Polytopic Approximation of Explicit Model Predictive Controllers

Colin N. Jones *Member, IEEE,* and Manfred Morari *Fellow, IEEE,*

***Abstract*—A model predictive control law (MPC) is given by the solution to a parametric optimization problem that can be pre-computed offline, which provides an explicit map from state to input that can be rapidly evaluated online. However, the primary limitations of these optimal 'explicit solutions' are that they are applicable to only a restricted set of systems and that the complexity can grow quickly with problem size. In this paper we compute approximate explicit control laws that trade-off complexity against approximation error for MPC controllers that give rise to convex parametric optimization problems.**

The algorithm is based on the classic double-description method and returns a polyhedral approximation to the optimal cost function. The proposed method has three main advantages from a control point of view: it is an incremental approach, meaning that an approximation of any specified complexity can be produced, it operates on implicitly-defined convex sets, meaning that the prohibitively complex optimal explicit solution is not required and finally it can be applied to any convex parametric optimization problem.

A sub-optimal controller based on barycentric interpolation is then generated from this approximate polyhedral cost function that is feasible and stabilizing. 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, which is demonstrated on several examples.

I. Introduction

Implementing a model predictive controller (MPC) 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 x as the parameter

$$J^\star(x) := \min \{h(x,u) \mid g(x,u) \leq 0\} \quad . \qquad (1)$$

In this paper we restrict our attention to those problems in which $J^\star$, h and g are convex functions. Solving this parametric problem off–line results in an explicit function $u^\star(x)$ mapping the measured state to the optimal system input [8], [18], [33]. 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 several orders of magnitude for some systems.

There are two main limitations of this approach. The first is that only a restricted class of systems give rise to problems that can be reasonably solved parametrically. For example, linear systems with piecewise linear [5], [30] or quadratic cost functions [8], piecewise affine systems (PWA) with linear [4] and quadratic [7] cost functions and some classes of polynomial systems [3]. If the control law can be computed explicitly, then it is often the case that the complexity of the control law (i.e. the number of 'pieces') grows quickly with problem size. The reader is referred to the recent surveys [1], [20] for an overview.

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 $\mathrm{epi}\,(J^\star)$. The approach is based on the well established double description method [13], [27], 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

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

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. However if a certificate of stability is desired, then there is a minimum level of complexity required, although this level is normally quite small.

Several authors have proposed approximation algorithms that produce simpler PWA control laws at the cost of optimality. As is the case in this paper, these approaches operate in a two–stage procedure: the epigraph of the optimal cost function $J^\star$ is approximated with a polyhedron $\tilde{J}$, which will determine the stability and performance properties of the approximate controller and then a feasible control law $\tilde{u}(x)$ is computed such that $\tilde{J}$ is a Lyapunov function for the resulting closed-loop system. However, generating such a feasible control law $\tilde{u}$ is not immediate given the polyhedral function $\tilde{J}$ and so existing proposals either produce a triangulation and then interpolate the optimal control law at the vertices [6], [20], have a post-processing step in which an exact parametric program is calculated based on the approximate cost [10], [23] or compute control laws based on sub-divisions of hypercubes [19]. 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 $\tilde{u}$ from any polyhedral approximate cost. The main benefit is that we do not have to restrict ourselves to considering approximation approaches that generate triangulations, and hence can directly compute a control law $\tilde{u}$ for the non–simplical regions produced by the double–description algorithm, which 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 studies the application of these tools to 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 $\{x \mid \langle a, x\rangle \leq 1,\ \forall a \in A\}$, which is a polyhedron if A is finite. If V is a subset of $\mathbb{R}^d$, then the convex hull of V, $\mathrm{conv}(V)$ is the intersection of all convex sets containing V. If $V = \{v_0, \ldots, v_n\}$ is a finite set, then $\mathrm{conv}(V) = \{\sum_{i=0}^{n} v_i\lambda_i \mid \lambda_i \geq 0,\ \sum \lambda_i = 1\}$.

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 $\rho(S, C) \leq \epsilon$, where $\rho(\cdot, \cdot)$ is the Hausdorff distance. S is called an outer (inner) $\epsilon-$approximation if $C \subseteq S$ ($S \subseteq C$).

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

Theorem 2 ([12], [31]): If $C \subset \mathbb{R}^d$ is a convex and compact set, then for every $\epsilon > 0$, there exists a finitely generated polytope S such that $\rho(S, C) < \epsilon$.

The 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 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

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

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: $\langle a, x \rangle \geq 1$ and $C \subset \{ x \mid \langle a, x \rangle \leq 1 \}$.

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

Remark 3: A maximally separating hyperplane for a point x can be determined by finding the closest point $x \in C$ to v in the 2-norm $v^\star \in \operatorname{argmin}_x \{ (v - x)^T (v - x) \mid x \in C \}$. The normal of the separating hyperplane is then given by the scaled tangent of the norm-ball at the point $v^\star$, $a = \frac{x - v^\star}{\langle v^\star, x - v^\star \rangle}$.

Remark 4: Note that the above optimization problems are convex, since the set C is assumed to be convex. For example, if C is a polytope, then computing `extr` requires a linear program and `maxsep` a quadratic. If C contains linear and quadratic constraints, as is the case for MPC problems with a quadratic cost and linear constraints (Section V), then both `extr` and `maxsep` require the solution of a second order cone problem.

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 $\epsilon-$approximation of an implicitly defined convex and compact set.

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 5 ([13], [27]): 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 $\mathcal{A}$ and the goal is to compute all vertices of $P(\mathcal{A})$. This is accomplished in an incremental fashion, beginning with a small subset $A \subset \mathcal{A}$ 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 $a \in \mathcal{A}$, or equivalently by intersecting the polytope $P(A)$ with the halfspace $\{ x \mid \langle a, x \rangle \leq 1 \}$ and the set of vertices V is updated so that (A', V') remains a DD pair. This procedure continues until all of $\mathcal{A}$ has been inserted, at which point we have the DD pair $(\mathcal{A}, V)$ and therefore all vertices V of the polytope $P(\mathcal{A})$.

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 6 (DD Lemma [13]): 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 $V' = V^+ \cup V^= \cup V^{\text{new}}$

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

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 6 in hand, we can now state the double description method as shown in Algorithm 1. The procedure is depicted in Figure 1, where one can see the insertion of one new halfspace into an existing hypercube.

