

Parametric Analysis of Controllers for Constrained Linear Systems

Miroslav Barić, Colin Jones, Manfred Morari*

Abstract—We analyze properties of closed-loop systems based on explicit model predictive control (MPC) when parameters of the controllers are changing. Formulation of the problem in the framework of MPC with a cost based on piecewise linear norms leads to the generalized multi-parametric linear program containing parameters both in the cost and in the constraints. The focus of the paper is on describing a novel simplex-based algorithm for solving such a class of problems. The algorithm uses the concept of lexicographic perturbation to resolve problems caused by degeneracy.

I. INTRODUCTION

Recently introduced concept of the explicit solution extended the application of the MPC paradigm to faster processes [1]. In such an approach, the MPC problem is formulated as a *multi-parametric* convex optimization problem, where the states of the system are treated as parameters, and control inputs as optimization variables. The solution to the multi-parametric program gives a closed-form expression for the optimal control input as a function of the state variables. For piecewise linear (norms 1 and ∞) and quadratic costs, i.e. for multi-parametric linear programs (mpLPs) and multi-parametric quadratic programs (mpQPs), the optimal control input is defined as an affine state feedback law over a polyhedral partition of the feasible state space. By pre-solving the optimization problem using parametric programming, the computational burden is moved off-line, while the on-line procedure reduces to simple calculation of the affine control law.

In this framework parametric programming has reached its renaissance as the major computational tool. A number of different algorithms for solving parametric linear and quadratic programs have been developed ([2], [3], [4]). The application in the area of control motivated the search for algorithms guaranteeing uniqueness and continuity of the explicit control law ([5], [6]). In [6] the author provides an elegant algorithm for the solution of multi-parametric linear programs with parameters either in the cost or in the constraints (but not in both). The algorithm uses the concept of lexicographic perturbations ([7]) to guarantee both uniqueness and continuity of the control input in the presence of degeneracy.

Application of parametric programming in control has focused almost exclusively on the computation of explicit controllers. A different application was presented in [8], where authors described a method for analyzing the stability of a class of explicit controllers when parameters of the

controller are changing. The proposed analysis procedure requires the solution to so-called *RIM mpLPs* [9], i.e. multi-parametric linear programs with parameters both in the cost and in the constraints. However, the authors did not provide an algorithm which resolves the problem of degeneracy efficiently.

This paper contains an extension of the work presented in [8]. We present a novel, efficient algorithm for solving *RIM mpLPs*, which guarantees the correctness of the solution. The algorithm is based on the idea of lexicographic perturbation and uses a series of primal and dual simplex iterations to exhaustively enumerate optimal basic solutions of the mpLP.

II. NOTATION

Given an *ordered* set $\mathcal{B} \subseteq \mathbb{N}$, \mathcal{B}_i represents the i -th element of the set. Given a vector \mathbf{c} and a set of indices \mathcal{B} , $\mathbf{c}_{\mathcal{B}}$ is a vector consisting of the elements of \mathbf{c} indexed by \mathcal{B} . Similarly, given a matrix \mathbf{A} , $\mathbf{A}_{\mathcal{B}}$ denotes a submatrix consisting of columns of \mathbf{A} indexed by the set \mathcal{B} . If $\mathcal{B} = \{i\}$ is a singleton, we will write $\mathbf{A}_{.i}$ for $\mathbf{A}_{\{i\}}$. The j^{th} row of a matrix \mathbf{A} is denoted as \mathbf{A}_j . Superscript indices are used for enumeration: the matrix \mathbf{A}^i is the i -th among several matrices and the set \mathcal{B}^i is the i -th among several sets etc.

A *polyhedron* \mathcal{P} is an intersection of finitely many half-spaces: $\mathcal{P} \triangleq \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{H}\mathbf{x} \leq \mathbf{k}\}$. The affine hull of a polyhedron \mathcal{P} is the intersection of all affine combinations of points $\mathbf{x} \in \mathcal{P}$ and is denoted $\text{aff}(\mathcal{P})$. The dimension of a polytope, $\dim(\mathcal{P})$, is dimension of its affine hull, i.e. $\dim(\mathcal{P}) \triangleq \dim(\text{aff}(\mathcal{P}))$. Polyhedron \mathcal{P} is called *full dimensional* if $\dim(\mathcal{P}) = n$, otherwise, it is called *lower dimensional*. Given a halfspace $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}^T \mathbf{x} \leq k\}$, a *face* of a polyhedron $\mathcal{P} \subseteq \mathcal{H}$ is any set of the form $\mathcal{P} \cap \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{h}^T \mathbf{x} = k\}$. Faces whose dimension is $n - 1$ are called *facets*.

A point \mathbf{x}_0 is called an *interior point* of a polyhedron $\mathcal{P} = \{\mathbf{x} \mid \mathbf{H}\mathbf{x} \leq \mathbf{k}\}$ if $\mathbf{H}\mathbf{x}_0 < \mathbf{k}$. The set of all interior points of \mathcal{P} is denoted $\text{int}(\mathcal{P})$. The relative interior of a polyhedron \mathcal{P} is defined as $\text{relint}(\mathcal{P}) \triangleq \{\mathbf{x} \in \mathcal{P} \mid \text{for any } \mathbf{x}_1 \in \mathcal{P}, \mathbf{x}_1 \neq \mathbf{x} \exists \mathbf{x}_2 \in \mathcal{P} \mid \mathbf{x} = \lambda \mathbf{x}_1 + (1 - \lambda) \mathbf{x}_2, 0 < \lambda < 1\}$.

