Bilevel programming for analysis of low-complexity control of linear systems with constraints

Henrik Manum[†], Colin N. Jones[‡], Johan Löfberg[⋆], Manfred Morari[‡], and Sigurd Skogestad[†,*]

Abstract— In this paper we use bilevel programming to find the maximum difference between a reference controller and a low-complexity controller in terms of the infinity-norm difference of their control laws. A nominal MPC for linear systems with constraints, and a robust MPC for linear systems with bounded additive noise are considered as reference controllers. For possible low-complexity controllers we discuss partial enumeration (PE), Voronoi/closest point, triangulation, linear controller with saturation, and others. A small difference in the norm between a low-complexity controller and a robust MPC may be used to guarantee closed-loop stability of the low-complexity controller and indicate that the behaviour or performance of the low-complexity controller will be similar to that of the reference one. We further discuss how bilevel programming may be used for closed-loop analysis of model reduction.

Index Terms— bilevel programming, closed-loop analysis, optimal control

I. INTRODUCTION

Bemporad et. al. [1] introduced an explicit solution of the model predictive control (MPC) problem for control of *linear systems with constraints* using a quadratic performance index. Later these results have been extended to cover a broader class of systems and performance objectives, see [2] for a survey.

The main drawback of explicit MPC is that the control law, due to the combinatorial nature of the problem, can grow exponentially with the size of the optimal control problem [3].

Alessio and Bemporad [2] proposed to reduce complexity of explicit MPC by either storing only the L regions with the highest Chebysev radius (if a full explicit solution is available), or to run extensive simulations of closed-loop MPC and collect the L most recurrent combinations of active constraints for implementation, similar to [4]. (Storing only a subset of the possible regions of a MPC and using them for implementation is called partial enumeration (PE).)

Pannocchia et. al. [4] recently reported that by using a PE policy on an industrial example with more than 250 states, 32 inputs and a 25-sample control horizon, small look-up tables with only 25-200 entries gave a control that was less than 0.01% suboptimal compared to the full model predictive

controller (MPC) for the same example. The MPC could theoretically enter $3^{800} = 4.977 \times 10^{381}$ regions.

In this paper we use bilevel programming to investigate the PE-schemes described above, but also more general low-complexity policies. The main idea is to calculate the maximum difference between a either a nominal or a robust MPC and the low-complexity policy, and then, based on this difference, draw conclusions about the proposed low-complexity controller.

In addition to guarantees of feasibility and stability the method can be used to give bounds on the sub-optimality of the low-complexity scheme, by using the value of the objective function of the reference controller as a difference-metric of the reference and low-complexity controller.

II. NOTATION AND PRELIMINARIES

A *polyhedron* is the intersection of a finite number of halfspaces and a *polytope* is a bounded polyhedron. Given two sets $S_1, S_2 \subseteq \mathbb{R}^n$ the Minkowski sum is defined as $S_1 \oplus S_2 \triangleq \{s_1 + s_2 | s_1 \in S_1, \ s_2 \in S_2\}$, and the Pontryagin difference as $S_1 \ominus S_2 \triangleq \{s_1 | s_1 + s_2 \in S_1, s_2 \in S_2\}$. Bold-face $\mathbf{x}$ and $\mathbf{u}$ means the sequences $\mathbf{x} = (x_0, x_1, \ldots, x_N)$ and $\mathbf{u} = (u_0, u_1, \ldots, u_{N-1})$, while boldface $\mathbf{1}$ is a vector of 1's of appropriate length.

We consider control of the following discrete-time linear system

$$x^+ = Ax + Bu, \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ are the states and $u \in \mathbb{R}^{n_u}$ are the inputs, and x^+ above is a short-hand notation for $x_{k+1} = Ax_k + Bu_k$. In addition we have constraints such that $x \in \mathbb{X}$ and $u \in \mathbb{U}$, where $\mathbb{X} = \{x \mid Fx \le f\} \subset \mathbb{R}^{n_x}$ and $\mathbb{U} = \{u \mid Gu \le g\} \subset \mathbb{R}^{n_u}$ are polytopic sets.

The solution of an explicit MPC with quadratic objective, linear process and polytopic constraints, can be written as a piecewise affine function of the state. A piecewise affine function $u(x) : X \mapsto \mathbb{R}^{n_u}$, where $X \subset \mathbb{R}^{n_x}$ is a polyhedral set, is piecewise affine if it is possible to partition X into convex polyhedral regions, CR_i, and $z(x) = K^i x + c^i, \quad \forall x \in CR_i$ [1]. In this paper "region" denotes CR_i, written "region i", and (K^i, c^i) is the corresponding optimal control law, i.e. the part of $u(x)$ that belongs to CR_i. In order to conform with notation used in [2], we use $L_i = \{x \in \mathbb{R}^{n_x} \mid A^i x \le b^i\}$ in the place of CR_i.

