Approximate robust dynamic programming and robustly stable MPC

Jakob Björnberg, Moritz Diehl

Centre for Mathematical Sciences, Wilberforce Road, CB3 0WB Cambridge, UK
IWR, University of Heidelberg, Im Neuenheimer Feld 368, D-69120 Heidelberg, Germany

Received 20 January 2005; received in revised form 30 June 2005; accepted 22 December 2005
Available online 14 February 2006

Abstract

We present a technique for approximate robust dynamic programming that is suitable for linearly constrained polytopic systems with piecewise affine cost functions. The approximation method uses polyhedral representations of the cost-to-go function and feasible set, and can considerably reduce the computational burden compared to recently proposed methods for exact robust dynamic programming [Bemporad, A., Borrelli, F., & Morari, M. (2003). Min–max control of constrained uncertain discrete-time linear systems. IEEE Transactions on Automatic Control, 48(9), 1600–1606; Diehl, M., & Björnberg, J. (2004). Robust dynamic programming for min–max model predictive control of constrained uncertain systems. IEEE Transactions on Automatic Control, 49(12), 2253–2257]. We show how to apply the method to robust MPC, and give conditions that guarantee closed-loop stability. We finish by applying the method to a state constrained tutorial example, a parking car with uncertain mass.

Keywords: Dynamic programming; Receding horizon control; Min–Max model predictive control; Robustness

1. Introduction

Robust model predictive control (MPC), originally proposed by Witsenhausen (1968), is an emerging control technique based on a worst-case optimization of future system behaviour. While a key assumption in classical MPC is that the system is deterministic and known, in robust MPC the system is not assumed to be known exactly, and the optimization is performed against the worst-case predicted system behaviour. Robust MPC thus typically leads to min–max optimization problems, which either arise from an open loop, or from a closed-loop formulation of the optimal control problem (Lee & Yu, 1997). In this paper, we are concerned with the less conservative, but computationally more demanding closed-loop formulation treating the same system class as this paper see Pluymers, Rossiter, Suykens, & De Moor, 2005). We regard discrete-time polytopic systems with piecewise affine cost and linear constraints only.

For this problem class, the closed-loop formulation of robust MPC leads to multi-stage min–max optimization problems that can be attacked by a scenario tree formulation (Kerrigan & Maciejowski, 2004; Scokaert & Mayne, 1998) or by robust dynamic programming (DP) approaches (Bemporad, Borrelli, & Morari, 2003; Diehl & Björnberg, 2004; Mayne, 2001). Note that the scenario tree formulation treats a single optimization problem for one initial state only, whereas DP produces the feedback solution for all possible initial states. Unfortunately, the computational burden of both approaches quickly becomes prohibitive even for small-scale systems as the size of the prediction horizon increases. The contribution of this article is a novel approximation technique for robust DP that considerably reduces the computational burden. This is at the expense of optimality, but still allows to generate robustly stable feedback laws that respect control and state constraints under all circumstances.

We first show, in Section 2, how the robust DP recursion can be compactly formulated entirely in terms of operations on sets. For the problem class we consider, these sets are polyhedra and can be explicitly computed (Bemporad et al., 2003; Diehl & Björnberg, 2004), as reviewed in Section 3. In Section 4, we generalize an approximation technique originally proposed in Lincoln and Rantzer (2002) (for deterministic DP). This allows...
us to approximate the result of the robust DP recursion with considerably fewer facets than in the exact approach. We also give conditions that guarantee robust closed-loop stability of the generated feedback law, entirely in terms of the polyhedral set representation, and illustrate in Section 5 with an example how the approximation approach can be used to synthesize such a robustly stable feedback. Finally, we conclude the paper in Section 6. Implementations of all algorithms presented in this paper are publicly available (Björnberg & Diehl, 2005).