III. MOTIVATION

Consider discrete-time linear time-invariant system:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \quad (1)$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$, $\mathbf{B} \in \mathbb{R}^{n \times m}$. The system (1) is subject to constraints:

$$\mathbf{P}_x \mathbf{x}_k + \mathbf{P}_u \mathbf{u}_k \leq \mathbf{p}_c \quad (2)$$

*Automatic Control Laboratory, ETH Zentrum, ETL I 22, CH-8092 Zürich, Switzerland, phone: +41 1 632 61 45, fax: +41 1 632 12 11
 baric | baotic | morari@control.ee.ethz.ch

for all time instances $k \geq 0$. Define the following cost function:

$$J(\mathbf{U}_0^{N-1}, \mathbf{x}_0) \triangleq \|\mathbf{P}_N \mathbf{x}_N\|_\ell + \sum_{k=0}^{N-1} \|\mathbf{Q} \mathbf{x}_k\|_\ell + \|\mathbf{R} \mathbf{u}_k\|_\ell, \quad (3)$$

where N is a prediction horizon, \mathbf{P}_N is a matrix defining the weight on the terminal state \mathbf{x}_N , $\|\cdot\|_\ell$ with $\ell \in \{1, \infty\}$ denotes the vector norm and $\mathbf{U}_0^{N-1} = [\mathbf{u}_0^T, \dots, \mathbf{u}_{N-1}^T]^T \in \mathbb{R}^{m \cdot N}$ is the vector of control moves over the time horizon. Constrained finite time optimal control (CFTOC) requires the solution to the following problem:

$$J^*(\mathbf{x}_0) := \min_{\mathbf{U}_0^{N-1}} J(\mathbf{U}_0^{N-1}, \mathbf{x}_0), \quad (4)$$

$$\text{s. t. } \begin{cases} \mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{B} \mathbf{u}_k, \\ \mathbf{P}_x \mathbf{x}_k + \mathbf{P}_u \mathbf{u}_k \leq \mathbf{p}_c, \\ \mathbf{x}_N \in \mathcal{T}_{set}, \end{cases} \quad (5)$$

where \mathcal{T}_{set} is a *terminal set*, i.e. the set of admissible states at the final time instance $k = N$.

As shown in [10], the CFTOC problem (4)- (5) can be formulated and solved as a linear program, for $\ell \in \{1, \infty\}$. Moreover, it can be solved *explicitly* as a multi-parametric linear program (mpLP), where an initial state vector \mathbf{x}_0 is considered as a parameter. In [8] we extended this concept by including the parameters from the cost function (matrices \mathbf{Q} , \mathbf{R} , \mathbf{P}_N) into the vector of parameters of the mpLP. The extended problem was formulated as a *RIM* multi-parametric linear program, i.e. parametric linear program containing parameters *both in the cost and in the constraints*. The solution to this type of problems enables the direct analysis of the properties of the controller with respect to the parameters of the cost function (3). The computational scheme used in [8] for solving the *RIM* mpLP is derived from the simple “geometric” algorithm for solving parametric quadratic programs described in [3]. This computational procedure may be very inefficient for highly degenerate problems, due to the extensive use of projection as the mean to obtain the unique solution. Additionally, the algorithm cannot guarantee the completeness of a solution, which has to be verified *a posteriori*. Therefore, in this paper we focus on the development of a more efficient and reliable algorithm for solving *RIM* mpLPs.

IV. PRELIMINARIES

Before presenting the algorithm for solving *RIM* mpLPs we will give some preliminaries on linear programs (LPs) in general. The focus will be on the phenomenon of *degeneracy* in LPs and on an algorithm for solving degenerate LPs. As will be shown further in the text, a systematic procedure for resolving degeneracy in LPs is crucial for the development of the *RIM* mpLP algorithm.

A. Linear Programming

Consider the following primal-dual pair of LPs:

$$\min_{\mathbf{x}} \mathbf{c}^T \mathbf{x}, \quad \text{s. t. } \mathbf{A} \mathbf{x} = \mathbf{b}, \quad \mathbf{x} \geq 0 \quad (6)$$

$$\max_{\boldsymbol{\pi}} \mathbf{b}^T \boldsymbol{\pi} \quad \text{s. t. } \mathbf{A}^T \boldsymbol{\pi} \leq \mathbf{c}, \quad (7)$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\text{rank}(\mathbf{A}) = m$. We will refer to LP (6) as the *primal* and to LP (7) as the *dual*. A crucial concept in the theory of linear programming is the concept of a *basic solution*. A *basis* is defined as set of indices $\mathcal{B} \subset \{1, \dots, n\}$, $|\mathcal{B}| = m$ such that $\text{rank}(\mathbf{A}_{\mathcal{B}}) = m$. The basis \mathcal{B} and a complementary set of *non-basic indices* $\mathcal{N} = \{1, \dots, n\} \setminus \mathcal{B}$ define a partition of \mathbf{x} into *basic variables* $\mathbf{x}_{\mathcal{B}}$ and *non-basic variables* $\mathbf{x}_{\mathcal{N}}$. A *basic solution* is a solution to the system of equations in (6) with $\mathbf{x}_{\mathcal{N}}$ restricted to zero. A basic solution for which $\mathbf{x}_{\mathcal{B}} \geq 0$ is called a *basic feasible solution* (BFS) and the corresponding basis \mathcal{B} is referred to as a *primal feasible basis*. In the remainder of the text the following notation will be used:

$$\bar{\mathbf{A}} \triangleq \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{A}, \quad \bar{\mathbf{b}} \triangleq \mathbf{A}_{\mathcal{B}}^{-1} \mathbf{b}, \quad (8)$$

$$\bar{\mathbf{c}} \triangleq \mathbf{c} - \bar{\mathbf{A}}^T \mathbf{c}_{\mathcal{B}} \quad (9)$$

Expression (8) denotes the *updated* matrix \mathbf{A} and the basic solution $\mathbf{x}_{\mathcal{B}} = \bar{\mathbf{b}}$, while (9) defines the vector of *reduced costs* $\bar{\mathbf{c}}$. A basic dual solution is given by the equation $\mathbf{A}_{\mathcal{B}}^T \boldsymbol{\pi} = \mathbf{c}_{\mathcal{B}}$. If a basic dual solution satisfies constraints in (7) (dual feasible), then a basis \mathcal{B} is called *dual feasible basis*. It can be easily verified that a basis \mathcal{B} is dual feasible if and only if $\bar{\mathbf{c}} \geq 0$.