III. BILEVEL OPTIMIZATION

The main focus of this paper is the application of bilevel optimization for analysis of low-complexity controllers. Here we give an introduction to bilevel optimization and solution

[†]H. Manum and S. Skogestad are with Department of Chemical Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway.

[‡]C. Jones and M. Morari are with ETH Zürich, Automatic Control Laboratory, ETL I28, Physikstrasse 3, 8092 Zürich, Switzerland.

[⋆]J. Löfberg is with Department of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden.

[*]Author to whom correspondence should be addressed. Email: skoge@ntnu.no

methods, following [5]. For more background details the reader is referred to a recent survey [6].

Bilevel problems are hierarchical in that the optimization variables (y, z) are split into upper y and lower z parts, with the lower level variables constrained to be an optimal solution to a secondary optimization problem:

$$
\begin{aligned}
\min_y \ & V_U(y, z) \\
\text{subject to } & G_U(y, z) \leq 0 \\
& z = \arg\min_z V_L(y, z) \\
& \qquad \text{subject to } G_L(y, z) \leq 0
\end{aligned}
\tag{2}
$$

In this paper we will only consider problems where the lower-level problem has an unique optimizer. Moreover, we will usually have two low-level problems, one for the reference controller and one for the low-complexity controller.

A. Solution methods

If the lower level problem is convex and regular, then it can be replaced by its necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions, yielding a standard single-level optimization problem [5]:

$$
\begin{aligned}
\min_{y, z, \lambda} \ & V_U(y, z) \\
\text{subject to } & G_U(y, z) \leq 0 \\
& G_L(y, z) \leq 0 \\
& \lambda \geq 0 \\
& \lambda^{\mathrm{T}} G_L(y, z) = 0 \\
& \nabla_z \mathcal{L}(y, z, \lambda) = 0
\end{aligned}
\tag{3}
$$

where $\mathcal{L}(y, z, \lambda) := G_L(y, z) + \lambda^{\mathrm{T}} G_L(y, z)$ is the Lagrangian function associated with the lower-level problem. For the special case of linear constraints and a quadratic cost, all constraints of (3) are linear and the complimentary condition $\lambda^{\mathrm{T}} G_L(y, z) = 0$ is a set of disjunctive linear constraints, which can be described using binary variables, and thus leads to a mixed-integer linear problem.

B. Bilevel optimization for analysis of controllers

In this paper we use bilevel programming to find the maximal difference between a reference controller and a low-order controller. Hence, for a subset $\mathcal{X} \subset \mathbb{R}^{n_x}$, we solve

$$
\begin{aligned}
\max_{x \in \mathcal{X}} \ & d(u_{\text{ref}}, u_{\text{low-complexity}}) \\
\text{subject to } & \text{KKT(reference controller)} \\
& \text{KKT(low-complexity controller)}
\end{aligned}
\tag{4}
$$

Typically, $\mathcal{X}$ is the intersection of the feasible states for the reference and the low-complexity controller.

Note that explicit solutions of neither the reference nor the low-complexity controllers are needed, because the solutions are implicitly given by the KKT conditions.

The distance measure $d(u_{\text{ref}}, u_{\text{low-complexity}})$ can be, for example, the difference between the next state,

$$
\begin{aligned}
& d(u_{\text{ref}}, u_{\text{low-complexity}}) = \\
& \| x^+{}_{\text{ref}}(x, u_{\text{ref}}) - x^+{}_{\text{low-complexity}}(x, u_{\text{low-complexity}}) \|_\infty = \\
& \| B u_{\text{ref}} - B u_{\text{low-complexity}} \|_\infty,
\end{aligned}
\tag{5}
$$

but also differences between trajectories of either states or inputs.

Remark 1: We observe that (5) renders (4) non-convex due to the term $\max \|t\|_\infty$ (where t is a convex function of $(u_{\text{ref}}, u_{\text{low-complexity}})$) . However, the problem may be converted into a mixed integer linear program (MILP) using a standard technique (e.g. [7]), in which we introduce binary variables n_i, p_i for each element of t and add the condition that the binary variable p_i is one if $\|t\|_\infty = t_i$ and n_i is one if $\|t\|_\infty = -t_i$. The method adds only linear and binary conditions to (4) and therefore the overall problem remains a MILP [5].