2. Constrained robust dynamic programming

We consider discrete-time dynamic systems

\[ x_{k+1} = f_k(x_k, u_k), \quad f_k \in F, \quad k \in \mathbb{N} \]  

(1)

with states \( x_k \in \mathbb{R}^{n_x} \) and controls \( u_k \in \mathbb{R}^{n_u} \). The transition functions \( f_k \) in each step are uncertain, but we assume they are known to be in a certain set \( F \). Given an \( N \)-stage policy \( \pi = (\pi_0, \pi_1, \ldots, \pi_{N-1}) \), and given an uncertainty realization \( \phi = (f_0, f_1, \ldots, f_{N-1}) \in F^N \), as well as an initial value \( x_0 \), we define the corresponding state sequence \( x_k^{\pi, \phi, x_0}, k = 0, \ldots, N \), by \( x_0^{\pi, \phi, x_0} = x_0 \) and \( x_k^{\pi, \phi, x_0} = f_k(x_{k-1}^{\pi, \phi, x_0}, u_k(x_{k-1}^{\pi, \phi, x_0})) \). To each admissible event \( (x_k, u_k) \), i.e. state \( x_k \) and control \( u_k \) at time \( k \), a cost \( L_k(x_k, u_k) \) is associated, which is additive over time. The objective in robust DP is to find a feedback policy \( \pi^* \) that minimizes the worst-case total cost, i.e. solves the min–max optimization problem

\[
\min_{\pi \in F^N} \max_{\phi \in F^N} \left( \sum_{k=0}^{N-1} L_k(x_k^{\pi, \phi, x_0}, u_k(x_k^{\pi, \phi, x_0})) + V_N(x_N^{\pi, \phi, x_0}) \right)
\]

s.t. \( x_k^{\pi, \phi, x_0} \in \mathbb{X}_k \), \( \forall \phi \in F^N, \quad k \in [0, \ldots, N-1] \).

(2)

The sets \( \mathbb{X}_k \) and \( \mathbb{X}_N \) specify constraints on states and controls, and \( V_N(\cdot) \) is a final cost. Starting with \( V_N \) and \( \mathbb{X}_N \), we compute the optimal cost-to-go functions \( V_k \) and feasible sets \( \mathbb{X}_k \) recursively, for \( k = N-1, \ldots, 1 \), by the robust Bellman equation with constraints (cf. Mayne, 2001):

\[ V_k(x) := \min_{u \in \mathbb{R}^{n_u}} \tilde{V}_k(x, u) \quad \text{s.t.} \quad (x, u) \in \tilde{\mathbb{X}}_k, \]  

(3)

and

\[ \mathbb{X}_k := \{ x | \exists u : (x, u) \in \tilde{\mathbb{X}}_k \}, \]  

(4)

where \( \tilde{V}_k(x, u) := L_k(x, u) + \max_{f \in F} V_{k+1}(f(x, u)) \) and \( \tilde{\mathbb{X}}_k := \{ (x, u) \in \mathbb{X}_k | f(x, u) \in \mathbb{X}_{k+1} \forall f \in F \} \). The optimal feedback control \( u_k(x) \) from the optimal policy \( \pi^* \) can be determined as the minimizer of (3).

2.1. Epigraph representation

Given a set \( W \) and a function \( g : W \to \mathbb{R} \), define the epigraph of \( g \) by

\[ \text{epi}(g) := \{ (w, s) \in W \times \mathbb{R} | w \in W, s \geq g(w) \}. \]  

(5)

Given two subsets \( \mathbb{A} \) and \( \mathbb{B} \) of \( W \times \mathbb{R} \), define their cut-sum \( \mathbb{A} \cap \mathbb{B} \) by

\[ \mathbb{A} \cap \mathbb{B} := \{ (x, s + t) | (x, s) \in \mathbb{A}, (x, t) \in \mathbb{B} \}. \]  

(6)

If \( W \subseteq \mathbb{X} \times \mathbb{U} \) and \( f \) is any function \( W \to \mathbb{X} \), define an “epigraph function” \( f_E : W \times \mathbb{R} \to \mathbb{X} \times \mathbb{R} \) by \( f_E(x, u, s) := (f(x, u), s) \). We think of \( \mathbb{X} \) as a state space and \( \mathbb{U} \) as a space of controls. If we have a DP recursion with stage constraints defined by the sets \( \mathbb{L}_k \) and stage costs \( \mathbb{L}_k : \mathbb{L}_k \to \mathbb{R} \), let \( e_k = \text{epi}(L_k) \). Suppose the final cost and constraint set are \( \mathbb{V}_N \) and \( \mathbb{X}_N \), and let \( \mathbb{E}_N := \text{epi}(\mathbb{V}_N) \). Similarly, define \( \mathbb{E}_k = \text{epi}(V_k) \) for any \( 0 \leq k \leq N-1 \). Here, we regard \( V_k \) as a function defined on \( \mathbb{X}_k \) only.

