The Double Description Method for the Approximation of Explicit MPC Control Laws

Colin N. Jones and Manfred Morari

***Abstract*— A standard model predictive controller (MPC) can be written as a parametric optimization problem whose solution is a piecewise affine (PWA) map from the measured state to the optimal control input. The primary limitation of this optimal 'explicit solution' is that the complexity can grow quickly with problem size, and so in this paper we seek to compute approximate explicit control laws that can trade-off complexity for approximation error. This computation is accomplished in a two-phase process: First, inner and outer polyhedral approximations of the the convex cost function of the parametric problem are computed with an algorithm based on an extension to the classic double-description method; a convex hull approach. The proposed method has two main advantages from a control point of view: it is an incremental approach, meaning that an approximation of any specified complexity can be produced and it operates on implicitly-defined convex sets, meaning that the optimal solution of the parametric problem is not required. In the second phase of the algorithm, a feasible approximate control law is computed that has the cost function derived in the first phase. For this purpose, a new interpolation method is introduced based on recent work on barycentric interpolation. The resulting control law is continuous, although non-linear and defined over a non-simplical polytopic partition of the state space. The non-simplical nature of the partition generates significantly simpler approximate control laws than current competing methods, as demonstrated on computational examples.**

I. Introduction

Implementing a model predictive controller (MPC) generally requires the solution of an optimization problem on–line at each sampling instant. In recent years, it has become well-known that this optimization problem can be posed parametrically, with the measured state as the parameter. For the case of linear systems subject to linear constraints with the goal of minimizing a polytopic norm, the result is a (multi)parametric linear program (pLP)

$$J^\star(x) := \min\left\{g^T u \mid Gu \le Ex + w\right\} \quad , \qquad (1)$$

where x is the state, u is the control input and appropriate slack variables and the cost function $J^\star$ is a convex piecewise affine function (PWA). Solving this parametric problem off–line results in a PWA function $u^\star(x)$ mapping the measured state to the optimal system input [1]–[3]. If this PWA function is available, the on–line calculation of the control input then becomes one of evaluating $u^\star(x)$ at the current measured state x, which can decrease the required online computation time by orders of magnitude for some systems.

The primary limitation of this approach is the complexity of the PWA control law (i.e. the number of 'pieces'), which can grow quickly with problem size. In this paper, we propose a new algorithm for computing inner and outer polyhedral approximations of arbitrary convex sets, which we then put to work approximating the epigraph of the optimal cost function epi $J^\star$. The approach is based on the well established double description method [4], [5], which is an incremental algorithm for computing convex hulls. We extend this method so that it can work on *implicitly* defined convex bodies, such as the unknown cost function of a convex parametric program (i.e. we compute the approximation without first computing the optimal solution $J^\star$).

The incremental nature of the approach has a very useful benefit from the explicit MPC point of view. Specifically, the common reason for such an approximation is to generate a control law that can be evaluated in a given amount of time, or be stored in a given amount of space. Because the double description algorithm is incremental, it can simply be run until the complexity of the approximation has reached the physical time or storage limits of the on–line computational platform.

Several authors have proposed approximation algorithms that can produce simpler PWA control laws at the cost of optimality. These approaches operate in a two–stage procedure: first the epigraph of the optimal cost function $J^\star$ is approximated with a simpler polyhedron $\tilde{J}$, which determines the stability and performance properties of the approximate controller and then second, a control law $\tilde{u}$ is computed such that $g^T\tilde{u} = \tilde{J}$. However, generating such a feasible control law $\tilde{u}$ is not immediate. Existing proposals either produce a triangulation and then interpolate the optimal control law at the vertices [6], [7], have a post-processing step in which an exact parametric program is calculated based on the approximate cost [8], [9] or compute control laws based on sub-divisions of hypercubes [10]. In all cases, the requirement of computing a control law that can generate the approximate cost places restrictions on the structure of the cost approximation and generally causes a significant increase in the achievable complexity.

In this paper, we introduce a new method of post-processing an approximate polyhedral cost $\tilde{J}$ based on barycentric interpolation, in order to compute a feasible non–linear control law $\tilde{u}$ so that $g^T\tilde{u} = \tilde{J}$. This allows us to derive a control law from any polyhedral approximate cost. The main benefit of this is that we do not have to restrict ourselves to considering approximation approaches that generate triangulations, and hence can directly compute

C.N. Jones and M. Morari are with the Department of Electrical Engineering, Swiss Federal Institute of Technology Zürich, ETL I28, Physikstrasse 3, 8092 Zürich, Switzerland, e-mail: cjones|morari@ee.ethz.ch

a control law $\tilde{u}$ for the non–simplical regions produced by the double–description algorithm, which we will see often produces much simpler approximations.

The remainder of the paper is organized as follows. Section II outlines the general problem of approximation for convex and compact sets. Section III provides background on the double description method and the following section generalizes this so that it can operate on implicitly defined convex sets based on two oracles that need to be specified for the structure of the set in question and Section V then defines these two oracles for the specific case of linear MPC. Section V-A introduces barycentric interpolation, which allows the computation of a control law from the approximate cost and finally Section VI provides some computational examples.

NOTATION

A *polyhedron* is the intersection of a finite number of halfspaces and a *polytope* is a bounded polyhedron. If A is a subset of $\mathbb{R}^d$, then $P(A)$ is the set $\{x \mid \langle a,x\rangle \le 1,\ \forall a \in A\}$, which is a polyhedron if A is finite. If V is a subset of $\mathbb{R}^d$, then the convex hull of V, $\operatorname{conv}(V)$ is the intersection of all convex sets containing V. If $V = \{v_0,\dots,v_n\}$ is a finite set, then $\operatorname{conv}(V) = \{\sum_{i=0}^n v_i\lambda_i \mid \lambda_i \ge 0,\ \sum \lambda_i = 1\}$.