IV. APPLICATIONS OF BILEVEL PROGRAMMING FOR ANALYSIS OF LOW-COMPLEXITY CONTROLLERS

We first present a nominal MPC policy based on optimizing a quadratic performance objective subject to a linear model of the process at and a set of polytopic constraints on both states and inputs. We thereafter present a robust MPC, where the process is subject to bounded disturbances on the states. Both these schemes fit into the bilevel problem as a reference controller.

The choice of which reference controller to use depends on the problem at hand, as this defines a benchmark for control of the process. The robust MPC scheme can be used to give a feasibility and stability certificate of the low-complexity scheme. However, in some cases the robust MPC can be quite conservative, and the nominal MPC may be a better benchmark.

Thereafter we show how several low-complexity polices can be expressed in the bilevel framework. The main "tool" we use here is to represent any logic and bilinear terms in the KKT-conditions with mixed integer linear constraints in order to let the resulting problem be a MILP.

A. Nominal MPC as reference controller

Consider the following semi-infinite horizon optimal control problem [5]:

$$
\begin{aligned}
\min_{\mathbf{x}, \mathbf{u}} \ & J(\mathbf{x}, \mathbf{u}) = \frac{1}{2} x_N^{\mathrm{T}} P x_N + \dots \\
& \qquad\qquad + \frac{1}{2} \sum_{i=0}^{N-1} u_i^{\mathrm{T}} R u_i + x_i^{\mathrm{T}} Q x_i, \\
\text{subject to } & x_{i+1} = A x_i + B u_i, \quad \forall i = 0, \dots, N-1, \\
& x_i \in \mathbb{X}, \quad \forall i = 1, \dots, N-1, \\
& u_i \in \mathbb{U}, \quad \forall i = 0, \dots, N-1, \\
& x_N \in \mathbb{X}_N, \\
& x_0 = x.
\end{aligned}
\tag{6}
$$

Here $\mathbb{X}_N = \{ x \mid Hx \leq h \} \subset \mathbb{X}$ is a polytopic invariant set for the system $x^+ = Ax + B\mu(x)$ for some given control law $\mu : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_u}$. Further $P \in \mathbb{R}^{n_x \times n_x}$ and $Q \in \mathbb{R}^{n_x \times n_x}$ are positive definite matrices and $R \in \mathbb{R}^{n_u \times n_u}$ is a positive semi-definite matrix. We define $\mathcal{X} \subset \mathbb{R}^{n_x}$ to be the set of states x for which there exists a feasible solution to (6).

If $\mathbf{u}^*(x)$ is the optimal input sequence of (6) for the state x, and $u_0^*(x)$ is the resulting control law, then stability of the system $x^+ = Ax + Bu_0^*(x)$ can be established under the assumption that $V_N(x) = x^\mathrm{T} P x$ 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 $l(u,x) = u^\mathrm{T} R u + x^\mathrm{T} Q x$ within the set $\mathbb{X}_N$ [5].

By using $x_k = A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$ the MPC problem (6) can be rewritten as [1]:

$$V(x_0) = \frac{1}{2} x_0^\mathrm{T} Y x_0 + \dots$$
$$+ \min_U \ \{ \frac{1}{2} U^\mathrm{T} H U + x_0^\mathrm{T} F U, \tag{7}$$
$$\text{subject to} \quad GU \leq W + E x_0 \},$$

where $U^\mathrm{T} = \begin{bmatrix} u_0^\mathrm{T} & u_1^\mathrm{T} & \cdots & u_{N-1}^\mathrm{T} \end{bmatrix}$.

We want to use (7) as a lower-level problem in bilevel programming. The following equations define the KKT conditions for this problem:

$$HU + F^\mathrm{T} x_0 + G^\mathrm{T} \lambda = 0$$
$$GU - W - E x_0 \leq 0$$
$$\lambda \geq 0 \tag{8}$$
$$\lambda \leq Ms$$
$$GU - W - E x_0 \geq -M(1-s)$$

Here $s \in \{0,1\}^{n_W}$, where n_W is the number of inequality constraints in (7). The two last equations in (8) correspond to the complementary condition $\lambda^\mathrm{T} G_L(y,z) = 0$ in the general bilevel problem, here described with binary variables s. M is a constant that is large enough such that the solution to (8) corresponds to the solution of (7). (This is called a "big-M" formulation.)