We define the operation \( T_k \) on the set \( \mathbb{E}_{k+1} \) as follows:

\[
T_k(\mathbb{E}_{k+1}) := \left\{ \mathbb{E}_k \mid \bigcap_{f \in \mathbb{F}_k} f_E^{-1}(\mathbb{E}_{k+1}) \right\}, \]  

(7)

where \( \mathbb{F}_k := \{ f_E | f \in \mathbb{F} \} \).  

Proposition 1. \( \mathbb{E}_k = T_k(\mathbb{E}_{k+1}) \) for \( k = 0, \ldots, N-1 \).

The proof is straightforward and may be found in Björnberg and Diehl (2004). In view of Proposition 1, we call \( T_k \) the robust DP operator. We will also denote this by \( T_k \), suppressing the subscript, whenever the time dependency is unimportant. Using Proposition 1, we easily deduce the following monotonicity property, motivated by a similar property in DP (Bertsekas, 1995). Again, the proof can be found in Björnberg and Diehl (2004).

Proposition 2. If \( \mathbb{E}' \subseteq \mathbb{E} \) then \( T(\mathbb{E}') \subseteq T(\mathbb{E}) \).  

3. Polyhedral dynamic programming

From now on, we consider only affine systems with polytopic uncertainty, of the form

\[ f(x, u) = Ax + Bu + c. \]  

(8)

Here, the matrices \( A \) and \( B \) and the vector \( c \) are contained in a polytope

\[ \mathbb{F} = \text{conv}\{ (A_1 | B_1 | c_1), \ldots, (A_n | B_n | c_n) \}, \]  

(9)

and we identify each matrix \( (A|B|c) \in \mathbb{F} \) with the corresponding function \( f \). Polytopic uncertainty may arise naturally in application problems (see the example in this paper) or may be used to approximate nonlinear systems (for which other MPC techniques might be more appropriate).

We consider convex piecewise affine (CPWA) cost functions

\[ L(x, u) \geq \max \left( \tilde{P} \begin{bmatrix} x \\ 1 \end{bmatrix}, \tilde{Q} u \right), \quad V_N(x) = \max P_N \begin{bmatrix} x \\ 1 \end{bmatrix}, \]  

(10)

where the maximum is taken over the components of a vector, and \( \tilde{P}, \tilde{Q}, \) and \( P_N \) are matrices of appropriate dimensions.
For simplicity of notation, we assume here and in the following that the stage costs $L_k$ and feasible sets $\mathcal{U}_k$ in (2) do not depend on the stage index $k$. We treat linear constraints that result in polyhedral feasible sets

$$\mathcal{L} = \left\{ (x, u) \left| \hat{P} \begin{bmatrix} x \\ 1 \end{bmatrix} + \hat{Q} u \leq 0 \right. \right\} \cup \mathcal{X}_N = \left\{ x \left| \hat{P}_N \begin{bmatrix} x \\ 1 \end{bmatrix} \leq 0 \right. \right\}.$$

**Definition 3.** By polyhedral DP we denote robust DP (3)–(4) for affine systems $f$ with polytopic uncertainty (9), CPWA cost functions $L$ and $V_N$, and polyhedral constraint sets $\mathcal{L}$ and $\mathcal{X}_N$.

The point is that for polytopic $F$, CPWA $L$, $V_N$, and polyhedral sets $\mathcal{L}$, $\mathcal{X}_N$, all cost-to-go functions $V_k$ are also CPWA, and the feasible sets $\mathcal{X}_k$ are polyhedral. This is proved in Bemporad et al. (2003), and Diehl and Björnberg (2004), the latter of which uses epigraphs. We give an alternative formulation here, using the ideas and notation from Section 2.1.

**Theorem 4.** For $T$ a polyhedral DP operator and $E$ a polyhedron, also $T(E)$ as defined in (7) is a polyhedron.

**Proof.** We begin by proving that $\bigcap_{f \in E} f_{E}^{-1}(E)$ is a polyhedron. $E$ as given is the convex hull of the matrices $D_1, \ldots, D_{nf}$, with

$$D_i = \begin{pmatrix} A_i & B_i & 0 & c_i \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$ 

Assume that $E$ consists of all points $y$ satisfying $Q y \leq q$. Thus,

$$(x, u, s) \in f_{E}^{-1}(E) \iff QD \begin{bmatrix} x \\ u \\ s \\ 1 \end{bmatrix} \leq q.$$ 

It follows that

$$\bigcap_{f \in E} f_{E}^{-1}(E) = \bigcap_{i=1}^{n_f} \left\{ \begin{bmatrix} x \\ u \\ s \\ 1 \end{bmatrix} \bigg| \sum_{i=1}^{n_f} t_i QD_i \begin{bmatrix} x \\ u \\ s \\ 1 \end{bmatrix} \leq q \right\}$$

which as the finite intersection of polyhedra is again a polyhedron. In addition, the following lemmas hold (Björnberg & Diehl, 2004).