If there are vectors \mathbf{x} and $\boldsymbol{\pi}$ satisfying constraints in (6) and (7) respectively, then there exists an *optimal* primal-dual pair $(\mathbf{x}^*, \boldsymbol{\pi}^*)$ as a solution to (6)- (7). A basis \mathcal{B}^* defining an optimal solution is called the *optimal basis*. The following standard result provides the optimality condition for LP (6):

Theorem 4.1 (LP Optimality): A primal feasible basis \mathcal{B} is the optimal basis if and only if it is dual feasible.

B. Simplex Algorithm

A basic algorithm for solving an LP is the *simplex algorithm* [7]. The simplex algorithm is also the core of the proposed computational scheme for solving *RIM* mpLPs and as such will be outlined here.

The simplex algorithm starts with any primal or dual feasible basis and performs *pivot operations*, until the basis becomes both primal and dual feasible (i.e. optimal), or until unboundedness is detected. In each of the pivot steps a variable leaves the basis (the *leaving* variable) and one enters the basis (the *entering* variable). These variables are selected such that the initial primal or dual feasibility of the basis is preserved. If the initial basis is primal feasible, *primal pivots* are performed, preserving the primal feasibility. Similarly, if the algorithm starts with a dual feasible basis, a series of *dual pivots* are executed, maintaining the initial dual feasibility. The selection of entering and leaving variables

Algorithm IV.1 Primal Simplex Pivot

- 1: choose index e of the entering variable such that $\bar{c}_e < 0$
 - 2: choose the leaving variable l as:

$$l = \underset{j \in \{1, \dots, m\}}{\text{argmin}} \{ \bar{b}_j / \bar{A}_{je} \mid \bar{A}_{je} > 0 \}$$
 - 3: update the basis: $\mathcal{B} \leftarrow \mathcal{B} \cup \{e\} \setminus \{l\}$
-

for primal and dual simplex pivots is given by Algorithms IV-B-IV-B (checking of optimality and feasibility conditions are omitted).

Algorithm IV.2 Dual Simplex Pivot

- 1: choose index l of the leaving variable such that $\bar{b}_l < 0$
 - 2: choose index of the entering variable e as:

$$e = \operatorname{argmin}_{j \in \mathcal{N}} \{ \bar{c}_j / (-\bar{A}_{lj}) \mid \bar{A}_{lj} < 0 \}$$
 - 3: update the basis: $\mathcal{B} \leftarrow \mathcal{B} \cup \{e\} \setminus \{\mathcal{B}_l\}$
-

The case when the optimal basis \mathcal{B}^* or the intermediate bases are not unique is described by the notion of *degeneracy*.

Definition 4.1 (Primal degeneracy): Basis \mathcal{B} for the problem (6) is called *primal degenerate* if the number of nonzero elements in the vector $\bar{\mathbf{b}}$ is less than m .

Definition 4.2 (Dual degeneracy): Basis \mathcal{B} is called *dual degenerate* for the problem (6) if the vector of reduced costs $\bar{\mathbf{c}}$ contains more than $n - m$ zero elements.

In theory, degeneracy can cause the appearance of a basis \mathcal{B} more than once in the pivoting procedure which can lead to an infinite number of pivots being performed, i.e. cycling of the simplex algorithm. Furthermore, degeneracy in parametric linear programming may introduce additional complexity in the solution procedure and may cause the solution to be incorrect or incomplete if not resolved properly [5]. A systematic way to resolve degeneracy in LPs and parametric LPs is the use of *lexicographic perturbation* [7].

C. Lexicographic perturbation

The crucial concept used in this scheme is *lexicographic (lexico) positivity*.

Definition 4.3 (Lexico positivity [7]): A vector $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_n]^T$ is called *lexico positive* if $\boldsymbol{\gamma} \neq \mathbf{0}$ and the first non-zero component of $\boldsymbol{\gamma}$ is strictly positive.

Lexico positivity of a vector $\boldsymbol{\gamma}$ is denoted as $\boldsymbol{\gamma} \succ 0$. A vector $\boldsymbol{\gamma}$ is referred to as *lexico negative* if $-\boldsymbol{\gamma} \succ 0$. For a pair of vectors $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$, $\boldsymbol{\gamma}_1 \succ \boldsymbol{\gamma}_2$ if $\boldsymbol{\gamma}_1 - \boldsymbol{\gamma}_2 \succ 0$. Following the same reasoning, in the set of vectors $\{\boldsymbol{\gamma}^1, \dots, \boldsymbol{\gamma}^n\}$, a *lexico minimum* is defined as the vector $\boldsymbol{\gamma}^p$ such that $\boldsymbol{\gamma}^q - \boldsymbol{\gamma}^p \succ 0$ for all $q \in \{1, \dots, n\}$ and $q \neq p$. A matrix is called *lexico positive* if all of its rows are *lexico positive*. We will abuse the notation and denote *lexico positivity* of a matrix \mathbf{M} as $\mathbf{M} \succ 0$.

Consider the following *fully perturbed* problem:

$$\begin{aligned} \min_{\mathbf{x}} \quad & (\mathbf{c} + \boldsymbol{\delta})^T \mathbf{x} \\ \text{s. t.} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} + \boldsymbol{\varepsilon}, \quad \mathbf{x} \geq 0, \end{aligned} \quad (10)$$

where $\boldsymbol{\delta} \triangleq [\delta_0, \delta_0^2, \dots, \delta_0^n]$, $\boldsymbol{\varepsilon} \triangleq [\varepsilon_0, \varepsilon_0^2, \dots, \varepsilon_0^m]$ and the scalars δ_0 and ε_0 are strictly positive. For the perturbed problem (10), given a feasible basis \mathcal{B} and the corresponding set of non-basic indices \mathcal{N} , the following statements are valid.

Theorem 4.2 ([7]): If the basis \mathcal{B} is primal feasible for (10) for a sufficiently small $\varepsilon_0 > 0$, then it is primal feasible

for (6). Furthermore, if the basis \mathcal{B} is *dual feasible* for (10), it is also dual feasible for unperturbed problem (6)-(7).

Feasibility of a basis for a perturbed LP problem (10) will be referred to as *lexicographic* or *lexico* feasibility. The following theorem gives a criterion for verifying *lexico* feasibility of a basis.