B. Robust MPC as reference controller

In this subsection the results are from Mayne et. al. [8] unless otherwise noted.

Consider control of the linear system (1) with *additive disturbances* w on the states:

$$x^+ = Ax + Bu + w. \tag{9}$$

The disturbance is assumed to be bounded,

$$w \in W, \tag{10}$$

where W is compact and contains the origin (but may not have an interior).

Suppose $K \in \mathbb{R}^{n_u \times n_x}$ is such that $A_K \triangleq A + BK$ is stable. Let Z be a disturbance invariant set for the controlled uncertain system $x^+ = A_K x + w$ satisfying, therefore

$$A_K Z \oplus W \subseteq Z. \tag{11}$$

We use the following proposition as a basis for the robust MPC:

Proposition 1: Suppose Z is disturbance invariant for $x^+ = A_K x + w$. If $x \in \bar{x} \oplus Z$ and $u = \bar{u} + K(x - \bar{x})$, then $x^+ \in \bar{x}^+ \oplus Z$ for all $w \in W$ where $x^+ = Ax + Bu + w$ and $\bar{x}^+ = A\bar{x} + B\bar{u}$.

Proposition 1 states that the feedback policy $u = \bar{u} + K(x - \bar{x})$ keeps the states x of the uncertain system (9) close to the states $\bar{x}$ of the so-called nominal system $\bar{x}^+ = A\bar{x} + B\bar{u}$.

We can now define the robust MPC problem:

$$\min_{\bar{\mathbf{x}}, \bar{\mathbf{u}}} \ J(\bar{\mathbf{x}}, \bar{\mathbf{u}}) = \frac{1}{2} \bar{x}_N^\mathrm{T} P \bar{x}_N + \dots$$
$$+ \frac{1}{2} \sum_{i=0}^{N-1} \bar{u}_i^\mathrm{T} R \bar{u}_i + \bar{x}_i^\mathrm{T} Q \bar{x}_i,$$
$$\text{subject to } \bar{x}_{i+1} = A\bar{x}_i + B\bar{u}_i, \quad \forall i = 0, \dots, N-1, \tag{12}$$
$$\bar{x}_i \in \mathbb{X} \ominus Z, \quad \forall i = 1, \dots, N-1,$$
$$\bar{u}_i \in \mathbb{U} \ominus KZ, \quad \forall i = 0, \dots, N-1,$$
$$\bar{x}_N \in \mathcal{X}_f,$$
$$\bar{x}_0 = x \oplus Z.$$

In order to achieve closed loop robust stability, the terminal constraint set $\mathcal{X}_f$ must satisfy the following axioms [8]:

$$\text{A1} : A_K \mathcal{X}_f \subset \mathcal{X}_f, \ \mathcal{X}_f \subset \mathbb{X} \ominus Z, \ K\mathcal{X}_f \subset \mathbb{U} \ominus KZ$$
$$\text{A2} : V_f(A_k x) + l(x, Kx) \leq V_f(x), \ \forall x \in X_f, \tag{13}$$

where $V_f(v) = v^\mathrm{T} P v$ and $l(v,z) = v^\mathrm{T} Q v + u^\mathrm{T} R u$ in the scope of this paper.

Assume that Z is a polytopic set such that $\{v \in \mathbb{R}^{n_x} \mid H_z v \leq k_z\}$.

As for the nominal MPC, we can rewrite the robust MPC problem as:

$$\min_{(U, \bar{x}_0)} \begin{bmatrix} U^\mathrm{T} \\ \bar{x}_0^\mathrm{T} \end{bmatrix}^\mathrm{T} \underbrace{\begin{bmatrix} H & F^\mathrm{T} \\ F & 2Y \end{bmatrix}}_{\tilde{H}} \begin{bmatrix} U \\ \bar{x}_0 \end{bmatrix}$$
$$\text{subject to} \ \underbrace{\begin{bmatrix} G & -E \\ 0 & -H_z \end{bmatrix}}_{\tilde{G}} \begin{bmatrix} U \\ \bar{x}_0 \end{bmatrix} \leq \underbrace{\begin{bmatrix} W \\ k_z \end{bmatrix}}_{\tilde{W}} + \underbrace{\begin{bmatrix} 0 \\ -H_z \end{bmatrix}}_{\tilde{E}} x \tag{14}$$

Let $v = (U, \bar{x}_0)$. The KKT-conditions corresponding to (14) are

$$\tilde{H} v + \tilde{G} \lambda = 0$$
$$\tilde{G} v \leq \tilde{W} + \tilde{E} x$$
$$\lambda \geq 0 \tag{15}$$
$$\lambda \leq Ms$$
$$\tilde{G} v \geq W + Ex - M(1-s)$$

Note that the KKT conditions in (8) are a special case of the KKT-conditions above, since above $\bar{x}_0$ is included as a degree of freedom. For both nominal and robust MPC the current state x is a parameter driving the controller, but for the nominal MPC we have substituted this with x_0, as $x_0 = x$ is a constraint in the nominal MPC formulation.