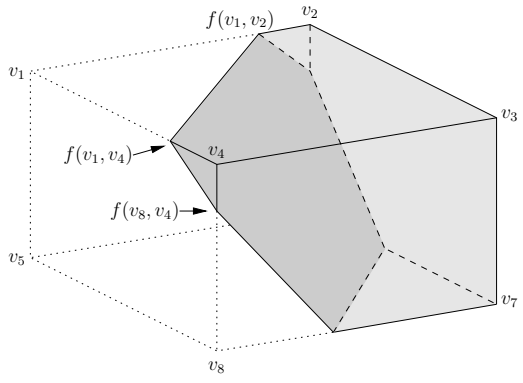


Fig. 1. Illustration of a single iteration of the Double Description algorithm.

Algorithm 1 Classic Double Description Method

Require: A finite set $\mathcal{A} := \{a_1, \ldots, a_N\} \subset \mathbb{R}^d$, such that $\dim P(\mathcal{A}) = d$
Ensure: A minimal set $V \subset \mathbb{R}^d$, such that $P(\mathcal{A}) = \text{conv}(V)$
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 6
5: $K := K \cup \{j\}$, $V := V'$
6: **end while**

Remark 7: The first step of Algorithm 1 is to choose a set K such that the list of vertices of $P(\{a_i \mid i \in K\})$ can be easily computed, which is achieved by selecting exactly $d+1$ elements a_i such that $P(\{a_i \mid i \in K\})$ is a simplex.

Remark 8: The double description algorithm has been extensively studied since its inception in 1953 [27] and improvements have been made both for the practical and the worst-case complexity over the simple algorithm in Table 1. In order to achieve these improvements, the set $\mathcal{A}$ must be analyzed in a pre-processing phase. As will be seen in the following sections, the method proposed in this paper constructs the set $\mathcal{A}$ as it runs and so cannot do this pre-processing. As a result, the worst-case complexity (the maximum number of vertices over all iterations) will match that of the original algorithm and will be doubly exponential in the size of $\mathcal{A}$, although this is rarely seen in practice.

Remark 9: Clearly, an efficient implementation of the algorithm requires that the set V^{new} in Lemma 6 be rapidly calculable. This calculation can be done in time linear in the size of V^{new} and is not a function of the size of the sets A or V [15].

Remark 10: Lemma 6 requires that the adjacency relationships of the vertices be known. These can easily and directly be determined from the incidence map of the polyhedron (which halfspaces contain which vertices). See [13] for details.

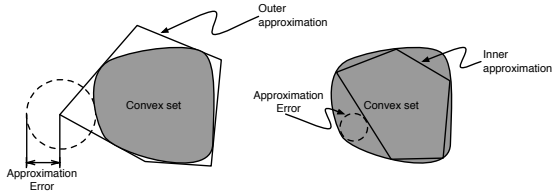


Fig. 2. Inner and outer polytopic approximations of a convex set.

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 [29, Thm. 11.5]; $C = P(\mathcal{A}) = \operatorname{conv}(\mathcal{V})$ for some sets $\mathcal{A}$ and $\mathcal{V} \subset \mathbb{R}^d$. Computing a polytopic outer $\epsilon-$approximation can then be stated as finding a finite subset A of $\mathcal{A}$ such that $\rho(P(A), C) \leq \epsilon$. Equivalently, an inner approximation consists of a finite subset $V \subseteq \mathcal{V}$ such that $\rho(\operatorname{conv}(V), C) \leq \epsilon$. Figure 2 illustrates the proposed inner and outer polytopic approximations.

The ideal would be to determine a set $A \subset \mathcal{A}$ 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 $\mathcal{A}$ is finite and known [26]. 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 $\operatorname{conv}(V_I)$ is an inner $\hat{\epsilon}-$approximation of C and $P(A_O)$ an outer for some $\hat{\epsilon} > \epsilon$. 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 $\operatorname{conv}(V_I)$. The next section demonstrates how we choose an element of $\mathcal{A}$ 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 $a^\star \in \mathcal{A}$ to decrease the approximation error

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

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 $\hat{\epsilon}$ is given by the Hausdorff distance between $P(A_O)$ and C

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

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. By assumption we cannot do direct computations on C, but can only evaluate the support function, which leads us to the following lemma.

Lemma 11: 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

$$\rho(S, C)^2 = \max \left\{ \frac{\langle a,\, v \rangle - 1}{\|a\|_2} \;\middle|\; \begin{array}{l} v \in \operatorname{extreme}(S) \\ a = \texttt{maxsep}(v \mid C) \end{array} \right\}$$

where $\operatorname{extreme}(S)$ are the vertices of S.

Proof: The Hausdorff distance is given by $\rho(C, S) = \max\{q(y) \mid y \in S\}$, where $q(y) := \min\{\|x - y\|_2 \mid x \in C\}$. The function $q(\cdot)$ is convex and therefore the maximum is obtained at an extreme point of S [29, Thm. 32.2]; $\rho(C, S) = \max\{q(v) \mid v \in \operatorname{extreme}(S)\}$. For a given extreme point $v \in \operatorname{extreme}(S)$, the minimum distance $q(v)$ is given by the `maxsep` function (3). ■

From Lemma 11 one can see that the Hausdorff distance $\rho(C, P(A_O))$ is equal to $\rho(C, V_O)$ and can therefore be computed through a finite number of evaluations of the `maxsep` function applied to each element of V_O.

Remark 12: Because the vertices V_O are computed in an incremental fashion, it is not necessary to evaluate `maxsep` in Lemma 11 for each v in V_O in each iteration, but only those newly created in Lemma 6, V^{new}.

With Lemma 11 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 11 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 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:

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

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.

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

Equation 6 requires the solution of one quadratic program (QP) of size $|A_I|$ per vertex of the outer approximation. In each iteration of the algorithm, the majority of these QPs will not change since the 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 6.

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 led 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 [2], [21].

Lemma 13 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 13 (e.g. [13]): 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 6 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.

Remark 14: The approach presented here for computing inner approximations is similar to that in [20]

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 6. 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 $\epsilon > 0$.