Let S and C be convex and compact sets, then the *Hausdorff distance* $\rho(S,C)$ is

$$\rho(S,C) := \max\left\{ \sup_{x\in S}\inf_{y\in C}\|x-y\|_2,\ \sup_{y\in C}\inf_{x\in S}\|y-x\|_2 \right\}$$

II. PROBLEM STATEMENT AND PRELIMINARIES

Our goal is to find a polytope S that approximates to within a given tolerance a convex and compact (closed and bounded) set $C \subset \mathbb{R}^d$.

Definition 1 (ε-approximation): Let $C \subset \mathbb{R}^d$ be a compact and convex set that contains the origin and is full–dimensional $\dim C = d$. If ε is a strictly positive real number, then the polytope S is called an *ε-approximation* of C if $\rho(S,C) \le \varepsilon$, where $\rho(\cdot,\cdot)$ is the Hausdorff distance. S is called an outer (inner) $\varepsilon-$approximation if $C \subseteq S$ ($S \subseteq C$).

The following theorem states that searching for a polytopic approximation to a convex set is well-founded.

Theorem 1 ([11], [12]): If $C \subset \mathbb{R}^d$ is a convex and compact set, then for every $\varepsilon > 0$, there exists a finitely generated polytope S such that $\rho(S,C) < \varepsilon$.

The specific goal of this paper is to approximate the convex sets that arise in the computation of explicit MPC control laws. In this case a description of the convex set C, the epigraph of the optimal cost function, is generally not known *explicitly*, but rather only *implicitly* in terms of an optimization problem. While it is possible in some cases to generate an explicit representation of the set C, it is often computationally prohibitive and we seek to avoid it here. For this reason, we don't assume that a description of the set C is available, but only that we can evaluate its support function, which is defined as

$$\delta^\star(a \mid C) := \sup\{\langle a,x\rangle \mid x \in C\} \ .$$

In turn, the support function allows us to define two optimization problems that will be required. First, given a vector a defining a direction, we must be able to find an extreme point that maximizes the linear function $\langle a,x\rangle$ over the set C.

$$\texttt{extr}(a \mid C) \in \{x \in C \mid \langle a,x\rangle = \delta^\star(a \mid C)\} \tag{2}$$

Second, given a point $x \notin C$, the function $\texttt{maxsep}(x \mid C)$ returns a vector a defining a hyperplane that maximally separates x from C: $\langle a,x\rangle \ge 1$ and $C \subset \{x \mid \langle a,x\rangle \le 1\}$.

$$\texttt{maxsep}(x \mid C) \in \operatorname{argmax}\{\langle a,x\rangle - \delta^\star(a \mid C) \mid \|a\|_2 = 1\} \tag{3}$$

The next section gives a generic overview of the classic double description method as applied to polytopes. The section following then generalizes the method using the above two functions so that it can be used to compute an $\varepsilon-$approximation of an implicitly defined convex and compact set. The final section defines the functions `maxsep` and `extr` for the specific case of linear MPC and gives some computational examples.

III. CLASSIC DOUBLE DESCRIPTION METHOD

The Minkowski-Weyl theorem states that every polytope can be represented either as a convex combination of a finite number of points, or as the intersection of a finite number of halfspaces. This naturally leads to the following definition.

Definition 2 ([4], [5]): A pair (A,V) of finite sets $A,V \subset \mathbb{R}^d$ is called a *double description* (DD) if the following relationship holds:

$$x \in P(A) \qquad \text{if and only if} \qquad x \in \operatorname{conv}(V)$$

The classic double description method takes as input a description of a polytope in terms of a finite set $\mathscr{A}$ and the goal is to compute all of the vertices of $P(\mathscr{A})$. This is accomplished in an incremental fashion, beginning with a small subset $A \subset \mathscr{A}$ for which the vertices V of $P(A)$ can be directly computed, i.e. so that (A,V) is a DD-pair. During each iteration the set $A' = A \cup \{a\}$ is created by adding one vector $a \in \mathscr{A}$, or equivalently by intersecting the polytope $P(A)$ with the halfspace $\{x \mid \langle a,x\rangle \le 1\}$ and the set of vertices V is updated so that (A',V') remains a DD pair. This procedure continues until all of $\mathscr{A}$ has been inserted, at which point we have the DD pair $(\mathscr{A},V)$ and the vertices V of the polytope $P(\mathscr{A})$ have been enumerated.

The main operation of the algorithm is the updating of the set of vertices V so that (A',V') is a double description pair, which can be accomplished by a direct application of the following Lemma.

Lemma 1 (DD Lemma [4]): Let $A,V \subset \mathbb{R}^d$ be finite sets such that (A,V) is a DD pair and $\dim P(A) = d$. Let a be a vector in $\mathbb{R}^d$ and partition V into three sets

$$V^+ := \{v \mid \langle a,v\rangle < 1\}$$
$$V^= := \{v \mid \langle a,v\rangle = 1\}$$
$$V^- := \{v \mid \langle a,v\rangle > 1\}$$

If $A' := A \cup \{a\}$, then the pair (A', V') is a DD pair, where $V' = V^+ \cup V^= \cup V^{\text{new}}$

$$V^{\text{new}} := \left\{ f(v^+, v^-) \;\middle|\; \begin{array}{l} (v^+, v^-) \in V^+ \times V^-, \\ v^+ \text{ and } v^- \text{ are adjacent in } P(A) \end{array} \right\} ,$$

where

$$f(v^+, v^-) := \frac{(1 - \langle a, v^- \rangle)v^+ - (1 - \langle a, v^+ \rangle)v^-}{\langle a, v^+ - v^- \rangle} .$$

Furthermore, if V is a set of minimal extreme points for $P(A)$, then V' is minimal for $P(A')$.