The main motivation for using robust MPC as a reference rather than nominal MPC is because the robust MPC can be used to prove feasibility and stability of the low-complexity

scheme. Both properties can be established using the following proposition:

Proposition 2: Consider the linear system for which robust stability and feasibility are guaranteed by the robust MPC:

$$x^+ = Ax + Bu + w, \quad w \in W,$$

and that

$$W = \{w \in \mathbb{R}^{n_x} \mid \|w\|_\infty \le \varepsilon\}$$

Let $u_{\text{l-c}}$ be the control input from the low-complexity controller, and u_{rMPC} the input from the robust MPC. The following holds for the system controlled by the low-complexity controller:

$$
\begin{aligned}
x^+ &= Ax + Bu_{\text{l-c}} \\
&= Ax + Bu_{\text{l-c}} - Bu_{\text{rMPC}} + Bu_{\text{rMPC}} \qquad (16) \\
&= Ax + Bu_{\text{rMPC}} + B(u_{\text{l-c}} - u_{\text{rMPC}}).
\end{aligned}
$$

Hence, if

$$\|B(u_{\text{l-c}} - u_{\text{rMPC}})\|_\infty \le \varepsilon, \qquad (17)$$

the low-complexity controller is both feasible and stable.

C. Low-complexity controllers as low-level problems in bilevel programming

In this section we describe various low-complexity controllers that fit into the bilevel programming framework. Several more are possible, but not included for space restrictions.

1) Linear quadratic regulator with saturation: A simple low-complexity control policy is the linear quadratic regulator (LQR) with saturation. In the "unconstrained region" this is optimal, and its behaviour can be modelled using few binary variables. First, we define $\hat{u}_{\text{LQR}} = -Kx$. For simplicity we assume that the constraints on u may be written as

$$u_i^l \le u_i \le u_i^h, \quad i = 1, \dots, n_u \qquad (18)$$

Now, for each row in (18), we define a corresponding binary vector $d^i \in \{0,1\}^3$. The saturation can now be modelled using

$$
\begin{aligned}
u_i &\le u_i^h + M d_1^i, \\
u_i &\ge u_i^l - M d_3^i, \\
d_1^i + d_2^i + d_3^i &= 1, \qquad (19) \\
-M(1 - d_k^i) \le \text{sat}(u_i) &- \{u_i\}_k \le M(1 - d_k^i), \\
k &= 1, 2, 3,
\end{aligned}
$$

where $\{u_i\} = \{u_i^h, u_i, u_i^l\}$, and $\{u_i\}_k$ is the k'th element of $\{u_i\}$.

2) Partial enumeration (PE): Here we follow the ideas of [4] and [2], and we store only a subset of the possible active sets. The controller implementation is here to first locate the closest region to the current state x, and then use the control law from the corresponding region. In order to satisfy $u \in \mathbb{U}$, we saturate the input before applying the input to the plant.

Here we use the *minimal-violation distance* from Christophersen et. al. [9] to find the closest region for a set $\mathcal{L}$ of stored polytopes.

Definition 1: (Minimal-violation distance [9]) Let the collection $\mathcal{L}$ be the set $\mathcal{L} = \{L_i\}_{i=1}^{N_{\mathcal{L}}}$, where $L_i := \{x \in \mathbb{R}^{n_x} \mid A^i x \le b^i\}$ are full-dimensional polyhedra in $\mathbb{R}^{n_x}$. We assume that $A^i x \le b^i$ are on Hessian normal form, i.e. each row $[A^i]_r$ of A^i is normalized with $\|[A^i]_r\|_2 = 1$.

The minimal-violation distance d_{MV} of x to $\mathcal{L}$ is given by

$$d_{\text{MV}} := \min_i \{\alpha_i^*(x)\}, \qquad (20)$$

where

$$\alpha_i^*(x) = \arg\min \{\alpha_i \in \mathbb{R} \mid A^i x \le b^i + \alpha_i \mathbf{1}\}, \qquad (21)$$

for all $i = 1, \dots, N_{\mathcal{L}}$ and $\mathbf{1} = [1 \ \cdots \ 1]^{\text{T}}$.