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

1: Obtain DD pairs (A_O, V_O) and (V_I, A_I), such that $\mathrm{conv}(V_I) \subseteq C \subseteq P(A_O)$
2: **while** $\rho(P(A_O), \mathrm{conv}(V_I)) \geq \epsilon$ **do** *Eqn 6*
3: // *Improve outer approximation*
4: Compute $v \in V_O$ farthest from $P(A_I)$ *§IV-A1*
5: Separate v from C : $a^\star := \texttt{maxsep}(v \mid C)$
6: **if** $\rho(v, C) > \epsilon$ **then**
7: $A_O := A_O \cup \{a^\star\}$
8: Compute V_O s.t. (A_O, V_O) is a DD pair *Lemma 6*
9: **end if**
10: // *Improve inner approximation*
11: Compute $a \in A_I$ farthest from $\mathrm{conv}(V_O)$ *§IV-A1*
12: Compute point $v^\star$ beyond a : $v^\star := \texttt{extr}(a \mid C)$
13: **if** $\rho(v^\star, \mathrm{conv}(V_I)) > \epsilon$ **then**
14: $V_I := V_I \cup \{v^\star\}$
15: Compute A_I s.t. (V_I, A_I) is a DD pair *Lemma 6*
16: **end if**
17: **end while**

V. Application to Model Predictive Control

The 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 finite horizon optimal control problem:

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

where

$$J(u_0,\ldots,u_{N-1},x_0,\ldots,x_N) := V_N(x_N) + \sum_{i=0}^{N-1} l(x_i,u_i) \tag{8}$$

and $\mathcal{X}$, $\mathcal{U}$ and $\mathcal{X}_F$ are convex constraints on the states and inputs and the stage cost l is a strictly convex function with $l(0,0) = 0$. A function $\gamma(\cdot) : \mathbb{R} \to \mathbb{R}$ is assumed to exist that is continuous, strictly increasing and has $\gamma(0) = 0$[1] such that $\gamma(\|x\|) \leq l(x,0)$ for all x. Under the standard assumptions that $\mathcal{X}_F \subseteq \mathcal{X}$ is an invariant set under the control law $\mu(x)$, V_N is a Lyapunov function for the system $x^+ = Ax + B\mu(x)$ and that the decay rate of V_N is greater than the stage cost within

[1] i.e. γ is a K-function

the set $\mathcal{X}_F$, the problem (7) generates a stabilizing control law when applied in a receding horizon fashion [25]. The optimal control problem (7) can be re-written as a parametric optimization problem:

$$u^\star(x) := \arg\min_u \{h(x,u) \mid g(x,u) \leq 0\} \quad (9)$$

where u is a vector containing the sequence of inputs $u_0, \ldots, u_{N-1}$ and appropriate auxiliary variables and the functions h and g are convex. The system input is then given in a receding horizon fashion by $u_0^\star(x)$, which is the first input in the optimal control sequence of (7). See [1] for a survey of papers providing details on the conversion from the optimal control formulation (7) to the parametric optimization problem (9) for some important classes of systems.

Let $\tilde{J} : \mathcal{R} \mapsto \mathbb{R}$ be a piecewise affine function, where $\mathcal{R}$ is a polytopic subset of $\mathbb{X} := \left\{x \in \mathbb{R}^d \mid \exists u,\, g(x,u) \leq 0\right\}$. Assume that the function $\tilde{J}$ has the property

$$J^\star(x) \leq \tilde{J}(x) \leq J^\star(x) + \epsilon\gamma\left(\|x\|\right) ,\forall x \in \mathcal{R} \ ,$$

for some $\epsilon < 1$.

The following sections will demonstrate that such a PWA approximate function can be used to generate a stabilizing feasible explicit control law for the MPC problem (7). Section V-D will then discuss how to generate an appropriate PWA function using the techniques developed in this paper.

A. Barycentric Control Input

Using the techniques proposed in this paper, it is possible to compute a PWA convex function $\tilde{J}$ that is of any specified complexity or error and is an upper approximation of the optimal cost $J^\star$ of (7) (Section V-D). Let $\tilde{J}$ be the piecewise affine function

$$\tilde{J}(x) := b_i^T x + c_i, \quad \text{if } x \in R_i \quad (10)$$

where the polytopes R_i form a convex partition: $\mathcal{R} = \cup R_i$ and $\operatorname{int} R_i \cap \operatorname{int} R_j = \emptyset$ for all $i \neq j$. We assume an approximation error of ϵ, $J^\star(x) \leq \tilde{J}(x) \leq J^\star(x) + \epsilon\gamma\left(\|x\|\right)$, $\forall x \in \mathcal{R}$ and that $\tilde{J}$ is defined over a subset of the feasible set of (9), $\mathcal{R} \subseteq \mathbb{X}$.

Our goal is now to use this function $\tilde{J}$ to compute an approximate feasible solution $\tilde{u}(x) := \begin{bmatrix}\tilde{u}_0(x)^T & \ldots & \tilde{u}_{N-1}(x)^T\end{bmatrix}^T$ to (9) and demonstrate that there exists a Lyapunov function for the resulting approximate closed-loop system $x^+ = Ax + B\tilde{u}_0(x)$.

The authors are aware of three proposals in the literature to tackle the problem of computing a function $\tilde{u}$ from an approximate cost $\tilde{J}$, 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], [20]. 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 [6] 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 and furthermore, the resulting control law will be discontinuous. The third approach [10], [23] 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 solution based on Barycentric coordinates, which does not generate any new regions.

Definition 15 (Barycentric function): Let $S := \operatorname{conv}\left(\{v_1, \ldots, v_n\}\right) \subset \mathbb{R}^d$ be a polytope. The set of functions $w_v(x)$, $v \in \operatorname{extreme}(S)$ is called

barycentric if three conditions hold for all $x \in S$

$$w_v(x) \geq 0 \quad \textit{positivity} \tag{11a}$$

$$\sum_{v \in \text{extreme}(S)} w_v(x) = 1 \quad \textit{partition of unity} \tag{11b}$$

$$\sum_{v \in \text{extreme}(S)} v w_v(x) = x \quad \textit{linear precision} \tag{11c}$$

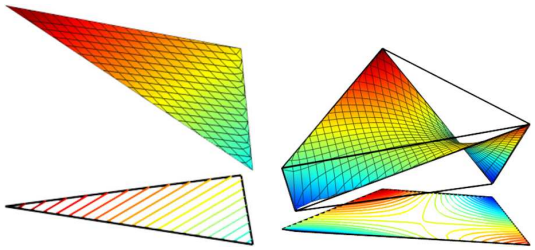
For each vertex $v \in \text{extreme}(R_i)$ and region R_i, we define $u^\star(v)$ to be an optimizer of (9) at the point v. Note that each such v is feasible, by the assumption that $\mathcal{R} \subseteq \mathbb{X}$ and so an appropriate $u^\star(v)$ can always be computed. If a set of barycentric functions $w_v(x)$, $v \in \text{extreme}(R_i)$ is available for each region R_i in (10), then we can define an approximate solution $\tilde{u}(x)$ by interpolating among these points over the regions R_i.

$$\tilde{u}(x) := \sum_{v \in \text{extreme}(R_i)} u^\star(v) w_v(x) \ , \quad \text{if } x \in R_i \ . \tag{12}$$

Remark 16: Note that the proposed approximate controller will be applied in a receding horizon fashion, as is the case for the optimal control law. The controller is therefore defined only by the first step $\tilde{u}_0(x)$ of the N-step prediction sequence. As is the case in optimal MPC, the remaining $N-1$ steps of the prediction sequence are used only to prove recursive feasibility and stability of the resulting approximate receding horizon control law.