**Lemma 5.** The cut-sum of two polyhedra is a polyhedron.

Therefore, also the cut-sum $\tilde{E} = E \cup \bigcap_{f \in E} f_{E}^{-1}(E)$ is a polyhedron. Finally,

**Lemma 6.** If $\tilde{E}$ is a polyhedron, also $p(\tilde{E})$ is a polyhedron.

This is proven in Björnberg and Diehl (2004), and Diehl and Björnberg (2004) along with an explicit construction of $p(\tilde{E})$. □

The possibility to represent the result of the robust DP recursion algebraically can be used to obtain an algorithm for polyhedral DP. Such an algorithm was first presented by Bemporad et al. (2003), where the explicit solution of multi-parametric programming problems is used. The representation via convex epigraphs and the duality-based construction at the base of Lemma 6 above was first presented in Diehl and Björnberg (2004).

Polyhedral DP is exact, and does not require any tabulation of the state space. Thus, it avoids Bellman’s “curse of dimensionality”. However, the numbers of facets required to represent the epigraphs $E_k$ will in general grow exponentially. Even for simple examples, the computational burden quickly becomes prohibitive as the horizon $N$ grows. We propose a method for performing approximate polyhedral DP, which greatly reduces the computational burden, yet is able to preserve stability properties in MPC applications.

**4. An approximation technique for polyhedral dynamic programming**

Let $\mathcal{V} \subseteq \mathcal{W}$ be two polyhedra in $\mathbb{R}^n$. Suppose $\mathcal{V} = \{ x \left| v_i^T x_i \leq 0, i \in I_\mathcal{V} \}$, $\mathcal{W} = \{ x \left| w_j^T x_j \leq 0, j \in I_\mathcal{W} \}$.

We describe how to generate another polyhedron $\mathcal{A} = \{ x \in \mathbb{R}^n \left| v_i^T x_i \leq 0, \forall i \in I_\mathcal{A} \subseteq I_\mathcal{V} \}$, which satisfies $\mathcal{V} \subseteq \mathcal{A} \subseteq \mathcal{W}$, and which uses some of the inequalities from $\mathcal{V}$. In general, $\mathcal{A}$ will be represented by fewer inequalities than $\mathcal{V}$.

We show how to generate the index set $I_\mathcal{A}$, using intermediate sets $I_\mathcal{A}^{(k)}$. The method is a generalization of an idea from Lincoln and Rantzer (2002) to polyhedral sets. At each step, let $I_\mathcal{A}^{(k)} : = \{ x \in \mathbb{R}^n \left| v_i^T x_i \leq 0, \forall i \in I_\mathcal{A}^{(k)} \subseteq I_\mathcal{W} \}$. We denote robust DP (3)–(4) as defined in (7) is a polyhedron.

**Procedure—Polyhedron Pruning**

(1) Let $k = 0$ and $I_\mathcal{A}^{(0)} : = \emptyset$ (i.e. $\mathcal{A}^{(0)} : = \mathbb{R}^n$).

(2) Pick some $j \in I_\mathcal{W}$.

(a) If there exists an $x^* \in \mathcal{A}^{(k)}$ such that $w_j^T x^* \geq 0$, then let $i^* = \arg \max \{ i \in I_\mathcal{V} \left| w_j^T x_i^* \geq 0 \}$. Set $I_\mathcal{A}^{(k+1)} : = I_\mathcal{A}^{(k)} \cup \{ i^* \}$, and remove $i^*$ from $I_\mathcal{V}$.

(b) If no such $x^*$ exists, set $I_\mathcal{A}^{(k+1)} : = I_\mathcal{A}^{(k)}$ and remove $j$ from $I_\mathcal{W}$.

(3) If $I_\mathcal{W} = \emptyset$ then let $I_\mathcal{A} : = I_\mathcal{A}^{(k)}$ and end.

Otherwise set $k : = k + 1$ and go to 2.