Theorem 4.3 (Primal and dual lexico feasibility [7]):

For the perturbed problem (10) a basis \mathcal{B} is:

- a) **primal lexico feasible** if and only if $\begin{bmatrix} \bar{\mathbf{b}} & \mathbf{A}_{\cdot, \mathcal{B}}^{-1} \end{bmatrix} \succ 0$.
- b) **dual lexico feasible** if and only if $\begin{bmatrix} \bar{\mathbf{c}} & \mathbf{I} - \hat{\mathbf{A}} \end{bmatrix} \succ 0$,

$$\text{where } \hat{\mathbf{A}}_{\cdot, \mathcal{B}} = \bar{\mathbf{A}}^T, \hat{\mathbf{A}}_{\cdot, \mathcal{N}} = \mathbf{0}.$$

The following theorem justifies the use of *lexicographic perturbation*:

Theorem 4.4: Given any $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{c} \in \mathbb{R}^n$, there exist $\delta_1, \varepsilon_1 > 0$ such that for all $\delta, 0 < \delta < \delta_1$, and $\varepsilon, 0 < \varepsilon < \varepsilon_1$, the problem (10) is neither primal nor dual degenerate.

Remark 4.1: If the original problem (6)-(7) has an optimal solution, according to Theorem 4.4, for arbitrarily small positive values of δ and ε , the optimal solution of the perturbed problem (10) is defined by a single *unique basis*. *Lexicographic perturbation* is applied through modified primal and dual pivoting rules by replacing positivity by *lexico* positivity and negativity by *lexico non-positivity*, as given by the following algorithms.

Algorithm IV.3 Lexico Primal Simplex pivot

- 1: choose index e of the entering variable such that

$$\begin{bmatrix} \bar{\mathbf{c}} & \mathbf{I} - \hat{\mathbf{A}} \end{bmatrix}_e \preceq 0$$
 - 2: choose the leaving variable index l as:

$$l = \operatorname{arglexmin}_{j \in \{1, \dots, m\}} \left\{ \frac{\begin{bmatrix} \bar{\mathbf{b}} & \mathbf{A}_{\cdot, \mathcal{B}}^{-1} \end{bmatrix}_j}{\bar{\mathbf{A}}_{je}} \mid \bar{\mathbf{A}}_{je} > 0 \right\}$$
 - 3: update the basis: $\mathcal{B} \leftarrow \mathcal{B} \cup \{e\} \setminus \{\mathcal{B}_l\}$
-

Algorithm IV.4 Lexico Dual Simplex pivot

- 1: choose index l of the leaving variable as $\begin{bmatrix} \bar{\mathbf{b}} & \mathbf{A}_{\cdot, \mathcal{B}}^{-1} \end{bmatrix}_l \preceq 0$
 - 2: choose the entering variable index e as:

$$e = \operatorname{arglexmin}_{j \in \mathcal{N}} \left\{ -\frac{\begin{bmatrix} \bar{\mathbf{c}} & \mathbf{I} - \hat{\mathbf{A}} \end{bmatrix}_j}{\bar{\mathbf{A}}_{lj}} \mid \bar{\mathbf{A}}_{lj} < 0 \right\}$$
 - 3: update the basis: $\mathcal{B} \leftarrow \mathcal{B} \cup \{e\} \setminus \{\mathcal{B}_l\}$
-

Similarly to the standard simplex algorithm, *lexico* simplex pivot rules preserve a particular feasibility of the bases generated in the solution process.

Theorem 4.5 ([7]): Given a primal *lexico* feasible basis, performing a primal pivot according to Algorithm IV.3 preserves *lexico* primal feasibility. Analogously, given an initial dual *lexico* feasible basis, performing dual *lexico* pivots according to Algorithms IV.4 maintains *lexico* dual feasibility.

It can be shown that, if the optimal solution exists, after finitely many iterations a *lexico* simplex algorithms IV.3-IV.4 terminate with a basis which is *lexico* primal and *lexico* dual feasible, i.e. *lexico optimal*.

Remark 4.2: Note that the variables δ and ε are not assigned real values. Therefore, the use of the *lexicographic perturbation* scheme does not raise any numerical problems.

V. RIM MULTI-PARAMETRIC LINEAR PROGRAM

Consider the following *RIM* mpLP:

$$J^*(\theta) = \min_{\mathbf{x}} (\mathbf{c} + \mathbf{E}\theta)^T \mathbf{x} \quad (11)$$

$$\text{s. t. } \mathbf{A}\mathbf{x} = \mathbf{S}\theta + \mathbf{b}, \quad \mathbf{x} \geq 0 \quad (12)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of optimization variables, $\theta \in \mathcal{Q} \subset \mathbb{R}^p$ is the vector of parameters and $\mathbf{A} \in \mathbb{R}^{m \times n}$. In the following, we only state the relevant properties of the problem (11) and the corresponding solution. For an in-depth treatment of the subject, the reader is referred to [11], [12], [9].

The following theorem summarizes properties of the solution to the mpLP (11):

Theorem 5.1: Let $\mathcal{Q}^* \subseteq \mathcal{Q}$ be the set of parameters θ for which the linear program (11) has a finite optimal solution. Then, \mathcal{Q}^* is a closed polyhedral set and the optimizer $\mathbf{x}^*(\theta)$ is a *piecewise affine function defined over polyhedra* within the set \mathcal{Q}^* , i.e. $\mathbf{x}^*(\theta) = \Phi^i \theta + \gamma^i$, if $\theta \in \mathcal{CR}^i$, where $\{\mathcal{CR}^i\}_{i=1}^R$, with $\mathcal{Q}^* = \bigcup_{i=1}^R \mathcal{CR}^i$ and $\mathcal{CR}^i \cap \mathcal{CR}^j = \emptyset$ for $i \neq j$.