Figure 3 shows two examples of the proposed Barycentric interpolation. One can see that interpolation across a simplex leads to an affine function, whereas a more general polytope gives a smooth and continuous function with the key property that it lies within the convex hull of the extreme points, which is proven in the next section. It is this containment that allows Corollary 18 to prove that the approximate control law $\tilde{u}_0(x)$ is feasible.

The following section provides sufficient conditions on the approximate polyhedral cost function for the resulting barycentric control law to be stabilizing. The techniques proposed in this paper (Algorithm 2) can also, of course, be used to generate controllers that do not satisfy these conditions, but may nevertheless still be stabilizing.



(a) Interpolation over a simplex.

(b) Barycentric interpolation over a polytope.

Fig. 3. Example of Barycentric interpolation over a simplex and a polytope. Note that the interpolation lies within the convex hull of the extreme points.

B. Stability of Barycentric Control Law

This section demonstrates that the proposed barycentric control law is stabilizing by constructing a Lyapunov function for the approximate closed-loop system $x^+ = Ax + B\tilde{u}_0(x)$.

We first show that the approximate solution $\tilde{u}(x)$ is everywhere feasible with the following lemma, which proves that $\tilde{u}(x)$ lies inside the convex hull of the optimizer $u^\star(x)$ at each of the vertices of R for each $R \in \mathcal{R}$.

Lemma 17: If $R = \text{conv}\,(v_0, \ldots, v_m) \in \mathcal{R}$, $u^\star(v_i)$ is the optimizer of (9) for the state v_i and $\tilde{u}$ is defined as in (12), then

$$\begin{pmatrix} x \\ \tilde{u}(x) \end{pmatrix} \in \text{conv}\left(\begin{pmatrix} v_0 \\ u^\star(v_0) \end{pmatrix} \cdots \begin{pmatrix} v_m \\ u^\star(v_m) \end{pmatrix} \right) \ , \tag{13}$$

for all $x \in R$.

Proof: The statement holds if for each $x \in \mathcal{R}$ there exists a set of positive multipliers $\lambda_0, \ldots, \lambda_m$ such that

$$\begin{pmatrix} x \\ \tilde{u}(x) \end{pmatrix} = \sum_{i=0}^{m} \lambda_i \begin{pmatrix} v_i \\ u^\star(v_i) \end{pmatrix}$$

and $\sum_{i=0}^{m} \lambda_i = 1$. The properties of barycentric functions (11) clearly satisfy this requirement. ■

Lemma 17 leads directly to the statement that $\tilde{u}(x)$ is a feasible solution of the parametric program (9) for all $x \in \text{conv}\,(\mathcal{R})$.

Corollary 18: The function $\tilde{u}$ is a feasible solution of (12)

$$g(x, \tilde{u}(x)) \leq 0, \qquad \text{for all } x \in \mathcal{R} \ .$$

Proof: Follows directly from Lemma 17 and the convexity of g.

■

We now show that the cost function J of (8) evaluated for the barycentric approximate solution $\tilde{u}(x)$ is no more sub-optimal than the PWA function used to generate it. Note that the barycentric interpolation will cause the cost $J(\tilde{u}(x))$[2] to be a non-linear and possibly non-convex function of the state.

Lemma 19: If $\tilde{u}$ is the function defined in (12) by barycentric interpolation from the polyhedral function $\tilde{J}$ (10), then the following bounds hold

$$J^\star(x) \leq J(\tilde{u}(x)) \leq \tilde{J}(x) \ , \qquad \forall x \in \mathcal{R} \ .$$

Proof: The solution $\tilde{u}$ is feasible for all $x \in \mathcal{R}$ (Corollary 18) and so the cost function $J(\tilde{u}(x))$ must be sub-optimal, which gives the lower bound $J^\star(x) \leq J(\tilde{u}(x))$.

We now show that the upper bound holds. If the barycentric input sequence $\tilde{u}(x) = \begin{bmatrix}\tilde{u}_0(x)^T & \dots & \tilde{u}_{N-1}(x)^T\end{bmatrix}^T$ is applied to the system that is currently in state $x \in R$, then the resulting state at time i will be

$$\begin{aligned}
\tilde{x}_i(x) &= A^i x + \sum_{j=0}^{i-1} A^j B \tilde{u}_j(x) \\
&= A^i x + \sum_{j=0}^{i-1} A^j B \left(\sum_{v \in \text{extreme}(R)} u_j^\star(v) w_v(x) \right) \\
&= \sum_{v \in \text{extreme}(R)} w_v(x) \left(A^i x + \sum_{j=0}^{i-1} A^j B u_j^\star(v) \right) \\
&\qquad \text{(Linearity and partition of unity)} \\
&= \sum_{v \in \text{extreme}(R)} w_v(x) x_i^\star(v) \ ,
\end{aligned}$$

where $x_i^\star(v)$ is the state that the system would be in at time i if the input sequence $u_0^\star(v), \dots, u_{i-1}^\star(v)$ that is optimal for the MPC problem (7) at state v were applied to the system.