With Lemma 1 in hand, we can now state the double description method as shown in Algorithm 1.

Algorithm 1 Classic Double Description Method

Require: A finite set $\mathscr{A} := \{a_1, \ldots, a_N\} \subset \mathbb{R}^d$, such that $\dim P(\mathscr{A}) = d$

Ensure: A minimal set $V \subset \mathbb{R}^d$, such that $P(\mathscr{A}) = \text{conv}(V)$

1: Obtain a DD pair $(\{a_i \mid i \in K\}, V)$, for some set $K \subset \{1, \ldots, N\}$ such that V is minimal
2: **while** $K \neq \{1, \ldots, N\}$ **do**
3: Select any index j from $\{1, \ldots, N\} \setminus K$
4: Construct a DD pair $(\{a_i \mid i \in K \cup \{j\}\}, V')$ using Lemma 1
5: $K := K \cup \{j\}$, $V := V'$
6: **end while**

IV. Implicit Double Description

Every convex and compact set C can be described as the intersection of a possibly infinite set of halfspaces or as the convex hull of a set of points [13, Thm. 11.5]; $C = P(\mathscr{A}) = \text{conv}(\mathscr{V})$ for some sets $\mathscr{A}$ and $\mathscr{V} \subset \mathbb{R}^d$. Computing a polytopic outer $\varepsilon-$approximation can then be stated as finding a finite subset A of $\mathscr{A}$ such that $\rho(P(A), C) \leq \varepsilon$. Equivalently, an inner approximation consists of a finite subset $V \subseteq \mathscr{V}$ such that $\rho(\text{conv}(V), C) \leq \varepsilon$.

The ideal would be to determine a set $A \subset \mathscr{A}$ of minimal cardinality. Computing such a set, however, is known to be NP-hard even in the simplest case when C is a polytope and the set $\mathscr{A}$ is finite and known [14]. Therefore, we here adopt an heuristic and incremental approach based on the double description algorithm which nonetheless has very useful properties.

At a given stage of the proposed implicit DD algorithm, two DD pairs (A_O, V_O) and (V_I, A_I) are maintained such that $\text{conv}(V_I)$ is an inner $\hat{\varepsilon}-$approximation of C and $P(A_O)$ an outer for some $\hat{\varepsilon} > \varepsilon$. We proceed with the DD algorithm as in the previous section, alternatingly improving either the inner or the outer approximation in each iteration by adding either a halfspace to the outer approximation $P(A_O)$ or a vertex to the inner $\text{conv}(V_I)$. The next section demonstrates how we choose an element of $\mathscr{A}$ such that the outer approximation improves and the section following discusses how we utilize the DD algorithm to likewise improve the inner approximation.

A. Improvement of the Outer Approximation

Let us first assume that we are improving the outer approximation, and hence our task is to choose a vector $a^\star \in \mathscr{A}$ so that the approximation error decreases

$$\rho(C, P(A_O \cup \{a^\star\})) \leq \rho(C, P(A_O)) .$$

The procedure that we will use is to first locate the vertex $v^\star$ of $P(A_O)$ that is a maximum distance from C and hence is defining the current approximation error. We will then remove this vertex from the approximation by computing the halfspace $a^\star$ that maximally separates $v^\star$ from C.

The current approximation error $\hat{\varepsilon}$ is given by the Hausdorff distance between $P(A_O)$ and C

$$\rho(C, P(A_O)) = \max_{y \in P(A_O)} \min_{x \in C} \|x - y\|_2 , \tag{4}$$

where we need only take the max over $P(A_O)$ and min over C and not vice versa because C is a subset of $P(A_O)$. We now seek to evaluate (4) in order to determine the point of $P(A_O)$ that is farthest from C. However, by assumption we cannot do direct computations on C, but can only evaluate the support function, which leads us to the following well-known lemma.

Lemma 2: If $C \subset \mathbb{R}^d$ is a convex, compact and full–dimensional set containing the origin and S is a polytope such that $C \subseteq S$, then

$$\rho(S, C)^2 = \max \left\{ \langle a, v \rangle - \delta^\star(a \mid C) \;\middle|\; \begin{array}{l} v \in \text{extr}(S) \\ a = \texttt{maxsep}(v \mid C) \end{array} \right\} ,$$

where $\text{extr}(S)$ are the vertices of S.

Proof: The Hausdorff distance is given by $\rho(C, S) = \max\{J(y) \mid y \in S\}$, where $J(y) := \min\{\|x - y\|_2 \mid x \in C\}$. The function $J(\cdot)$ is convex and therefore the maximum is obtained at an extreme point of S [13, Thm. 32.2]; $\rho(C, S) = \max\{J(v) \mid v \in \text{extr}(S)\}$. For a given extreme point $v \in \text{extr}(S)$, the minimum distance $J(v)$ is given by the `maxsep` function (3). ■

Remark 1: Because the vertices V_O are computed in an incremental fashion, it is not necessary to evaluate `maxsep` in Lemma 2 for each v in V_O in each iteration, but only those newly created in Lemma 1, V^{new}.