The idea of polyhedron pruning is illustrated in Fig. 1.

**4.1. Choice of $x^*$ and scaling**

In step (2) of the algorithm, for given $j \in I_\mathcal{W}$ we have to find an $x^* \in \mathcal{A}^{(k)}$ such that $w_j^T x^* \geq 0$, or make sure that no such $x^*$ exists. We address this task using the following linear program:

$$\max w_j^T \begin{bmatrix} x \\ 1 \end{bmatrix} \text{ s.t. } w_i^T \begin{bmatrix} x \\ 1 \end{bmatrix} \leq \eta_i, \quad \forall i \in I_\mathcal{A}^{(k)},$$

(11)
Each vector $v_i$ no such optimal value of problem (11) is negative or zero, we know that $v_i$ by some $\eta$ (in our implementation we used $\eta=0.05$). If the optimal problem of value (11) is negative or zero, we know that no such $x^*$ exists. Otherwise we take its optimizer to be $x^*$.

From now on we make the following assumption:

**Assumption 1.** Each vector $v_i = \left[ \begin{array}{c} R \\ W \end{array} \right] \in \mathbb{R}^n \times \mathbb{R}$ defining $\mathcal{V}$ is normed so that $\|v_i\|_2 = 1$.

We require Assumption 1 for the following reason. The quantity

$$v_i^T \left[ \begin{array}{c} x^* \\ 1 \end{array} \right]$$

in (2) of the pruning procedure measures the vertical “height” of the hyperplane described by $v_i$. If Assumption 1 holds, then by maximizing (12) we find the piece of $\mathcal{V}$ that is furthest away from $x^*$, because then the height above $x^*$ and the distance from $x^*$ coincide. Note that we employ a different selection rule from Lincoln and Rantzer (2002): instead of adding the constraint $v_i^T \delta_i$ that is furthest away from $x^*$ in a predetermined direction, we choose it so that it maximizes the perpendicular distance to $x^*$.

### 4.2. Approximate polyhedral DP

We use the pruning method to do approximate DP.

**Procedure—Approximate Polyhedral DP**

1. Given the epigraph $\mathcal{E}_k$, the stage cost $L$ and feasible set $\mathcal{W}_k$, let $\mathcal{E}^{\text{out}} = T(\mathcal{E}_k)$ as in Section 2.1.
2. Choose some polyhedron $\mathcal{E}^{\text{in}}$ that satisfies $\mathcal{E}^{\text{in}} \subseteq \mathcal{E}^{\text{out}}$.
3. Let $\mathcal{E}_{k-1}$ be the result ($\mathcal{A}_k$) of applying the polyhedral pruning procedure with $\mathcal{V} = \mathcal{E}^{\text{in}}$ and $\mathcal{W} = \mathcal{E}^{\text{out}}$.

### 4.3. Choice of $\mathcal{E}^{\text{in}}$

There are many ways of choosing the set $\mathcal{E}^{\text{in}}$, and the choice is largely heuristic. Recall that $\mathcal{E}^{\text{out}} = T(\mathcal{E}_k)$ is the epigraph of the cost-to-go $V_k$ restricted to the feasible set $X_{k-1}$. To get a polyhedron contained in $\mathcal{E}^{\text{out}}$ you could raise or steepen the cost-to-go $V_k$, or you could diminish the size of the feasible set $X_{k-1}$, or both. We are mainly interested in the case where $x = 0$ is always feasible and the minimal value of the cost-to-go $V_k$ is attained at zero for all $k$. Then you can let

$$\mathcal{E}^{\text{in}} := \{(x, s) \in \mathbb{R}^n \times \mathbb{R} \mid (x, s) \in \mathcal{E}^{\text{out}} \}$$

with some $\alpha > 1$. This is illustrated in Fig. 2. It is difficult to assess the loss of optimality due to this approximation, as not only the cost function is made steeper, but also the feasible set is reduced. However, with the above definition, $\mathcal{E}^{\text{in}}$ is a factor $1/\alpha$ smaller than $\mathcal{E}^{\text{out}}$. For example, if we choose $\alpha = 1.1$ in the tutorial example below, we lose at most roughly 10% of the volume in each iteration.