We can now evaluate the cost function for the sub-optimal input sequence defined by the Barycentric interpolation:

$$\begin{aligned}
J(\tilde{u}(x)) &= V_N(\tilde{x}_N(x)) + \sum_{i=0}^{N-1} l(\tilde{x}_i(x),\ \tilde{u}_i(x)) \\
&= V_N \left(\sum_{v \in \text{extreme}(R)} x_N^\star(v) w_v(x) \right) + \\
&\qquad \sum_{i=0}^{N-1} l \left(\sum_{v \in \text{extreme}(R)} x_i^\star(v) w_v(x), \right. \\
&\qquad\qquad \left. \sum_{v \in \text{extreme}(R)} u_i^\star(v) w_v(x) \right)
\end{aligned}$$

Convexity of l and V_N then gives the following relation:

$$\begin{aligned}
&\leq \sum_{v \in \text{extreme}(R)} w_v(x) \left(V_N(x_N^\star(v)) + \right. \\
&\qquad \left. \sum_{i=0}^{N-1} l(x_i^\star(v),\ u_i^\star(v)) \right) \\
&= \sum_{v \in \text{extreme}(R)} w_v(x) J^\star(v)
\end{aligned}$$

By assumption $J^\star(v) \leq \tilde{J}(v)$ for all $v \in R$

$$\leq \sum_{v \in \text{extreme}(R)} w_v(x) \tilde{J}(v)$$

Lemma 17 states that for each $x \in R$, the above equation will lie within the convex hull of the extreme points $\left\{ (v, \tilde{J}(v)) \mid v \in \text{extreme}(R) \right\}$. Since the function $\tilde{J}(x)$ is affine within the region R, this implies that the above equation simply equals $\tilde{J}(x)$ because the convex hull of an affine set is the set itself, which gives the desired relation. ■

[2]We will use the shorthand notation $J(\tilde{u}(x))$ to mean the evaluation of the cost function (8) for $J(\tilde{u}_0(x), \dots, \tilde{u}_{N-1}(x), \tilde{x}_0(x), \dots, \tilde{x}_N(x))$, where $\tilde{x}_i(x)$ is the state of the system $x^+ = Ax + Bu$ at time i given that the state at time zero is x and the input sequence $\tilde{u}_0(x), \dots, \tilde{u}_{i-1}(x)$ has been applied.

The above lemma shows that the nonlinear, non-convex cost function $J(\tilde{u}(x))$ inherits the approximation error of the PWA function $\tilde{J}$ used to create it. With this key result in place, we can then make use of the approximate stability result given in [32], which shows that $J(\tilde{u}(x))$ is a Lyapunov function for the approximate system $x^+ = Ax + B\tilde{u}_0(x)$. Note that the statement of the theorem has been changed to match the notation of this paper.

Theorem 20 ([32]): Let $J^\star : \mathbb{R}^d \to \mathbb{R}$ be the cost function of the optimal control problem (7) and a Lyapunov function for the system $x^+ = Ax + Bu_0^\star(x)$. The approximate value function $J(\tilde{u}(x))$ is a Lyapunov function for the system $x^+ = Ax + B\tilde{u}_0(x)$ if for all $x \in \mathcal{R}$ the condition $J^\star(x) \leq J(\tilde{u}(x)) \leq J^\star(x) + \gamma(\|x\|)$ holds.

A Lyapunov function is insufficient to prove stability for a constrained system, since the system must also be invariant or feasible for all time. As discussed in [20], since level sets of Lyapunov functions are invariant [11], it is possible to determine an invariant subset of $\mathcal{R}$ given the vertices of each region R_i with minor additional processing.

Corollary 21 ([20]):
If $J_{\min} := \min\{J(\tilde{u}(v)) \mid v \in \text{extreme}(\mathcal{R})\}$ and the conditions of Theorem 20 are satisfied, then the set

$$I := \{x \in \mathcal{R} \mid J(\tilde{u}(x)) \leq J_{\min}\}$$

is invariant under the control law $\tilde{u}(x)$.

C. Barycentric Functions

Our goal is now to define an easily computable barycentric function for each polytope R_i in (10). 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 [35] a very elegant method of computing a barycentric function for arbitrary polytopes was proposed that can be put to use here.

Lemma 22 (Barycentric coordinates for polytopes [35]): Let $S = \text{conv}(V) \subset \mathbb{R}^d$ 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 $P(V - \{x\}) \cap \{y \mid \langle v - x, y\rangle = 1\}$; 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 [35] 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 $P(V - \{x\})$

$$\oint_{P(V-\{x\})} y dy = \sum_i \alpha_i n_i = \sum_i \alpha_i \frac{v_i - x}{\|v_i - x\|_2} = 0 \ , \tag{14}$$

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 $v_i - x$ [17]. With some minor algebraic manipulation, (14) 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 [35]. Such computation is straightforward because both the vertices and halfspaces of each region are available due the double-description representation.

D. Approximate Polyhedral Cost Function

In this section we demonstrate how the methods developed in this paper can be used to compute an approximate PWA cost function that satisfies the stability conditions of Theorem 20. We begin by defining the function

$$J^\epsilon(x) := J^\star(x) + \epsilon\gamma(\|x\|) \ ,$$

for some $0 \leq \epsilon < 1$, which is clearly convex. One can now see that the conditions of Theorem 20 are equivalent to stating that the approximate function $\tilde{J}$ must lie between $J^\star$ and $J^1 := J^\star(x) + \gamma(\|x\|)$

$$J^\star(x) \leq \tilde{J}(x) \leq J^\epsilon(x) \leq J^1(x)\,, \text{ for all } x \in \mathcal{R}\ .$$

The parameter $0 \leq \epsilon < 1$ can be used to define a trade-off between the approximation error and the complexity of the resulting controller, since all $\epsilon < 1$ define stabilizing control laws. The implicit double description algorithm can now be used to compute an outer approximation of the epigraph of J^ϵ that is of sufficient accuracy that it also lies above $J^\star$. The remainder of this section outlines how to achieve this goal.

The epigraph of J^ϵ is a convex set defined implicitly through a projection operation

$$\mathrm{epi}(J^\epsilon) = \left\{ (x,t) \in \mathbb{R}^d \times \mathbb{R} \;\middle|\; \begin{array}{l} \exists u,\, g(x,u) \leq 0, \\ t \geq h(x,u) + \epsilon\gamma(\|x\|) \end{array} \right\}\ . \tag{15}$$

The methods described in this paper cannot be used for the problem of approximating $\mathrm{epi}\,(J^\epsilon)$ directly, since it is unbounded and so we first derive a bounded convex set to which Algorithm 2 can be applied.

The feasible set of (9) is defined by the projection operation

$$\mathbb{X} := \left\{ x \in \mathbb{R}^d \mid \exists u,\, g(x,u) \leq 0 \right\}\ , \tag{16}$$

which is convex and bounded by assumption. As a result, we can directly use Algorithm 2 to compute an inner approximation $\mathcal{R}$ of $\mathbb{X}$ of any desired error $\epsilon_{\mathbb{X}}$, $\rho(\mathcal{R}, \mathbb{X}) \leq \epsilon_{\mathbb{X}}$.