With Lemma 2 and the DD pair (A_O, V_O) in hand, we can now determine the set $V^\star \subset V_O$ of vertices that define the current approximation error; i.e. $\rho(\{v\}, C) = \rho(P(A_O), C)$ for all $v \in V^\star$. We proceed to choose a vertex $v^\star \in V^\star$ and compute the halfspace $P(\{a^\star\})$ that maximally separates $v^\star$ from C. The classic double description algorithm from the previous section then provides a mechanism to compute V'_O so that $(A_O \cup \{a^\star\}, V'_O)$ is a DD pair.

1) Approximation of the Hausdorff Distance: From Lemma 2 we see that evaluating the current approximation error between $P(A_O)$ and C requires the evaluation of the `maxsep` function up to $|V_O|$ times. In many cases, the evaluation of `maxsep` is very expensive and so we wish to avoid or reduce this if possible. In this section, we provide a method of bounding the Hausdorff distance without making any evaluations of the function `maxsep`.

We have available both an inner and an outer approximation of the set C, which together give us an upper bound on the error between $P(A_O)$ and C:

$$\rho\left(P\left(A_O\right), C\right) \leq \rho\left(P\left(A_O\right), \operatorname{conv}\left(V_I\right)\right) \quad , \tag{5}$$

which holds because $\operatorname{conv}(V_I) \subseteq C$.

Since both the inner and outer approximations are available as the DD pairs (V_I, A_I) and (A_O, V_O) respectively, it is relatively simple to compute the Hausdorff distance between them[1].

$$\rho\left(P\left(A_O\right), \operatorname{conv}\left(V_I\right)\right) = \max_{v \in V_O} \min_{x \in P(A_I)} \|x - y\|_2 \tag{6}$$

Equation 6 requires the solution of one QP of size $|A_I|$ per vertex of the outer approximation V_O. In each iteration of the algorithm, the majority of these QPs will not change since the double description method modifies the inner and outer approximations only locally. Those that do require re-computation are exactly those that depend on the new vertices V^{new} created in Lemma 1.

B. Improvement of the Inner Approximation

All polytopes can be expressed either as the convex combination of their vertices, or as the intersection of a finite number of halfspaces. This duality has lead to a number of algorithms that can operate on both representations equally well, and the double description algorithm is one such. The dual version is generally called the Beneath/Beyond algorithm and takes as input a finite set of points and returns the list of halfspaces representing the convex hull [15], [16].

Lemma 3 gives a useful and well-known result which allows the double-description algorithm to be used to compute an inequality description of a polytope as readily as it computes a vertex representation.

Lemma 3 (e.g. [4]): The finite sets $A, V \subset \mathbb{R}^d$ form a DD pair (A, V) if and only if (V, A) is a DD pair.

This basic duality result can be used in order to augment the double description algorithm of the previous section, which computes outer approximations, in order to calculate an inner approximation by reversing the roles of vertices and halfspaces in the approach. In other words, assume (A_I, V_I) is a DD pair representing the polytope $P(A_I) = \operatorname{conv}(V_I)$ and we wish to compute the set A_I' so that $(A_I', V_I \cup \{v\})$ is a DD pair for some v. The double description Lemma 1 can be used for this purpose by simply passing it the DD pair $(V_I \cup \{v\}, A_I')$.

We can now make use of the double description mechanism in order to iteratively construct an inner approximation of the set by inserting one extreme point $v^\star$ of C at a time. The choice of the point $v^\star$ to insert in each iteration of the algorithm is made in an analogous fashion to the previous section. Instead of computing the maximal separating halfspace for each vertex v of V_O, we compute the extreme point $v^\star$ of C that is a maximal distance from each hyperplane of the inner approximation using the `extr` function.

[1] Private communication with S.V. Raković.

Remark 2: The approach presented here for computing inner approximations is similar to that in [6] where we proposed an implicit approach for polyhedral projection based on the beneath/beyond procedure. We here extend this method to the computation of simultaneous inner and outer polytopic approximations for generic convex and compact sets. This simultaneous inner/outer approximation also provides the significant benefit of much simpler calculation of the current approximation error, as was discussed in the previous section.

The proposed method is shown as Algorithm 2. One can see that each iteration involves one improvement of the outer and one of the inner approximation (Lines 3 to 8 and 10 to 15 respectively). For the outer improvement, the algorithm first approximates the Hausdorff distance between $P(A_O)$ and C using (5) or (6) in order to select a vertex $v \in V_O$ to 'cut off' from the polytope. It then computes the hyperplane $a^\star$ that maximally separates v from C on line 5, which also gives the true distance between v and C as $\langle a^\star, v\rangle - \delta^\star(a^\star \mid C)$. If this distance is larger than the desired approximation, then the DD pair is updated to $(A_O \cup \{a^\star\}, V_O')$ using Lemma 1. These steps are then repeated on the inner approximation DD pair (V_I, A_I) until the approximation error is below that desired.

Algorithm 2 Implicit Double Description Method

Require: The functions `maxsep` and `extr` for some convex and compact set C and a desired approximation error $\varepsilon > 0$.

Ensure: DD pairs (A_O, V_O) and (A_I, V_I) such that $\operatorname{conv}(V_I) \subseteq C \subseteq P(A_O)$ and $\rho\left(P\left(A_O\right), \operatorname{conv}\left(V_I\right)\right) \leq \varepsilon$.