### 4.4. Feedback and stability

For a given polyhedral epigraph $\mathcal{E}$, we can define an approximate robustly optimal feedback law $u(\cdot)$ by letting $u(x)$ be the solution of a linear programming problem (LP),

$$u(x) := \arg \min_{u, s} s \text{ s.t. } \left[ \begin{array}{c} x \\ u \\ s \end{array} \right] \in \tilde{\mathcal{E}}$$

with $\tilde{\mathcal{E}} = e \cap \bigcap_{E_k \in \mathcal{E}} E_k^{-1}(\mathcal{E})$ as in Theorem 4.

The computational burden associated with this LP in only $n_x + 1$ variables is negligible and can easily be performed online. Alternatively, a precomputation via multiparametric linear programming is possible (Bemporad et al., 2003).

The question of robust stability is a major concern in MPC applications and has been addressed, e.g. in Bemporad and Morari (1999), Kerrigan and Maciejowski (2004), and Mayne, Rawlings, Rao, and Scokaert (2000). The following theorem formulates easily verifiable conditions that guarantee that a certain set $\mathcal{T}$ is robustly asymptotically attractive for the closed-loop system, in the sense defined, e.g. in Kerrigan and Maciejowski (2004). The simple proof uses $V(x) := \min_{x_k, s} s \text{ s.t. } \left[ \begin{array}{c} x_k \\ s \end{array} \right] \in T(\mathcal{E})$ as a Lyapunov function and is omitted for the sake of brevity.

**Theorem 7** (Attractive set for robust MPC). Consider the closed-loop system

$$x_{k+1} = f_k(x_k, u(x_k)), \quad f_k \in \mathcal{F},$$

for some $\eta > 0$ (in our implementation we used $\eta=0.05$). If the optimal problem of value (11) is negative or zero, we know that no such $x^*$ exists. Otherwise we take its optimizer to be $x^*$.
where \( u(x_k) \) is determined as solution of (14). Assume that

1. there is a non-empty set \( T \subseteq \mathbb{R}^n \) and an \( \varepsilon > 0 \) such that \( L(x, u) \geq \varepsilon \cdot d(x, S) \) for all \( (x, u) \in L \), where \( d(x, T) := \inf_{y \in T} \|x - y\| \) is the distance of \( x \) from \( T \).
2. \( T \subseteq T(E) \).
3. there exists an \( s_0 \in \mathbb{R} \) with \( (x_0, s_0) \in T(E) \), such that \( s_0 := \{(x, s) \in T(E) \mid s \leq s_0\} \) is compact.

Then the closed loop is robustly asymptotically attracted by the set \( T \), i.e., \( \lim_{k \to \infty} d(x_k, T) = 0 \), for all system realizations \( (f_k)_{k \in \mathbb{N}} \).

**Remark 8.** Note that we could determine \( E \) from exact dynamic programming, \( E := T^N (E_N) \), starting with a terminal epigraph \( E_N \) that satisfies \( E_N \subseteq T(E_N) \). Then \( E \subseteq T(E) \) would follow directly from the monotonicity property of Proposition 2, and the above theorem generalizes existing stability results for robust MPC (Kerrigan & Maciejowski, 2004). Note, however, that it is very difficult to find such a positively invariant terminal epigraph \( E_N \) in practice, which is the reason why we avoid this assumption.

As a heuristic to directly generate a polyhedron \( E \) with the desirable property \( E \subseteq T(E) \), we propose to start with some arbitrary set \( E_N \) and perform \( N \) stages of the approximate polyhedral programming method of Section 4.2, and to set \( E := E_0 \). Recall that in each stage we set \( E^{out} = T(E_k) \) and let \( E_{k-1} \) satisfy \( E^{out} \subseteq E_{k-1} \). Therefore, the epigraph \( E_0 \) generated by our approximate method is contained in its exact counterpart, \( T^N (E_N) \), from the monotonicity property of Proposition 2. On the other hand, for large \( N \) we expect \( T^N (E_N) \) to be close to a set \( E_\ast \) satisfying \( E_\ast = T(E_\ast) \), independent of the chosen set \( E_N \). It is hoped that our approximate epigraph \( E_\ast \) is contained in \( E_\ast \), and that an application of the exact operator \( T \) on \( E_\ast \) enlarges the resulting set sufficiently to bring it closer to the fixed point \( E_\ast \), so that \( E_\ast \subseteq T(E_\ast) \), which is condition (2). The following example illustrates that our heuristic can deliver positively invariant epigraphs after a suitable number of recursion steps, although no guarantee can be given at present.