Solving the mpLP (11)-(12) means to compute the closed form solution for all values of parameter $\theta \in \mathcal{Q}^*$, i.e. computing the representation for all regions \mathcal{CR}^i and the corresponding matrices defining the optimizer and the cost as functions of the parameters. The regions \mathcal{CR}^i are commonly referred to as *critical regions* and represent sets of parameters for which the optimal solution to the corresponding LP satisfies the same optimality conditions [12]. Conceptually, the procedure for solving the parametric program (11)-(12) amounts to solving a number of *linear programs* for different fixed values of the parameter θ and computing critical regions from their optimality conditions.

A. Lexico perturbed RIM mpLP

As already mentioned, degeneracy in parametric programming has to be handled properly in order to guarantee correctness of the solution. In this section we present a simple algorithm for solving *RIM mpLPs* using lexicographic perturbation for resolving degeneracy. Consider the following full lexicographic perturbed *RIM* mpLP:

$$J^*(\theta) = \min_{\mathbf{x}} (\mathbf{c} + \mathbf{E}\theta + \delta)^T \mathbf{x} \quad (13)$$

$$\text{s. t. } \mathbf{A}\mathbf{x} = \mathbf{b} + \mathbf{S}\theta + \varepsilon, \quad \mathbf{x} \geq 0, \quad (14)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the vector of optimization variables, $\theta \in \mathcal{Q} \subset \mathbb{R}^p$ is the vector of parameters, $\mathbf{A} \in \mathbb{R}^{m \times n}$ and ε and δ are lexicographic perturbations.

We will make two assumptions:

Assumption 5.1: There is a finite optimal solution for every $\theta \in \mathcal{Q}$.

Assumption 5.2: The set of parameters \mathcal{Q} is full dimensional.

For a basis \mathcal{B} , the associated critical region is defined as:

$$\mathcal{CR}_{\mathcal{B}} = \left\{ \theta \mid \left[\begin{array}{cc} \bar{\mathbf{b}} + \bar{\mathbf{S}}\theta & \mathbf{A}_{\cdot \mathcal{B}}^{-1} \\ \bar{\mathbf{c}} + \bar{\mathbf{E}}\theta & \mathbf{I} - \hat{\mathbf{A}} \end{array} \right] \succ 0 \right\} \quad (15)$$

Note that the critical region $\mathcal{CR}_{\mathcal{B}}$ can be an empty set if the basis \mathcal{B} is not *lexico* optimal for some $\theta \in \mathcal{Q}$. Since the values of ε and δ are considered to be arbitrarily close to zero, the closure of a critical region is given by:

$$\overline{\mathcal{CR}}_{\mathcal{B}} = \left\{ \theta \mid \left[\begin{array}{c} \bar{\mathbf{c}}_{\mathcal{N}} + \bar{\mathbf{E}}_{\mathcal{N}} \cdot \theta \\ \bar{\mathbf{b}} + \bar{\mathbf{S}}\theta \end{array} \right] \geq 0 \right\}. \quad (16)$$

The algorithm presented in this section enumerates all lexico optimal bases for which the corresponding critical region is full dimensional, i.e. $\dim(\mathcal{CR}_{\mathcal{B}}) = p$. Since the optimal basis for each parameter θ is unique, the intersection of any two full dimensional critical regions is empty. As we assume that for all parameter vectors $\theta \in \mathcal{Q}$ there exists a finite optimal solution, the set \mathcal{Q} is fully covered by the *closures* of full dimensional critical regions. For convenience, in the remainder of the paper we will use the term “critical region” to denote its closure.

The algorithm is initialized by choosing a parameter vector $\theta_0 \in \text{int}(\mathcal{Q})$ and solving a lexico perturbed linear program (10) for the chosen θ_0 . The initial parameter θ_0 can be chosen, for example, as a randomly perturbed Chebyshev center of the polyhedron \mathcal{Q} . Random perturbation practically ensures that the initial lexico optimal basis \mathcal{B}^0 defines a full dimensional critical region. If this is not the case then the whole procedure is repeated until the initial critical region is full dimensional.

The computed initial critical region $\overline{\mathcal{CR}}_{\mathcal{B}^0}$ is defined by the set of inequalities (16). This set of inequalities is not necessarily minimal, i.e. it may contain inequalities whose removal does not change the critical region. Such constraints are called *redundant* and the minimal representation of the critical region is obtained by removing all redundant constraints from (16). The procedure requires solving a number of LPs, equal in the worst case to the number of inequalities in (16) (cf. [13]).

Definition 5.1 (Adjacent critical regions):

Critical regions $\overline{\mathcal{CR}}_{\mathcal{B}^1}$ and $\overline{\mathcal{CR}}_{\mathcal{B}^2}$ are *adjacent* if $\overline{\mathcal{CR}}_{\mathcal{B}^1} \cap \overline{\mathcal{CR}}_{\mathcal{B}^2}$ is of dimension $p - 1$.

We will refer to the bases \mathcal{B}^1 and \mathcal{B}^2 defining adjacent critical regions as *adjacent bases*. Starting from the initial basis \mathcal{B}^0 , the algorithm explores the set \mathcal{Q} recursively by performing the following two actions:

- 1) computing the minimal representation of the critical region defined by the basis,
- 2) computing the bases defining adjacent critical regions along each facets of the critical region

These two steps are repeated for each of the newly discovered bases until all of the bases defining full-dimensional critical regions have been found. The complete procedure is outlined in Algorithm V.1.

The central component in the algorithm is the computation of adjacent critical regions along a facet of the current region. The following section explains the procedure in details.