Remark 23: An inner approximation of the feasible set with a Hausdorff error of $\epsilon_{\mathcal{R}}$ can also be directly computed from the outer approximation generated in Section IV-A. If $P(A)$ is an outer $\epsilon-$approximation of $\mathbb{X}$, then

$$\begin{aligned} \mathcal{R} &:= P(A) \ominus \{x \mid \|x\|_2 \leq \epsilon\} \\ &= \left\{ x \;\middle|\; a^T x \leq 1 - \epsilon\|a\|,\, \forall a \in A \right\}\ , \end{aligned}$$

is an inner $\epsilon-$approximation, where $\ominus$ is the Pontryagin difference (see, for example, [22]).

The convexity of J^ϵ and the set $\mathcal{R}$ provides a simple tight upper bound on the function J^ϵ restricted to the domain $\mathcal{R}$. If $\bar{J}^\epsilon := \max_{v \in \mathrm{extreme}(\mathcal{R})} J^\epsilon(v)$, then $J^\epsilon(x) \leq \bar{J}^\epsilon$ for all $x \in \mathcal{R}$ and we can simply define a bounded convex set whose lower convex hull is the function J^ϵ

$$\begin{aligned} &\mathbb{J}^\epsilon := \\ &\left\{ (x,J) \;\middle|\; \begin{array}{l} \exists u,\, \bar{J}^\epsilon \geq J + 1 \geq h(x,u) + \epsilon\gamma(\|x\|) \\ g(x,u) \leq 0 \end{array} \right\} \\ &\cap \left\{ (x,J) \mid x \in \mathcal{R} \right\}\ , \end{aligned} \tag{17}$$

Note that we shift the epigraph downwards by one so that the origin is in the strict interior of the set $\mathbb{J}^\epsilon$ as required by Algorithm 2.

The implicit double description Algorithm 2 can now be run on the bounded convex set $\mathbb{J}^\epsilon$.The algorithm is stopped when the outer approximation lies entirely within the epigraph of the optimal cost function $J^\star$. The piecewise affine function formed by the lower convex hull of the outer approximation will then lie above the optimal cost $J^\star$ and below the upper bound J^ϵ and be defined over the approximate feasible set $\mathcal{R}$, and hence provides the desired function.

VI. Examples

A. Linear MPC Example I

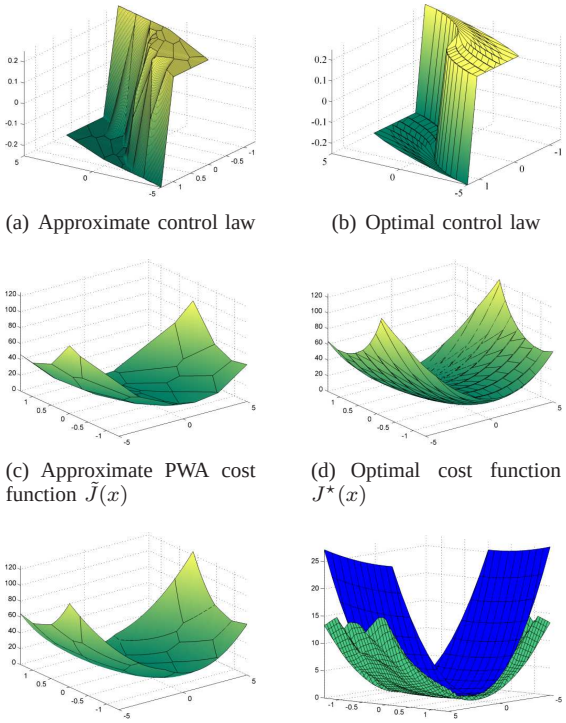
Consider the following simple two-dimensional example:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u\ ,$$

with the input and state constraints $|u| \leq 0.25$, $\|x\|_2 \leq 5$ and a horizon N of length 10 with the stage cost taken to be $l(x,u) := x'x + 0.01u'u$. The terminal control law $\mu(x)$ was taken to be the LQR controller $\mu(x) := Kx$ for the unconstrained system with the same weightings as the stage cost $l(\cdot,\cdot)$. The terminal constraint set $\mathcal{X}_F$ is the maximum invariant set for the closed-loop system $x^+ = (A + BK)x$ and the terminal weight $V_N(x)$ is the corresponding infinite-horizon cost $x^T P x$.

The optimal control law in this case requires 212 regions and can be seen in Figure 4. We here set

an approximation error $\epsilon = 0.5$ and compute a stabilizing control law using the method proposed in Section V that consists of 35 regions. The resulting control law and sub-optimal cost functions are shown in Figure 4.



(a) Approximate control law

(b) Optimal control law

(c) Approximate PWA cost function $\tilde{J}(x)$

(d) Optimal cost function $J^\star(x)$

(e) Cost function $J(\tilde{u}(x))$ evaluated for approximate function $\tilde{u}$

(f) $J(\tilde{u}(x)) - J^\star(x)$ (green) and $\gamma(\|x\|) = x^T x$ (blue)

Fig. 4. Approximate control law over 35 regions for example VI-A. Note that the barycentric interpolation is non-linear and continuous across the non-simplical regions. Figure 4(f) demonstrates that the sub-optimal controller is stable.

B. Stochastic MPC Example

We now study an example from [28], in which the linear model $x^+ = Ax + Bu + Ew$ represents a simplified building consisting of only one room. The states x are the temperatures of the room, the internal and the external walls respectively, while the input $u \in \mathbb{R}$ specifies the desired level of heating or cooling. The three disturbances $w \in \mathbb{R}^3$ are the external temperature, solar radiation and internal heat gains (people, computers, etc) and are assumed to be Gaussian and independent $w \sim \mathcal{N}(0, I)$. The system matrices are given by

$$A := \begin{bmatrix} 0.8511 & 0.0541 & 0.0707 \\ 0.1293 & 0.8635 & 0.0055 \\ 0.0989 & 0.0032 & 0.7541 \end{bmatrix} \qquad B := \begin{bmatrix} 0.35 \\ 0.03 \\ 0.02 \end{bmatrix}$$

$$E := \begin{bmatrix} 0.1005 & 0.3973 & 0.5022 \\ 0.0340 & 0.7555 & 0.2105 \\ 0.9019 & 0.0427 & 0.0555 \end{bmatrix} .$$

The goal is to maintain the room temperature within a given comfort range of $\pm 5^\circ C$ with a probability of $1 - \alpha = 99\%$:

$$\mathbf{P}\left(\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x \leq 5\right) \geq 1 - \alpha \tag{18}$$

$$\mathbf{P}\left(\begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x \geq -5\right) \geq 1 - \alpha \ , \tag{19}$$

where we note that the system is linearized around an operating point of $x = 25^\circ C$.