The solution of the LP (21) can be found using the KKT conditions:

$$
\begin{aligned}
1 - \mathbf{1}^{\text{T}} \lambda^i &= 0, \\
0 \le \lambda^i &\le M s^i, \qquad (22) \\
0 \le b + \alpha_i \mathbf{1} - A^i x &\le M(\mathbf{1} - s^i),
\end{aligned}
$$

where $s^i \in \{0,1\}^{n_{bi}}$ is a vector of binary variables of length corresponding to the number of faces in the polytope $L_i = \{x \in \mathbb{R}^{n_x} \mid A^i x \le b^i\}$.

Let $\beta \in \{0,1\}^{n_{\mathcal{L}}}$ be binary variables such that

$$\beta_i = 1 \leftrightarrow \alpha_i \le \alpha_j \quad \forall j \ne i, \qquad (23)$$

which implies that $\sum \beta_i = 1$. We can then define the PE control law as

$$\hat{u} = \text{sat}\left\{\sum_{i=1}^{n_{\mathcal{L}}} \beta_i \left(K^i x + c^i\right)\right\}, \qquad (24)$$

where (K^i, c^i) is the optimal feedback in region i, and $\text{sat}\{\cdot\}$ is a normal saturation function. Equation (24) is bilinear in the optimization variables β_i, x, and can be implemented in the bilevel framework with the following equations (added as constraints in the problem):

$$-M(1 - \beta_i) \le \hat{u} - (K^i x + c^i) \le M(1 - \beta_i). \qquad (25)$$

Remark 2: The proposed PE-scheme, which follows from [9], can be implemented on-line as follows:

$$
\begin{aligned}
\alpha_i &= \max\{A^i x - b^i\}, \quad i = 1, \dots, L \\
i^* &= \arg\min_i \{\alpha_i\}
\end{aligned} \qquad (26)
$$

3) Delaunay triangulation: Assume that for some points $(x_1, \dots, x_{n_L})$ we precompute a Delaunay triangulation. In addition we store the optimal input $(u_1^*, \dots, u_{n_L}^*)$ at those points. A Delaunay triangulation can be understood by the empty circle method [10]: Consider all triangles formed by the points such that the circumcircle of each triangle is empty of other sites, where the sites in this case are the stored points $(x_1, \dots, x_{n_L})$.

The Delaunay triangulation of the points $(x_1, \dots, x_{n_D})$ can be used to find an interpolated control law:

- Denote the triangles from the Delaunay triangulation by $L_1, \dots, L_{n_D}$.
- For a given state x:
  1) Find the current triangle L_i that containts x.

2) Express x as a convex combination of the vertices of L_i, $x = \sum \lambda_k x_k^i$, where x_k^i denotes the vertices of L_i

- Implement the following interpolated control law:

$$u_{\text{Delaunay}} = \sum \lambda_k u_k^{*,i}, \qquad (27)$$

where $u_k^{*,i}$ are the optimal inputs corresponding to the points x_k^i.

The Delaunay triangulation itself can be implicitly defined using the following set of equations, which can be added as mixed-integer linear constraints to the overall problem:

$$x = \sum \lambda_i x_i, \quad , \lambda_i \geq 0, \quad \sum \lambda_i = 1,$$
$$\lambda \leq \sigma_i, \quad \sum \sigma_i = n+1 \qquad (28)$$
$$\|c - x_i\|_2^2 \leq \|c - x_j\|_2^2 + M\sigma_j + M(1 - \sigma_i),$$

where the last equation can be rewritten as

$$\underbrace{c^{\mathsf{T}}c - 2x_i^{\mathsf{T}}c}_{a_i^{\mathsf{T}}} + \underbrace{x_i^{\mathsf{T}}x_i}_{b_i} \leq \underbrace{c^{\mathsf{T}}c - 2x_j^{\mathsf{T}}c}_{a_j^{\mathsf{T}}} + \underbrace{x_j^{\mathsf{T}}x_j}_{b_j} + \dots$$
$$\dots + M\sigma_j + M(1 - \sigma_i) \qquad (29)$$
$$a_i^{\mathsf{T}}c + b_i \leq a_j^{\mathsf{T}}c + b_j + M\sigma_j + M(1 - \sigma_i)$$

Here $c \in \mathbb{R}^{n_x}$ is an extra optimization variable, $\sigma \in \{0,1\}^{n_L}$ is a vector of binaries and M is a large constant.