B. Computation of adjacent regions

Assume that initially a lexico optimal basis \mathcal{B} is given and that the minimal representation of the region $\overline{\mathcal{CR}}_{\mathcal{B}}$ is

Algorithm V.1 RIM mpLP

```

1: compute an initial lexico optimal basis  $\mathcal{B}^0$ 
2: initialize the list of unexplored bases:
    $\mathcal{L}_{unexplored} \leftarrow \{\mathcal{B}^0\}$ 
3: initialize the list of discovered bases:  $\mathcal{L}_{found} \leftarrow \{\mathcal{B}^0\}$ 
4: while  $\mathcal{L}_{unexplored} \neq \emptyset$  do
5:   remove any basis  $\mathcal{B}$  from  $\mathcal{L}_{unexplored}$ 
6:   compute the min. representation of the region  $\mathcal{CR}_{\mathcal{B}}$ 
7:   for all facets  $\mathcal{F}^i$  of  $\mathcal{CR}_{\mathcal{B}}$  do
8:     compute bases  $\{\mathcal{B}^j\}_{j=1}^r$  defining adjacent critical
       regions intersecting  $\mathcal{CR}_{\mathcal{B}}$  on  $\mathcal{F}^i$ 
9:     for all of the bases  $\mathcal{B}^j$  do
10:      if the basis  $\mathcal{B}^j$  is not in  $\mathcal{L}_{found}$  then
11:         $\mathcal{L}_{found} \leftarrow \mathcal{L}_{found} \cup \{\mathcal{B}^j\}$ 
12:         $\mathcal{L}_{unexplored} \leftarrow \mathcal{L}_{unexplored} \cup \{\mathcal{B}^j\}$ 
13:      end if
14:    end for
15:  end for
16: end while

```

described by the following set of inequalities:

$$\overline{\mathcal{CR}}_{\mathcal{B}} = \{\boldsymbol{\theta} \mid \mathbf{H}\boldsymbol{\theta} \leq \mathbf{k}\}.$$

Consider the facet of the region $\overline{\mathcal{CR}}_{\mathcal{B}}$ given by the f^{th} constraint:

$$\mathcal{F} = \{\boldsymbol{\theta} \in \overline{\mathcal{CR}}_{\mathcal{B}} \mid \mathbf{H}_f \boldsymbol{\theta} = \mathbf{k}_f\}$$

where \mathbf{H}_f denotes the f^{th} row of the matrix \mathbf{H} . The goal is to find, starting from the initial basis \mathcal{B} , all the bases that are lexico optimal “just across the facet”, i.e. for parameter $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}_0 + \alpha \mathbf{H}_f^T$ with $\boldsymbol{\theta}_0 \in \mathcal{F}$ and $\alpha > 0$ arbitrarily small. When the value of α is increased from zero, some of the constraints in (15), defining the optimality of \mathcal{B} , will change from lexico positive to lexico non-positive. This change indicates the loss of primal or dual feasibility (or both) for the parameter $\hat{\boldsymbol{\theta}}$. The following two sets of indices identify constraints that may change sign when moving from some $\boldsymbol{\theta}_0$ to $\hat{\boldsymbol{\theta}}$:

$$\mathcal{T}_p = \{i \mid \bar{\mathbf{c}}_i + \bar{\mathbf{S}}_i \boldsymbol{\theta} = 0, \forall \boldsymbol{\theta} \in \mathcal{F}\} \quad (17)$$

$$\mathcal{T}_d = \{i \in \mathcal{N} \mid \bar{\mathbf{c}}_i + \bar{\mathbf{E}}_i \boldsymbol{\theta} = 0, \forall \boldsymbol{\theta} \in \mathcal{F}\}, \quad (18)$$

where \mathcal{N} is the ordered set of non-basic indices corresponding to the basis \mathcal{B} . Note that all constraints not in \mathcal{T}_p and \mathcal{T}_d remain strictly positive for some sufficiently small $\alpha > 0$. We distinguish the following cases:

- 1) ($\mathcal{T}_p \neq \emptyset$ and $\mathcal{T}_d = \emptyset$) **OR** ($\mathcal{T}_p = \emptyset$ and $\mathcal{T}_d \neq \emptyset$)
- 2) $\mathcal{T}_p \neq \emptyset$ and $\mathcal{T}_d \neq \emptyset$

The case when both \mathcal{T}_p and \mathcal{T}_d are empty is not possible as this would imply the optimality of the basis \mathcal{B} outside the region $\overline{\mathcal{CR}}_{\mathcal{B}}$. In the case 1, dual or primal lexico feasibility of the basis \mathcal{B} is preserved for the parameter $\hat{\boldsymbol{\theta}}$, i.e. on the other side of the facet \mathcal{F} . If the basis \mathcal{B} is dual lexico feasible at $\hat{\boldsymbol{\theta}}$, the goal is to achieve primal lexico feasibility while preserving the initial dual lexico feasibility. This is done by performing dual lexico pivots. Likewise, if the basis \mathcal{B} is primal lexico feasible at $\hat{\boldsymbol{\theta}}$, we take primal lexico pivots.

However, in case 2 when the initial basis \mathcal{B} is neither primal nor dual lexico feasible for the parameter $\hat{\boldsymbol{\theta}}$, we cannot

proceed with the simplex pivoting scheme outlined above. Instead, we consider a different pivoting scheme known as a *Criss-Cross method*, which does not require primal or dual lexico feasibility of the initial basis \mathcal{B} [14].

We will now give detailed description of the pivoting schemes for all three cases.

1) *Dual Pivots*: Assume the case when $\mathcal{T}_p \neq \emptyset$ and $\mathcal{T}_d = \emptyset$, i.e. when the initial basis \mathcal{B} is dual lexico feasible for the parameter $\hat{\boldsymbol{\theta}}$.

Consider the following auxiliary fully perturbed parametric LP:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_i [\mathbf{c} + \mathbf{E}\boldsymbol{\theta} + \alpha \mathbf{E}\mathbf{H}_f^T + \boldsymbol{\delta}]_i \mathbf{x}_i \\ \text{s. t.} \quad & \sum_i \mathbf{A}_{ji} \mathbf{x}_i = \mathbf{S}_j \mathbf{H}_f^T + \boldsymbol{\varepsilon}_j, \quad \mathbf{x}_i \geq 0, \quad (19) \\ & \boldsymbol{\theta} \in \mathcal{F}, \quad j \in \mathcal{T}_p, \quad i \in \mathcal{B}_{\mathcal{T}_p} \cup \mathcal{N} \end{aligned}$$

where $\alpha > 0$ is arbitrarily small and $\boldsymbol{\delta}$ and $\boldsymbol{\varepsilon}$ are lexicographic perturbation vectors.