We set up an MPC problem with a horizon of five steps with the goal of minimizing the expected value of the energy usage

$$J(x) := \mathbf{E}\left[\sum_{i=0}^{N} u_i^2\right] \ ,$$

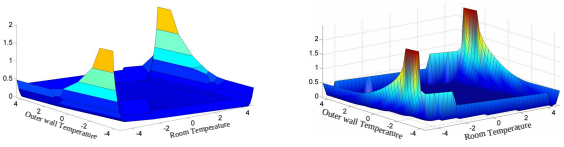
where the input u_i is chosen to be an affine function of the disturbances $w_0, \ldots, w_{i-1}$

$$u_i = \sum_{j=0}^{i} M_{i,j} w_j + v_i \ ,$$

and the optimization variables are $M_{\star,\star}$ and $v_\star$. This approach was originally suggested in [14] in the context of stochastic programs with recourse, but has recently generated a resurgence of interest in the robust and stochastic control community [9], [16], [24]. The resulting optimization can be recast as an equivalent second-order cone problem (SOCP) (e.g. [34]):

$$J^\star(x) := \min_{v_\star,\ M_{\star,\star}} \sum_i v_i^T v_i + \sum_{j=0}^{i-1} \operatorname{trace}\left(M_{i,j}^T M_{i,j}\right)$$
$$\sqrt{2}\,\mathrm{erf}^{-1}(1-2\alpha)\left\|\begin{bmatrix} A^{i-1}B & \cdots & A^0 B \end{bmatrix}\right\|$$
$$+ CA^i x + \sum_{j=0}^{i-1} CA^j B v_j \leq 0,$$
$$\forall i = 0, \ldots, 5$$

The proposed method, Algorithm 2, has been applied to the epigraph of the optimal cost function $J^\star$ and the resulting approximation error as a function of the number of regions is shown in Figure 6. The optimal and approximate cost function is shown in Figure 5 for an approximation error of 0.01. Note that stability has not been considered in this example.



(a) Approximate cost function (b) Optimal cost function

Fig. 5. Plot of the optimal and approximate cost function for Example VI-B.
Note: Interior wall temperature is set to zero in the plot in order to generate a three dimensional figure.

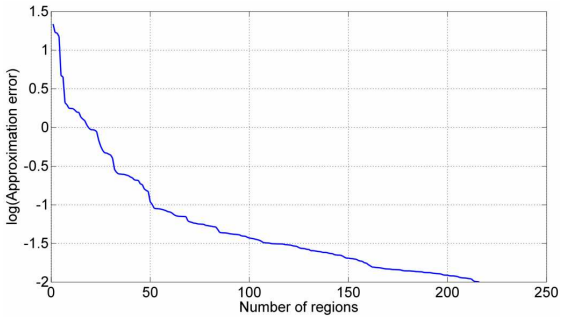


Fig. 6. Approximation error vs number of regions for Example VI-B.

C. Linear MPC Example II

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 stage cost $l(x,u) = \|x\|_\infty + \|u\|_\infty$ with a prediction horizon of $N = 5$.

Figure 7 shows a plot of complexity (number of regions) vs the approximation error of Algorithm 2. Figure 8 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. Note that stability has not been considered in this example.

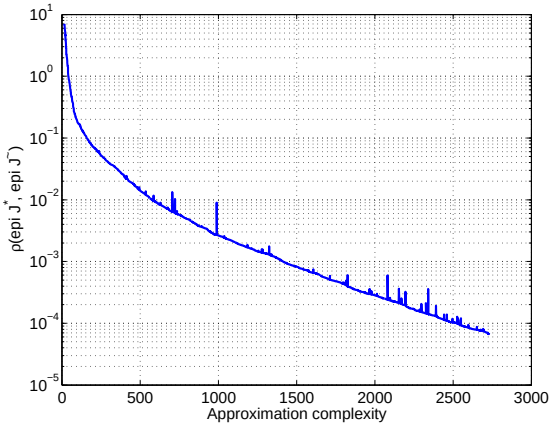


Fig. 7. Approximation error of the four-state system of Example VI-C vs the approximation complexity (number of polyhedral regions in the PWA cost function). The optimal solution consists of $12,128$ regions. (The noise in the plot is due to numerical errors.)

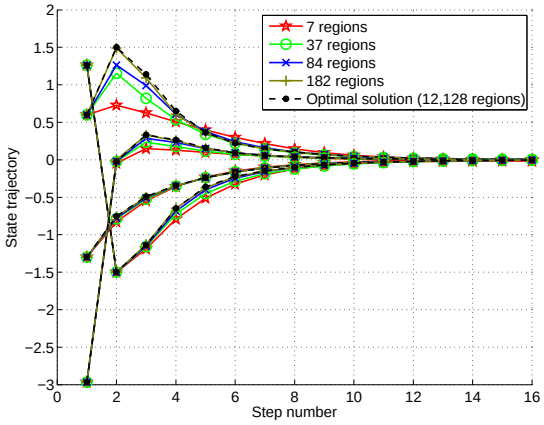


Fig. 8. Example trajectory of the approximate and optimal solutions of Example VI-C for various approximation levels.

VII. Conclusion

This paper has proposed a simple, constructive technique for generating inner and/or outer polytopic approximations of convex sets of any specified complexity. The algorithm is computationally efficient, in that it is based on the well-established double-description algorithm and requires the solution of only a single convex optimization problem per facet (vertex) of the outer (inner) approximation. This is a key improvement over existing methods, which require a number of optimization problems to be solved equal to the complexity of the *optimal* solution, which can often be many orders of magnitude larger than the approximation, if it is computable at all. The algorithm operates in a greedy-optimal incremental fashion, in that it updates the approximation at each step with the facet (vertex) that minimizes the Hausdorff distance between the approximation and the set to be approximated.

The proposed implicit double-description algorithm can be employed to synthesize sub-optimal, stabilizing explicit control laws for convex MPC problems. The key benefit is that the complexity, or number of pieces, in the resulting piecewise polynomial control law can be pre-specified, which is equivalent to stating that any given memory or online computational bounds can be met for a given embedded processor.

A proof of stability and invariance of the resulting sub-optimal closed-loop system was provided. Invariance, or feasibility, follows directly from the construction of the control law, which is based on barycentric interpolation, as well as the assumed convexity of the system constraints. Stability is based on a classic result [32], and it was shown that if the approximation is sufficiently close, then the conditions given in [32] will be satisfied. Furthermore, the algorithm will always satisfy the conditions of [32] after a finite number of iterations. The main limitation of the method is that this finite number cannot be determined a priori, although several examples (Section VI) demonstrate that this number is generally not too large.