We note that the last equation of (28) is an expression for the "empty-circle method".

V. EXAMPLES

In this section we show two examples where we use the bilevel programming to identify the worst-case distance between a reference controller and a proposed low-complexity controller. The calculations where done using ILOG CPLEX® and the problems were written in YALMIP [7]. Set calculations and explicit solution of MPC's were done using Multi-Parametric Toolbox (MPT) [11].

Example 1: Double integrator with nominal MPC as reference controller and PE as low-complexity controller

In this example we consider the double integrator described in [1], example 7.3, but with a sample time of $T_s = 0.1$ in order to match the conditions in [2]. The process is hence

$$x^+ = \underbrace{\begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} 0 \\ 0.1 \end{bmatrix}}_{B} u, \quad -1 \leq u \leq 1 \qquad (30)$$

The control parameters are $N = 8$, $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ and $R = 0.1$. The final weight P corresponding to the LQR controller is $P = \begin{bmatrix} 8.98 & 3.59 \\ 3.59 & 2.86 \end{bmatrix}$.

The nominal MPC problem is now:

$$\min_{\mathbf{x}, \mathbf{u}} x_8^{\mathsf{T}} P x_8 + \sum_{i=0}^{7} x_i^{\mathsf{T}} Q x_i + R u_i^2$$

subject to $x_{k+1} = A x_k + B u_k, k = 0, \dots, 7 \qquad (31)$
$$x_0 = x$$
$$-1 \leq u_k \leq 1, \quad k = 0, 1 \dots, 7$$

We do not add any terminal constraint on x_N as we want to compare our results with [2].

We want to compare the nominal MPC to a PE-scheme, hence we want to solve

$$\max_{x \in \mathcal{X}} \|B(u^* - \hat{u})\|_\infty$$

subject to $\alpha_i = \arg \min_\alpha \alpha$

subject to $A^i x \leq b^i + \alpha_i \mathbf{1}$

$$\beta_i = \begin{cases} 1, & \alpha_i \leq \alpha_j \ \forall j \neq i \\ 0, & \text{otherwise} \end{cases} \qquad (32)$$
$$\tilde{u} = \sum_{i=1,\cdots,L} \beta_i (K^i x + c^i),$$
$$\hat{u} = \text{sat}(\tilde{u})$$
$$u^* = \arg \min (31)$$

This problem can be rewritten to a MILP using (22) for the minimal violation distance.

The main focus of this paper is to calculate the difference between two controllers, but we may also use this method for controller synthesis. This can be achieved by:

- Solve (32) to get the worst point in the state space x^* and the worst case norm $\|B(u^* - \hat{u})\|_\infty = \|x^{*,+} - \hat{x}^+\|_\infty$.
- Add the corresponding region and corresponding optimal control law to the PE-controller.
- Resolve (32) and add the corresponding worst-case region until the worst-case norm is less than a user-defined value or the number of regions in the PE is larger than a user-defined value.

This example can be solved explicitly using MPT. The full enumeration is shown in the upper right part of figure 1. In order to test our software we started out with an initial PE controller using the 3 largest regions, shown in the top-left part of figure 1. The lower part of the figure shows the maximum difference between the reference controller (nominal MPC) and the PE-controller. We then performed iterations as described above, at each iteration we added the region corresponding to the worst case point x^*. One observes that initially the difference is equal to the maximum possible difference, as $B = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$ and $\|u\| \leq 1$. However, as we add regions to the PE controller the difference decreases to quite low levels.

Note that even though the full enumeration was available for this example, we do not use this solution while solving (32), rather we use the KKT-conditions of the corresponding MPC problem.

Closed-loop simulations, even from the worst case points, shows very small difference between the nominal MPC and the PE, also for quite high values of the worst-case norm, and are not included here for brevity.