1: Obtain DD pairs (A_O, V_O) and (V_I, A_I), such that $\operatorname{conv}(V_I) \subseteq C \subseteq P(A_O)$
2: **while** $\rho\left(P\left(A_O\right), \operatorname{conv}\left(V_I\right)\right) \geq \varepsilon$ **do** *Equation 6*
3: // *Improve outer approximation*
4: Compute $v \in V_O$ farthest from $P(A_I)$ *§IV-A.1*
5: Separate v from C : $a^\star := \texttt{maxsep}(v \mid C)$
6: **if** $\rho(v, C) > \varepsilon$ **then**
7: $A_O := A_O \cup \{a^\star\}$
8: Compute V_O s.t. (A_O, V_O) is a DD pair *Lemma 1*
9: **end if**
10: // *Improve inner approximation*
11: Compute $a \in A_I$ farthest from $\operatorname{conv}(V_O)$ *§IV-A.1*
12: Compute point $v^\star$ beyond a : $v^\star := \texttt{extr}(a \mid C)$
13: **if** $\rho\left(v^\star, \operatorname{conv}(V_I)\right) > \varepsilon$ **then**
14: $V_I := V_I \cup \{v^\star\}$
15: Compute A_I s.t. (V_I, A_I) is a DD pair *Lemma 1*
16: **end if**
17: **end while**

V. Application to Model Predictive Control

The recent interest in parametric programming in the control community has arisen from the ability to pose certain optimal control problems as parametric programs and thereby pre–compute the optimal control law offline. In this paper, we are specifically interested in the following standard semi-infinite horizon optimal control problem for a linear time-invariant system:

$$
\begin{aligned}
J^{\star}(x) = \min_{\{u_0,\ldots,u_{N-1}\}} \; & V_N(x_N) + \sum_{i=0}^{N-1} l(x_i,u_i) \qquad (7)\\
\text{s. t.} \quad & x_{i+1} = Ax_i + Bu_i, \quad \forall i = 0,\ldots,N-1\\
& (x_i,u_i) \in \mathscr{X} \times \mathscr{U}, \quad \forall i = 0,\ldots,N-1\\
& x_N \in \mathscr{X}_F,\\
& x_0 = x
\end{aligned}
$$

where $\mathscr{X}$, $\mathscr{U}$ and $\mathscr{X}_F$ are polytopic constraints on the states and inputs and the stage cost l is defined as $l(x_i,u_i) := \|Qx_i\|_p + \|Ru_i\|_p$, for some weighting matrix R and Q of appropriate size. Under the standard assumptions that $\mathscr{X}_F \subseteq \mathscr{X}$ is an invariant set, V_N is a Lyapunov function and that the decay rate of V_N is greater than the stage cost within the set $\mathscr{X}_F$, the problem (7) generates a stabilizing control law when applied in a receding horizon fashion [17]. If the norm is taken to be the one or infinity norm, then we get a pLP of the form:

$$u^{\star}(x) = \operatorname*{arg\,min}_{u} \left\{ g^T u \,|\, Gu \le Fx + w \right\} \ , \qquad (8)$$

where u is a vector containing the sequence of inputs $u_0,\ldots,u_{N-1}$ and appropriate slack variables. The system input is then given in a receding horizon fashion by $\kappa(u) := u_0$. See [18] for details on the computation of the matricies G, F and the vectors g and w for the MPC problem (7).

'Solving' the pLP (7) means to compute the optimizer $u^{\star}(x)$ of pLP (8) for every feasible value of the state. In this section the goal is to compute an approximate solution $\tilde{u}(x)$ that maps from the measured state to a feasible solution of pLP (8). We do this by approximating the epigraph of the optimal cost function $J^{\star}(x)$, which is a polyhedron defined implicitly through a projection operation [19]

$$\operatorname{epi}(J^{\star}) = \left\{ (x,J) \in \mathbb{R}^d \times \mathbb{R} \,\middle|\, \exists u, Gu \le Fx + w, J \ge g^T u \right\} \ . \qquad (9)$$

The method proposed in this paper can be used to compute an upper or lower approximate polyhedral cost function $\tilde{J}$ within a given error bound of the optimal epigraph $\rho\left(\operatorname{epi}(\tilde{J}), \operatorname{epi}(J^{\star})\right) \le \varepsilon$.

Approximating the optimal epigraph has been proposed in the literature, and conditions have been given for which both inner [20] and outer [8], [9] polyhedral approximations of the optimal cost function $J^{\star}(x)$ are also Lyapunov functions for the system under the control law $\kappa(\tilde{u}(x))$, where $\tilde{J}(x) = g^T \tilde{u}(x)$, for some feasible $\tilde{u}(x)$. These conditions can be reduced to essentially a minimum weighted Hausdorff distance requirement between the epigraph of the approximate cost function $\tilde{J}$ and the optimal $J^{\star}$. Due to space limitations we do not re-state these conditions here, but instead direct the interested reader to the aforementioned references.

Remark 3: Note that because the epigraph of the optimal $J^{\star}$ and approximate $\tilde{J}$ cost functions is unbounded, the methods in this paper cannot be applied directly. The reader is refered to [6] for details on how to *homogenize* and bound the epigraphs converting the problem to a polytopic approximation.

The application of the approach proposed in this paper to approximate explicit MPC for linear norms requires the definition of the two functions `maxsep` and `extr` that operate on implicitly defined polytopes such as epi $J^{\star}$. If $S := \{x \,|\, \exists u,\ Dx + Eu \le b\}$ is a polytope, then a linear program that computes the function `extr` can be written directly from the definition (2)

$$\texttt{extr}(a \mid S) := \operatorname*{argmax}_{x,u} \left\{ a^T x \,|\, Dx + Eu \le b \right\} \ . \qquad (10)$$

The function $\texttt{maxsep}(v \mid S)$ is to return a vector $a^{\star}$ such that $P(a^{\star})$ contains S and is a maximal distance from the point v. The following lemma allows us to describe the set of halfspaces that contains the polytope S.