References

[1] A. Alessio and A. Bemporad. A survey on explicit model predictive control. In *Assessment and Future Directions of NMPC*, Pavia, Italy, Sept 2008.

[2] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483, Dec 1996.

[3] D. J. Bates, I.A. Fotiou, and Ph. Rostalski. A numerical algebraic geometry approach to nonlinear constrained optimal control. In *IEEE Conference on Decision and Control*, New Orleans, LA, December 2007.

[4] A. Bemporad, F. Borrelli, and M. Morari. Explicit solution of constrained $1\backslash\infty-$norm model predictive control. In *Proceedings of the 39th IEEE Conference on Decision and Control*, 2000.

[5] A. Bemporad, F. Borrelli, and M. Morari. Model predictive control based on linear programming the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, Dec 2002.

[6] A. Bemporad and C. Filippi. An algorithm for approximate multiparametric convex programming. *Computational Optimization and Applications*, V35(1):87–108, 2006.

[7] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, Mar 1999.

[8] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[9] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, Mar 2004.

[10] J. Bjornberg and M. Diehl. Approximate robust dynamic programming and robustly stable MPC. *Automatica*, 42(5), 2006.

[11] F. Blanchini. Set invariance in control (survey paper). *Automatica*, 35:1747–1767, 1999.

[12] E. M. Bronshteyn and L. D. Ivanov. The approximation of convex sets by polyhedra. *Siberian Mathematical Journal*, 16(5):852–853, 1976.

[13] K. Fukuda and A. Prodon. Double description method revisited. In M. Deza, R. Euler, and I. Manoussakis, editors, *Combinatorics and Computer Science*, volume 1120 of *Lecture Notes in Computer Science*, pages 91–111. Springer-Verlag, 1996.

[14] Stanley J. Garstka and Roger J. B. Wets. On decision rules in stochastic programming. *Mathematical Programming*, 7(1):117–143, Dec 1974.

[15] J. E. Goodman and J. O'Roarke, editors. *Handbook of Discrete and Computational Geometry*. CRC Press, 1997.

[16] Paul J. Goulart, Eric C. Kerrigan, and Daniel Ralph. Efficient robust optimization for robust control with constraints. *Mathematical Programming*, 114(1):115–147, Jul 2008.

[17] B. Grünbaum. *Convex Polytopes*. Springer-Verlag, second edition, 2000.

[18] 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*, pages 662–667, 2000.

[19] T.A. Johansen and A. Grancharova. Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, 48(5):810–815, May 2003.

[20] CN Jones, M Baric, and M Morari. Multiparametric linear programming with applications to control. *European Journal of Control*, 13(2-3):152–170, May 2007.

[21] M. Kallay. Convex hull algorithms for higher dimensions. Technical report, Mathematics, Univ. of Oklahoma, Norman, Oklahoma, 1981.

[22] M. Kvasnica, P. Grieder, and M. Baotić. Multi-Parametric Toolbox (MPT), 2004. http://control.ee.ethz.ch/~mpt/.

[23] B. Lincoln and A. Rantzer. Relaxing dynamic programming. *IEEE Transactions on Automatic Control*, 51(8):1249–1260, Aug 2006.

[24] J. Lofberg. Approximations of closed-loop minimax MPC. In *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, volume 2, pages 1438–1442 Vol.2, 2003.

[25] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[26] J Mitchell. Separation and approximation of polyhedral objects. *Computational Geometry*, 5(2):95–114, Sep 1995.

[27] T. S. Motzkin, H. Raiffa, G. L. Thompson, and R. M. Thrall. *Contributions to the theory of games*, chapter The double description method. Princeton Uni. Press, Princeton, 1953.

[28] F. Oldewurtel, C.N. Jones, and M. Morari. A tractable approximation of chance constrained stochastic MPC based on affine disturbance feedback. In *Conference on Decision and Control, CDC*, Cancun, Mexico, December 2008.

[29] R. T. Rockafellar. *Convex Analysis*. Princeton Univ. Press, 1970.

[30] M. Schechter. Polyhedral functions and multiparametric linear programming. *Journal of Optimization Theory and Applications*, V53(2):269–280, 1987.

[31] R. Schneider and J. A. Wieacker. Approximation of convex bodies by polytopes. *Bulletin of the London Mathematical Society*, 13(2):149–156, Mar 1981.

[32] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *Automatic Control, IEEE Transactions on*, 44(3):648–654, 1999.

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

[34] D.H. van Hessem and O.H. Bosgra. A conic reformulation of model predictive control including bounded and stochastic disturbances under state and input constraints. In *Conference on Decision and Control, CDC*, volume 4, pages 4643–4648. IEEE, 2002.

[35] Joe Warren, Scott Schaefer, Anil N. Hirani, and Mathieu Desbrun. Barycentric coordinates for convex sets. *Advances in Computational Mathematics*, 27(3):319–338, Aug 2007.

Colin Jones (M'09) is a senior researcher at the ETH in Zurich. He obtained a PhD in 2005 from the University of Cambridge in Control Engineering. Prior to that, he was at the University of British Columbia in Canada, where he took a BASc in Electrical Engineering and Mathematics, as well as an MASc, also in Electrical Engineering. Colin has worked in a variety of industrial roles, ranging from control of heating, ventilation and air conditioning for commercial buildings, to control of hydro-metallurgical processes. He is co-founder of Apex Optimization, which specializes in optimal scheduling for retail outlets. His research interests are in the areas of robust and predictive control, computational geometry and optimization.

Manfred Morari (M'84-F'05) was appointed head of the Department of Information Technology and Electrical Engineering at ETH Zurich in 2009. He was head of the Automatic Control Laboratory from 1994 to 2008. Before that he was the McCollum-Corcoran Professor of Chemical Engineering and Executive Officer for Control and Dynamical Systems at the California Institute of Technology. He obtained the diploma from ETH Zurich and the Ph.D. from the University of Minnesota, both in chemical engineering. His interests are in hybrid systems and the control of biomedical systems. In recognition of his research contributions he received numerous awards, among them the Donald P. Eckman Award and the John R. Ragazzini Award of the Automatic Control Council, the Allan P. Colburn Award and the Professional Progress Award of the AIChE, the Curtis W. McGraw Research Award of the ASEE, Doctor Honoris Causa from Babes-Bolyai University, Fellow of IEEE, the IEEE Control Systems Field Award, and was elected to the National Academy of Engineering (U.S.). Manfred Morari has held appointments with Exxon and ICI plc and serves on the technical advisory boards of several major corporations.