Proposition 5.1: Let $\{\hat{\mathcal{B}}^i\}$ denote the set of lexico optimal bases defining full dimensional critical regions for the parametric problem (19). The set of bases defining critical regions adjacent to $\overline{\mathcal{CR}}_{\mathcal{B}}$ along the facet \mathcal{F} is defined as $\{\hat{\mathcal{B}}^i \cup \mathcal{B}_{\mathcal{J}}\}$ where $\mathcal{J} = \mathcal{B} \setminus \mathcal{B}_{\mathcal{T}_p}$.

Proof: See [15]. ■

As stated above, the solution to the problem (19) is obtained by performing dual simplex pivots starting with the basis \mathcal{B} , which is dual lexico feasible for a sufficiently small $\alpha > 0$, implying the following lexico positivity:

$$[\bar{\mathbf{c}} + \bar{\mathbf{E}}\boldsymbol{\theta} + \alpha \bar{\mathbf{E}}\mathbf{H}_f^T \quad \mathbf{I} - \hat{\mathbf{A}}] \succ 0. \quad (20)$$

The condition (20) has to be fulfilled for some $\alpha > 0$ arbitrary small and, therefore, the lexico inequality (20) can be equivalently written as:

$$[\bar{\mathbf{c}} + \bar{\mathbf{E}}\boldsymbol{\theta} \quad \bar{\mathbf{E}}\mathbf{H}_f^T \quad \mathbf{I} - \hat{\mathbf{A}}] \succ 0. \quad (21)$$

In each pivot step, given index l of the variable leaving the basis, the index of the entering variable is chosen according to the following rule:

$$e = \operatorname{arglexmin}_{j \in \mathcal{N}} \left\{ \frac{[\bar{\mathbf{c}} + \bar{\mathbf{E}}\boldsymbol{\theta} \quad \bar{\mathbf{E}}\mathbf{H}_f^T \quad \mathbf{I} - \hat{\mathbf{A}}]_j}{(-\bar{\mathbf{A}}_{lj})} \mid \bar{\mathbf{A}}_{lj} < 0 \right\}. \quad (22)$$

The lexico minimum rule (22) is derived directly from the standard dual simplex method (Algorithm IV.4) and guarantees that dual lexico feasibility of the basis is preserved for $\hat{\boldsymbol{\theta}}$. If $[\bar{\mathbf{c}} + \bar{\mathbf{E}}\boldsymbol{\theta}]_j > 0$ for all $j \in \mathcal{N}$ such that $\bar{\mathbf{A}}_{lj} < 0$, lexico minimum choice (22) is parameter dependent. For this case, we define the set of variables \mathcal{E} as the set of all $j \in \mathcal{N}$ for which there exists $\boldsymbol{\theta} \in \mathcal{F}$ and j is the minimum in (22). Consider the following polyhedron:

$$\tilde{\mathcal{F}}^j = \left\{ \boldsymbol{\theta} \in \mathcal{F} \mid \frac{\bar{\mathbf{c}}_j + \bar{\mathbf{E}}_j \boldsymbol{\theta}}{(-\bar{\mathbf{A}}_{lj})} < \frac{\bar{\mathbf{c}}_k + \bar{\mathbf{E}}_k \boldsymbol{\theta}}{(-\bar{\mathbf{A}}_{lk})} \right\}, \quad (23)$$

where $\bar{\mathbf{A}}_{lj}, \bar{\mathbf{A}}_{lk} < 0$ and $j, k \in \mathcal{N}, j \neq k$. A variable j enters the set \mathcal{E} if the polyhedron $\tilde{\mathcal{F}}^j \subset \mathcal{F}$ is not empty. This can

be verified by computing the Chebyshev center of the polyhedron $\tilde{\mathcal{F}}^j$, which requires solving a single linear program. The complete procedure is outlined in Algorithm V.2.

Algorithm V.2 Function DualPivot

INPUT: dual lexico feasible basis \mathcal{B} , the facet \mathcal{F}

OUTPUT: list of optimal adjacent bases \mathcal{L}_{adj}

```

1: select the leaving variable:
    $l = \operatorname{arglexmin}_{j \in \mathcal{T}_p} \left\{ \left[ \overline{\mathbf{S}}\mathbf{H}_f^T \quad \mathbf{A}_{\cdot j}^{-1} \right]_j \preceq 0 \right\}$ 
2: if  $\{l\} = \emptyset$  then
3:   return  $\mathcal{L}_{adj} \leftarrow \{\mathcal{B}\}$ 
4: end if
5: select entering candidates:  $\mathcal{E}_c = \{j \in \mathcal{N} \mid \overline{\mathbf{A}}_{lj} < 0\}$ 
6: if  $\mathcal{E}_c = \emptyset$  then
7:   return  $\emptyset$  {/* dual problem is unbounded */}
8: end if
9:  $\mathcal{Z}_e = \left\{ j \in \mathcal{E}_c \mid \left[ \overline{\mathbf{c}} + \overline{\mathbf{E}}\theta \right]_j = 0, \forall \theta \in \mathcal{F} \right\}$ 
10: if  $\mathcal{Z}_e \neq \emptyset$  then
11:   entering variable:  $e = \operatorname{arglexmin}_{j \in \mathcal{E}_c} \left\{ \frac{\left[ \overline{\mathbf{E}}\mathbf{H}_f^T \quad \mathbf{I} - \hat{\mathbf{A}} \right]_j}{(-\mathbf{A}_{lj})} \right\}$ 
12:   return DualPivot( $\mathcal{B} \cup \{e\} \setminus \{l\}, \mathcal{F}$ )
13: end if
14: for all  $j \in \mathcal{E}_c$  do
15:   compute  $\tilde{\mathcal{F}}$  according to (23)
16:    $\mathcal{L}_{adj} \leftarrow \mathcal{L}_{adj} \cup \text{DualPivot}(\mathcal{B} \cup \{j\} \setminus \{l\}, \tilde{\mathcal{F}})$ 
17: end for
18: return  $\mathcal{L}_{adj}$ 