Lemma 4 (Projection Lemma [21], [22]): Let $S := \{x \,|\, \exists u,\ Dx + Eu \le b\}$ be a polyhedron. Then the halfspace $\alpha^T x \le \beta$ contains S if and only if there exists a positive vector λ such that

$$\alpha = D^T \lambda, \qquad \beta = b^T \lambda, \qquad E^T \lambda = 0 \ .$$

We follow an approach similar to that used for support vector machines in order to compute the maximal separating hyperplane by solving a quadratic program.

Theorem 2: Let $S := \{x \,|\, \exists u,\ Dx + Eu \le b\}$ be an implicitly defined polytope and $v \notin S$ be a vector. Then $\texttt{maxsep}(v \mid S) = D^T \lambda^{\star} / (b^T \lambda^{\star})$, where $\lambda^{\star}$ is the optimizer of the following quadratic program

$$
\begin{aligned}
\lambda^{\star} := \ & \operatorname*{argmin}_{\lambda} \ a^T a\\
& \text{subject to} \quad \lambda \ge 0\\
& \qquad D^T \lambda = a\\
& \qquad b^T \lambda = a^T v - 1\\
& \qquad E^T \lambda = 0
\end{aligned}
$$

Proof: We define two halfspaces $h_0 := \left\{ x \,\middle|\, a^T x \le c \right\}$ and $h_1 := \left\{ x \,\middle|\, a^T x \ge c + 1 \right\}$ such that h_0 contains S and h_1 contains v. The goal can then be re-stated as maximizing the distance between h_0 and h_1. If x_0 is the point in h_0 that is closest to v, then we can write $v - x_0 = a \|v - x_0\|_2 / \|a\|_2$. Taking the inner product of $v - x_0$ and a gives $a^T v - a^T x_0 = c + 1 - c = 1 = \|v - x_0\|_2 a^T a / \|a\|_2 = \|v - x_0\|_2 \|a\|_2$, which shows that the distance $\|v - x_0\|_2$ is equal to $1/\|a\|_2$ and therefore maximizing $\|v - x_0\|_2$ is equivalent to minimizing the norm of a. The constraints given in the QP then simply require that $v \in h_1$ and, from Lemma 4, that h_0 contains S.

Finally, we have that $P(a^{\star}) = h_0 = \left\{ x \,\middle|\, D^T \lambda^{\star} x \le b^T \lambda^{\star} \right\} = \left\{ x \,\middle|\, D^T \lambda^{\star} / (b^T \lambda^{\star}) x \le 1 \right\}$. ■

A. Recovery of Control Input : Barycentric Coordinates

In the previous section we discussed how an approximate cost function $\tilde{J}$ for the parametric problem can be computed using the convex set approximation method proposed in this paper. However, from the MPC point of view, the point of approximating a pLP is to calculate a control law. In this section we will discuss a new method of recovering such an approximate control law from a piecewise affine sub-optimal cost function.

Let $\tilde{J}$ be the piecewise affine function

$$\tilde{J}(x) := b_i x + c_i, \quad \text{if } x \in R_i \tag{11}$$

where the polytopes R_i form a partition: $\mathscr{R} = \cup R_i$ is convex and $\operatorname{int} R_i \cap \operatorname{int} R_j = \emptyset$ for all $i \neq j$. Our goal is to find any function $\tilde{u}(x)$ such that

$$g^T \tilde{u}(x) = \tilde{J}(x) \tag{12a}$$

$$G\tilde{u}(x) \le Fx + w \qquad \text{for all } x \in \mathscr{R} \tag{12b}$$

The authors are aware of three proposals in the literature to tackle the problem of computing a function $\tilde{u}(\cdot)$ from an approximate cost $\tilde{J}(\cdot)$, all of which potentially generate an approximate control law that is significantly more complex than the approximate cost function. The first is simply to compute a tesselation of each polytopic region R_i. One can then interpolate uniquely amongst the vertices of each simplical region of the tesselation, which results in a feasible piecewise affine function [6], [7]. While this approach is easily stated and implemented, it has a significant downside in that such a tesselation can have exponentially more simplices than there were regions R_i. In [7] it was suggested that an affine function be fit in a least-squares fashion to the optimizers $u^\star(v)$ at the vertices v of each region R_i. However, if a region R_i is not a simplex, then there is no guarantee that the fitted function will be everywhere feasible. The third approach [8], [9] computes an approximate cost for the optimal control problem (7) in a recursive fashion. After a sufficient number of iterations, the approximate cost function is used as a 'cost-to-go' while the exact solution is computed in the last phase, which then provides the approximate control law. However, this last exact iteration can contain a much larger number of regions than the approximate cost function.

In this section we propose a new method of computing a feasible control law based on Barycentric coordinates, which does not generate any new regions.