5. Tutorial example

We want to park a car with uncertain mass in front of a wall as fast as possible, without colliding with the wall, following the lines of Diehl and Björnberg (2004). The state \( x = \{x_p, p\} \) consists of position \( x \) and velocity \( v \), the control \( u \) is the acceleration force, constant on intervals of length \( t = 1 \). We define the following discrete time dynamics:

\[
x_{k+1} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x_k + \frac{t}{2m} \begin{pmatrix} 1 \\ 2 \end{pmatrix} u_k.
\]

The mass \( m \) of the car is only known to satisfy \( 1 \leq m \leq 1.5 \), i.e. we have a polytopic system (8) with uncertain matrix \( B \). We impose the constraints \( p \leq 0 \) and \( |u| \leq 1 \), and choose \( L(x, u) = \max(-p+v, -p-v) \), \( X_N = \{\{x_p\} \mid p \leq 0, v \leq 0, -p-v \leq 0.3\} \), \( Y_N = 100 \cdot (-p-v) \).

First, we computed the cost-to-go functions and feasible sets using the exact polyhedral DP method described in Diehl and Björnberg (2004). Computations took almost 4 h for \( N = 7 \). The number of facets defining \( E_0 \) was 467.

Then we repeated the computation using the approximate method of Section 4.2, where we chose \( E_N \) according to (13) with \( x = 1.1 \). Computations took about 8 min for \( N = 49 \); the robustly feasible sets \( X_N, X_{N-1}, \ldots, X_0 \) are plotted in Fig. 3. The number of facets defining \( E_0 \) was 105. For a more detailed comparison, we refer to Björnberg and Diehl (2004).

5.1. Stability

We denote by \( E \) the ultimate epigraph obtained by approximate robust DP and use the robust MPC feedback law \( u(\cdot) \) as defined in (14) to control the parking car. We show robust attractivity of the origin, i.e. we set \( \bar{T} = \{0\} \) in Theorem 7. By construction, conditions 1 and 3 are met whenever \( x_0 \in X_0 \). To test condition 2, i.e. \( E \subseteq T(E) \), we found the vertices of \( E \) and checked that they indeed satisfy the inequalities that define \( T(E) \) (within a tolerance of \( 10^{-6} \)). Thus, the approximate robust MPC leads the closed loop robustly towards the origin for all initial states \( x_0 \) in the set \( X_0 \) that is much larger than what could be obtained by the exact procedure in a reasonable computing time.

6. Conclusions

We have presented a method for approximate robust DP that can be applied to polytopic systems with piecewise affine cost and linear constraints. The underlying DP technique uses a dual approach and represents the cost-to-go functions and feasible sets at each stage in one single polyhedron \( E_k \). A generalization of the approximation technique proposed in Lincoln and
Rantzer (2002) to polyhedral sets allows us to represent these polyhedra approximately. We also presented an easily verifiable robust stability certificate for the min–max MPC that can be used together with the approximate DP recursion. We demonstrated in a tutorial example, a parking car with uncertain mass, that our heuristic approximation approach is indeed able to generate the positively invariant set required for this stability certificate. Comparing the results with the exact robust DP method used in Diehl and Björnberg (2004), we were able to demonstrate a significant ease of the computational burden. The presented algorithm, which is publicly available (Björnberg & Diehl, 2005), is able to yield much larger positively invariant sets than the exact approach and thus considerably widens the range of applicability of robust MPC schemes with guaranteed stability.

Acknowledgements

Financial support of the DFG under Grant BO864/10-1 is gratefully acknowledged. The authors also thank David Mayne for helpful comments and for encouraging publication of this work.

References


Jakob Björnberg received the B.Sc. degree in mathematics from the University of Warwick, UK, in 2005. During his summer vacations 2003–2005, he worked together with Moritz Diehl at the University of Heidelberg on problems in Model Predictive Control and Robust Dynamic Programming. He is currently studying for the Part III of the Mathematical Tripos at Cambridge University, UK. His other research interests include combinatorics and discrete probability.

Moritz Diehl is a C1 lecturer at the University of Heidelberg, where he had obtained his Ph.D. in numerical mathematics in 2001. He works on smart optimization methods for feedback control. Besides teaching and algorithm development, which he both loves, he is involved in several interdisciplinary research projects with engineering partners, among them “Optimization based control of chemical processes”, “Predictive control of car engines”, and “Control of fast robot motions”. Major focus of his work are fast real-time optimization methods for estimation and control.