```

2) *Primal Pivots:* In the case when the basis \mathcal{B}^0 is primal lexico feasible for the parameter $\hat{\theta}$, according to the same reasoning, the following auxiliary problem is formulated:

$$\begin{aligned}
 \min_{\mathbf{x}} \quad & \sum_j \left[\overline{\mathbf{E}}\mathbf{H}_f^T + \delta \right]_j \mathbf{x}_j \\
 \text{s. t.} \quad & \sum_j \mathbf{A}_{\cdot j} \mathbf{x}_j = \mathbf{b} + \mathbf{S}\theta + \alpha \mathbf{S}\mathbf{H}_f^T + \varepsilon, \quad \mathbf{x}_j \geq 0, \quad (24) \\
 & \mathbf{x}_k = 0, \quad \theta \in \mathcal{F}, \quad j \in \mathcal{T}_d, \quad k \in \mathcal{N} \setminus \mathcal{T}_d
 \end{aligned}$$

The problem (24) is solved by performing primal simplex pivots. The lexicographic rule for the selection of the entering variable e and the leaving variable(s) is given by:

$$e = \operatorname{arglexmin}_{i \in \mathcal{N}} \left\{ \left[\overline{\mathbf{E}}\mathbf{h}_f^T \quad \mathbf{I} - \hat{\mathbf{A}} \right]_i \preceq 0 \right\} \quad (25)$$

$$l = \operatorname{arglexmin}_{j \in \{1, \dots, m\}} \left\{ \frac{\left[\overline{\mathbf{b}} + \overline{\mathbf{S}}\theta \quad \overline{\mathbf{E}}\mathbf{H}_f^T \quad \mathbf{A}_{\cdot j}^{-1} \right]_j}{\overline{\mathbf{A}}_{B_j e}} \mid \overline{\mathbf{A}}_{B_j e} > 0 \right\}, \quad (26)$$

3) *Criss-Cross Method:* In case 2 of Section V-B, when both primal and dual lexico feasibility of a current basis \mathcal{B} are violated for the parameter $\hat{\theta}$, direct application of the above proposed methods is not possible. An elegant and simple solution is to apply the so called *Criss-Cross* pivoting scheme [14]¹, which does not require an initial *lexico feasible* basis. The method used in the algorithm is also known as the *least-index Criss-Cross* method. A pivot iteration begins by choosing the variable with the smallest index among basic

variables violating primal lexico feasibility and non-basic variables with a negative reduced cost. If a non-basic variable is chosen, a primal pivot is performed. Similarly, if the least index variable is a basic variable, then the basis is updated using a dual simplex pivot. In order to obtain a unique lexico optimal basis, we incorporate lexico rules into the algorithm. Detailed explanation of the algorithm can be found in [15].

In theory, the presented Criss-Cross pivoting scheme could also be used for the cases when the initial basis is primal or dual lexico feasible for the parameter $\hat{\theta}$. However, practical experience shows that the Criss-Cross algorithm typically requires more pivoting iterations compared to the primal or dual simplex pivoting procedures.

VI. CONCLUDING REMARKS

A novel, computationally efficient algorithm for solving *RIM* multi-parametric linear programs is presented. The algorithm uses the concept of lexicographic perturbation for resolving problems related to degeneracy. The resulting solution procedure amounts to enumerating the bases defining full dimensional critical regions by simple pivoting operations. The algorithm presented in this paper guarantees completeness of the obtained solution.

REFERENCES

- [1] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, Jan. 2002.
- [2] F. Borrelli, A. Bemporad, and M. Morari, "A geometric algorithm for multi-parametric linear programming," *J. Optimiz. Theory App.*, vol. 118, no. 3, pp. 515–540, Sept. 2003.
- [3] M. Baotić, "An Efficient Algorithm for Multiparametric Quadratic Programming," ETH Zürich, Automatic Control Lab, Tech. Rep. AUT02-05, Apr. 2002. [Online]. Available: <http://control.ee.ethz.ch>
- [4] P. Tøndel, T. Johansen, and A. Bemporad, "An algorithm for multi-parametric quadratic programming and explicit mpc solutions," *Automatica*, vol. 39, no. 3, pp. 489–497, 2003.
- [5] J. Spjøtvold, P. Tøndel, and T. Johansen, "A method for obtaining continuous solutions to multiparametric linear programs," in *Proc. 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- [6] C. Jones, "Polyhedral tools for control," Ph.D. dissertation, University of Cambridge, Control group, Dept. of Engineering, 2004.
- [7] K. Murty, *Linear Programming*. John Wiley & Sons, 1983.
- [8] M. Barić, M. Baotić, and M. Morari, "On-line tuning of controllers for systems with constraints," in *Proc. 44th IEEE Conf. on Decision and Control*, Sevilla, Spain, dec 2005.
- [9] H. Greenberg, "Simultaneous primal-dual right-hand-side sensitivity analysis from a strictly complementary solution of a linear program," *SIAM J. Optim.*, vol. 10, no. 2, pp. 427–442, 2000.
- [10] F. Borrelli, *Constrained Optimal Control Of Linear And Hybrid Systems*, ser. Lecture Notes in Control and Information Sciences. Springer, 2003, vol. 290.
- [11] B. Bank, J. Guddat, D. Klatté, B. Kummer, and K. Tammer, *Non-Linear Parametric Optimization*. Birkhäuser, 1983.
- [12] T. Gal, *Postoptimal Analyses, Parametric Programming, and Related Topics*, 2nd ed. Berlin: de Gruyter, 1995.
- [13] K. Clarkson, "More output-sensitive geometric algorithms," in *Proc. 35th IEEE Sympos. Found. of Comput. Sci.*, 1994, pp. 695–702.
- [14] K. Fukuda and T. Terlaky, "Criss-cross methods: A fresh view on pivot algorithms," *Math. Program.*, vol. 79, pp. 369–395, 1997.
- [15] M. Barić, C. Jones, and M. Morari, "Parametric Analysis of Controllers for Constrained Linear Systems," ETH Zürich, Automatic Control Lab, Tech. Rep. AUT06-06, Mar. 2006. [Online]. Available: <http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=2471>

¹The authors gratefully acknowledge prof. Fukuda Komei for suggesting the application of the Criss-Cross method