Definition 3 (Barycentric function): Let $S := \operatorname{conv}(\{v_1, \dots, v_n\}) \subset \mathbb{R}^d$ be a polytope. The function $w(x\,|\,v)$ is called *barycentric* if three conditions hold for all $x \in S$ and $v \in \operatorname{extr}(S)$

$$w(x\,|\,v) \ge 0 \ , \qquad \textit{positivity} \tag{13a}$$

$$\sum_{v \in \operatorname{extr}(S)} w(x\,|\,v) = 1 \ , \qquad \textit{partition of unity} \tag{13b}$$

$$\sum_{v \in \operatorname{extr}(S)} v w(x\,|\,v) = x \ , \qquad \textit{linear precision} \tag{13c}$$

For each vertex $v \in \operatorname{extr}(R_i)$ and region R_i, we assume the availability of a vector u_v that is feasible for pLP (8) and equals the approximate cost function at v.

$$g^T u_v = \tilde{J}(v) \ , \qquad G u_v \le F v + w \ .$$

Such points u_v can be computed by solving a series of linear programs. If a barycentric function w_i is available for each region R_i in (12), then we can define an approximate control law $\tilde{u}(x)$ by interpolating amongst these points.

$$\tilde{u}(x) := \sum_{v \in \operatorname{extr}(R_i)} u_v w_i(x\,|\,v) \ , \quad \text{if } x \in R_i \ . \tag{14}$$

The following theorem demonstrates that if such a barycentric function is available, then $\tilde{u}(x)$ satisfies conditions (12) and therefore is a feasible control law.

Theorem 3: If $\tilde{J}$ is the piecewise affine function defined in (11) and $\tilde{u}$ is as defined in (14), then $\tilde{u}$ and $\tilde{J}$ satisfy (12).

Proof: Let S be the polytopic constraints of the pLP (8). By definition, $(v, u_v) \in S$ for all $v \in \operatorname{extr}(R_i)$ and all R_i. For any $x \in R_i$, we have that $(x, \tilde{u}(x)) = \sum_v (v, u_v) w_i(x\,|\,v)$ is in S by convexity, since $w_i(x\,|\,v) \ge 0$ and $\sum_v w_i(x\,|\,v) = 1$.

We now show that $g^T \tilde{u}(x) = \tilde{J}(x)$:

$$\begin{aligned} g^T \tilde{u}(x) &= g^T \sum_v u_v w_i(x\,|\,v) \ , \qquad x \in R_i \\ &= \sum_v (b_i v + c_i) w_i(x\,|\,v) \ , \end{aligned}$$

where $\tilde{J}(x) = b_i x + c_i$ for $x \in R_i$ (11) and by definition $g^T u_v = \tilde{J}(v)$.

$$\begin{aligned} g^T \tilde{u}(x) &= b_i (\sum_v v w_i(x\,|\,v)) + c_i (\sum_v w_i(x\,|\,v)) \\ &= b_i x + c_i \\ &= \tilde{J}(x) \ , \end{aligned} \tag{15}$$

where (15) follows due to the linear precision and the unity properties of barycentric coordinates (13). ■

From Theorem 3 we can see that although the barycentric coordinates define the approximate optimizer $\tilde{u}(x)$ through a non-linear interpolation, the resulting cost function is still piecewise affine.

Our goal is now to define an easily computable barycentric function for each polytope R_i in (11). If the polytope R_i is a simplex, then the barycentric function is unique, linear and trivially computed and so we focus on the non-simplical case. In [23] a very elegant method of computing a barycentric function for arbitrary polytopes was proposed that can be put to use here.

Lemma 5 (Barycentric coordinates for polytopes [23]): Let $S = \operatorname{conv}(V) \subset \mathbb{R}^d$ be a polytope and for each simple vertex v of S, let $b_v(x)$ be the function

$$b_v(x) = \frac{\alpha_v}{\|v - x\|_2}$$

where α_v is the area of the polytope $P(V - \{x\}) \cap \{y \,|\, \langle v - x, y \rangle = 1\}$; i.e. the area of the facet of the polar dual of $S - \{x\}$ corresponding to the vertex $v - x$. The function $w_v(x) := b_v(x) / \sum_v b_v(x)$ is barycentric over the polytope S.

Proof: We provide here a brief sketch of the proof and refer the reader to [23] for details.

The proof is based on Stokes theorem, which states that the surface integral over a compact set is zero. Consider the surface integral of the polar dual of S, the polytope

$P(V - \{x\})$

$$\oint_{P(V-\{x\})} y dy = \sum_i \alpha_i n_i = \sum_i \alpha_i \frac{v_i - x}{\|v_i - x\|_2} = 0 \ , \qquad (16)$$

where n_i is the outward facing normal to the i^{th} facet of the polytope and α_i is the area of the facet. From the definition of the polar dual, the normal of the i^{th} facet is proportional to $v_i - x$ [24]. With some minor algebraic manipulation, (16) leads directly to the theorem statement. ■

The areas of the facets of the polar duals α_v can be pre-computed offline and stored. If there are $d+1$ facets incident with the vertex v (i.e. v is simplical), then the area of the polar facet is $\det\left(\begin{bmatrix} a_0 & \cdots & a_{d+1} \end{bmatrix}\right)$, where $\{a_0, \ldots, a_{d+1}\}$ are the normals of the incident facets. If the vertex is not simplical, then the area can be easily computed by perturbing the incident facets [23]. Such computation is straightforward because both the vertices and halfspaces of each region are available due the double-description representation.

VI. Example

Consider the following four–state system:

$$x^+ = \begin{bmatrix} 0.7 & -0.1 & 0.0 & 0.0 \\ 0.2 & -0.5 & 0.1 & 0.0 \\ 0.0 & 0.1 & 0.1 & 0.0 \\ 0.5 & 0.0 & 0.5 & 0.5 \end{bmatrix} x + \begin{bmatrix} 0.0 & 0.1 \\ 0.1 & 1.0 \\ 0.1 & 0.0 \\ 0.0 & 0.0 \end{bmatrix} u$$

States and control inputs are constrained $\|x\|_\infty \leq 5$, $\|u\|_\infty \leq 5$ and we seek to solve the MPC problem (7) minimizing the infinity norm where Q and R are the identity and the prediction horizon is $N = 5$. Figure 1 shows a plot of complexity (number of regions) vs the approximation error for the proposed method. Figure 2 shows a time trajectory of the closed loop system at various complexities ranging from 7 to 182 regions, which is significantly lower than the optimal explicit control law, which consists of $12,128$ regions.

