow to target a class of common practical problems yielding convexly non-liftable cell complexes containing saturated regions, emerged to be capable of producing a lesser number of regions. This implies an expected applicability of the proposed scheme within implementations of explicit MPC, as compared to the generic convex liftings-based IpCP. It is evidenced by the presented numerical analysis which suggests that the inverse optimality concept applied to a complex but computable eMPC may lead to a lower complexity explicit control law. Subject of ongoing research remains the investigation of complexity bounds for the inverse optimality formulation. Apart from a valuable theoretic insight, the clarification of this topic shall yet improve computational tools necessary to fully exploit the efficacy of this control design approach.

**Acknowledgments** M. Gulan and B. Rohal-Ilkiv gratefully acknowledge the contribution of the Slovak Research and Development Agency under the project APVV-14-0399.

## References

1. A. Alessio, A. Bemporad, A survey on explicit model predictive control, in *Nonlinear Model Predictive Control—Towards New Challenging Applications*, Lecture Notes in Control and Information Sciences, ed. by L. Magni, D.M. Raimondo, F. Allgöwer (Springer, Berlin, 2009), pp. 345–369
2. M. Baes, M. Diehl, I. Necoara, Every continuous nonlinear control system can be obtained by parametric convex programming. *IEEE Trans. Autom. Control* **53**(8), 1963–1967 (2008)
3. A. Bemporad, F. Borelli, M. Morari, Model predictive control based on linear programming—the explicit solution. *IEEE Trans. Autom. Control* **47**(12), 1974–1985 (2002)
4. A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems. *Automatica* **38**(1), 3–20 (2002)
5. F. Borelli, A. Bemporad, M. Morari, The explicit linear quadratic regulator for constrained systems. *J. Optim. Theory Appl.* **118**(3), 515–540 (2003)
6. B. Grünbaum, *Convex Polytopes* (Wiley, New York, 1967)
7. A.B. Hempel, P.J. Goulart, J. Lygeros, Every continuous piecewise affine function can be obtained by solving a parametric linear program, in *13th European Control Conference* (Zürich, Switzerland, 2013), pp. 2657–2662
8. M. Herceg, M. Kvasnica, C.N. Jones, M. Morari, Multi-parametric toolbox 3.0, in *13th European Control Conference* (Zürich, Switzerland, 2013), pp. 502–510
9. J. Holaza, B. Takács, M. Kvasnica, S. Di Cairano, Nearly optimal simple explicit MPC controllers with stability and feasibility guarantees, in *Optimal Control Applications and Methods*. Available online (2014)
10. M. Kvasnica, M. Fikar, Clipping-based complexity reduction in explicit MPC. *IEEE Trans. Autom. Control* **57**(7), 1878–1883 (2012)
11. M. Kvasnica, J. Hledík, I. Rauová, M. Fikar, Complexity reduction of explicit model predictive control via separation. *Automatica* **49**(6), 1776–1781 (2013)
12. N.A. Nguyen, S. Oлару, P. Rodriguez-Ayerbe, On the complexity of the convex liftings-based solution to convex programming problems, in *15th European Control Conference* (Linz, Austria, 2015), pp. 3433–3438
13. N.A. Nguyen, S. Oлару, P. Rodriguez-Ayerbe, M. Hovd, I. Necoara, Inverse parametric convex programming problems via convex liftings, in *19th IFAC World Congress* (Cape Town, South Africa, 2014), pp. 2490–2494
14. N.A. Nguyen, S. Oлару, P. Rodriguez-Ayerbe, M. Hovd, I. Necoara, On the lifting problems and their connections with piecewise affine control design, in *14th European Control Conference* (Strasbourg, France, 2014), pp. 2164–2169
15. N.A. Nguyen, S. Oлару, P. Rodriguez-Ayerbe, M. Hovd, I. Necoara, Constructive solution of inverse parametric linear/quadratic programming problems, *Developments in Model-Based Optimization and Control* (Springer, 2016)
16. S. Oлару, D. Dumur, Avoiding constraints redundancy in predictive control optimization routines. *IEEE Trans. Autom. Control* **50**(9), 1459–1465 (2005)
17. S. Oлару, D. Dumur, On the continuity and complexity of control laws based on multiparametric linear programs, in *45th Conference on Decision and Control* (San Diego, USA, 2006), pp. 5465–5470
18. A. Schrijver, *Theory of Linear and Integer Programming* (Wiley, New York, 1998)
19. M.M. Seron, G.C. Goodwin, J.A. De Doná, Characterisation of receding horizon control for constrained linear systems. *Asian J. Control* **5**(2), 271–286 (2003)

20. J. Spjøtvold, E.C. Kerrigan, C.N. Jones, P. Tøndel, T.A. Johansen, On the facet-to-facet property of solutions to convex parametric quadratic programs. *Automatica* **42**(12), 2209–2214 (2006)
21. G. Takács, B. Rohal-Ilkiv, *Model Predictive Vibration Control: Efficient Constrained MPC Vibration Control for Lightly Damped Mechanical Structures* (Springer, London, 2012)
22. P. Tøndel, T.A. Johansen, A. Bemporad, An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* **39**(3), 489–497 (2003)