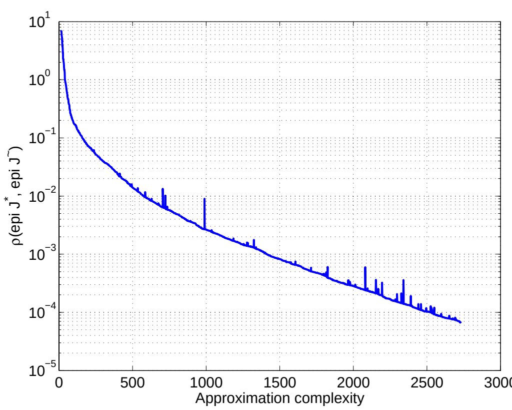

Fig. 1. Approximation error of the four-state system of Example VI vs the approximation complexity (number of polyhedral regions in the PWA cost function). The optimal solution consists of $12,128$ regions. (The noise in the plot is due to numerical errors.)

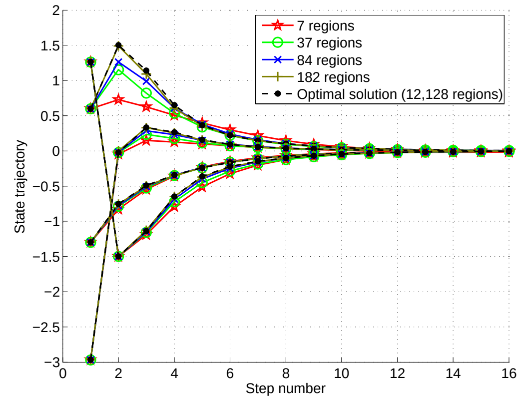

Fig. 2. Example trajectory of the approximate and optimal solutions of Example VI for various approximation levels.

References

[1] M. M. Seron, G. C. Goodwin, and J. A. De Doná, "Geometry of model predictive control for constrained linear systems," The University of Newcastle, Australia, Tech. Rep. EE0031, 2000.

[2] T. A. Johansen, I. Petersen, and O. Slupphaug, "On explicit suboptimal LQR with state and input constraints," in *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, 2000, pp. 662–667.

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

[4] K. Fukuda and A. Prodon, "Double description method revisited," in *Combinatorics and Computer Science*, ser. Lecture Notes in Computer Science, M. Deza, R. Euler, and I. Manoussakis, Eds. Springer-Verlag, 1996, vol. 1120, pp. 91–111.

[5] T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall, *The double description method*. Princeton: Princeton Uni. Press, 1953.

[6] C. N. Jones, M. Barić, and M. Morari, "Multiparametric linear programming with applications to control," *European Journal of Control*, vol. 13, no. 2-3, pp. 152–170, 2007.

[7] A. Bemporad and C. Filippi, "An algorithm for approximate multiparametric convex programming," *Computational Optimization and Applications*, vol. V35, no. 1, pp. 87–108, 2006.

[8] B. Lincoln and A. Rantzer, "Relaxing dynamic programming," *IEEE Trans. on Automatic Control*, vol. 51, no. 8, pp. 1249–1260, 2006.

[9] J. Bjornberg and M. Diehl, "Approximate robust dynamic programming and robustly stable mpc," *Automatica*, vol. 42, no. 5, 2006.

[10] T. A. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal search tree," *IEEE Trans. Automatic Control*, vol. 48, pp. 810–815, 2003.

[11] R. Schneider and J. A. Wieacker, "Approximation of convex bodies by polytopes," *Bulletin London Mathematical Society*, vol. 13, no. 2, pp. 149–156, 1981.

[12] E. M. Bronshteyn and L. D. Ivanov, "The approximation of convex sets by polyhedra," *Siberian Mathematical Journal*, vol. 16, no. 5, pp. 852–853, 1975.

[13] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1970.

[14] J. S. Mitchell and S. Suri, "Separation and approximation of polyhedral objects," *Computational Geometry*, vol. 5, no. 2, pp. 95–114, 1995.

[15] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.

[16] M. Kallay, "Convex hull algorithms for higher dimensions," 1981.

[17] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[18] F. Borrelli, A. Bemporad, and M. Morari, "A Geometric Algorithm for Multi-Parametric Linear Programming," *Journal of Optimization Theory and Applications*, vol. 118, no. 3, pp. 515–540, 2003.

[19] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski, "On polyhedral projection and parametric programming," *Journal of Optimization Theory and Applications*, vol. 137, no. 3, 2008.

[20] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)," *Automatic Control, IEEE Transactions on*, vol. 44, no. 3, pp. 648–654, 1999.

[21] S. N. Černikov, "Contraction of finite systems of linear inequalities (in russian)," *Doklady Akademiia Nauk SSSR*, vol. 152, no. 5, pp. 1075–1078, 1963, (English translation in *Societ Mathematics Doklady*, Vol. 4, No. 5 (1963), pp.1520-1524).

[22] G. M. Ziegler, *Lectures on Polytopes*. Springer-Verlag, 1995.

[23] J. Warren, S. Schaefer, A. Hirani, and M. Desbrun, "Barycentric coordinates for convex sets," *Advances in Computational Mathematics*, vol. 27, no. 3, pp. 319–338, 2007.

[24] B. Grünbaum, *Convex Polytopes*, 2nd ed. Springer-Verlag, 2000.