Example 2: Double integrator with robust MPC as reference controller

For the same process as in Example 1, with the same objectives for the controller, we designed a robust MPC using
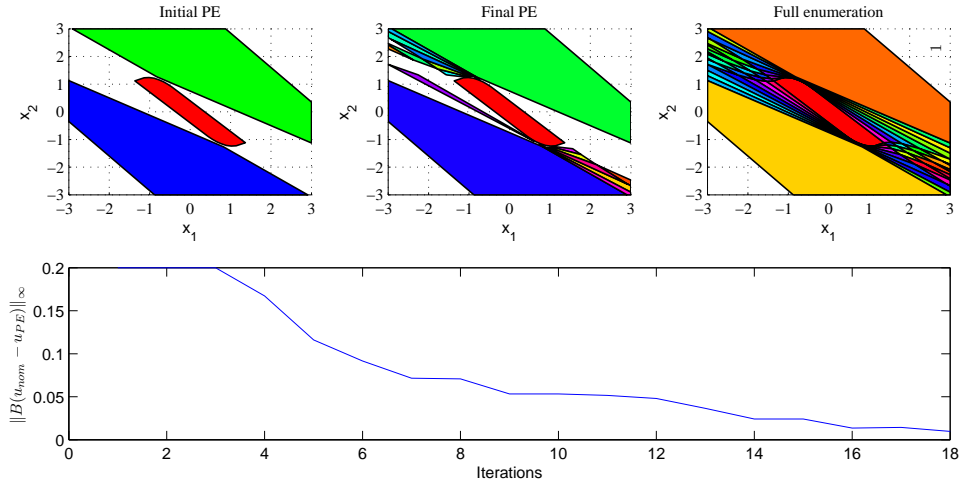
Fig. 1. Example 1: double integrator.

the method described in section IV-B, and we use this one as the reference controller. The motivation for using the robust MPC rather than the nominal MPC is because we can verify closed-loop stability of the low-complexity scheme, given that $\|B(u_{\text{robust}} - u_{\text{low-complexity}})\|_\infty \leq \|w\|_\infty$.

A box constraint on w was used such that $\|w\|_\infty \leq 0.01$, and we used the algorithm from Rakoviç et. al. [12] to compute Z, and in order to compute X_f we used MPT. We wanted to use this robust controller to prove closed-loop nominal stability of the PE-controller from Example 1. However, we observed that $\max_{x \in \mathcal{X}} \ \|B(u_{\text{robust MPC}} - u_{\text{nominal MPC}})\|_\infty$ was growing faster than $\|w\|_\infty$, i.e. the robust MPC was very conservative with increasing $\|w\|_\infty$. Since the PE-controller from Example 1 is close to the nominal MPC, it is clear that we cannot use the robust MPC scheme to prove stability of the PE-scheme, moreover we can not even use it to prove closed-loop stability of the nominal MPC.

One reason for why $\|B(u_{\text{robust MPC}} - u_{\text{nominal MPC}})\|_\infty$ is growing faster than $\|w\|_\infty$ is that the scalar input u can only act on the process in the direction B, while the vector w is acting directly on both states (through the identity transformation I). Changing the formulation of the robust MPC to restrict w to act only in the direction B is planned as further work in this project.

VI. CONCLUSIONS

A bilevel framework for closed loop comparison of different control schemes has been presented. Many challenges still remain, but it seems like this framework will be useful for proving stability for some "ad-hoc" low complexity control schemes, and moreover it seems to have potential in the field of model reduction.

ACKNOWLEDGEMENTS

REFERENCES

[1] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, pp. 3–20, 2002, see also corrigendum 39(2003), pages 1845-1846.

[2] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," in *Proc. Int. Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, Pavia, Italy, 2008.

[3] C. Wen, X. Ma, and B. Ydstie, "Analytical expression of explicit mpc solution via lattice piecewise-affine function," *Automatica*, no. 45, pp. 910–917, 2009.

[4] G. Pannocchia, J. B. Rawlings, and S. J. Wright, "Fast, large-scale model predictive control by partial enumeration," *Automatica*, vol. 43, pp. 852–860, May 2007.

[5] C. Jones and M. Morari, "Approximate Explicit MPC using Bilevel Optimization," in *European Control Conference*, Budapest, Hungary, Aug. 2009. [Online]. Available: http://control.ee.ethz.ch/index.cgi?page=publications;action=details;id=3339

[6] B. Colson, P. Marcotte, and G. Savard, "Bilevel programming: A survey," *4OR: A Quarterly Journal of Operations Research*, vol. 3, no. 2, pp. 87–107, June 2005.

[7] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in MATLAB," in *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. [Online]. Available: http://control.ee.ethz.ch/ joloef/yalmip.php

[8] D. Mayne, M. Seron, and S. Raković, "Robust model predictive control of constrained linear systems with bounded disturbances," *Automatica*, vol. 41, pp. 219–224, 2005.

[9] F. J. Christophersen, M. N. Zeilinger, C. N. Jones, and M. Morari, "Controller Complexity Reduction for Piecewise Affine Systems Through Safe Region Elimination," in *Proc. of the Conf. on Decision & Control*, New Orleans, USA, Dec. 2007.

[10] F. Aurenhammer, "Voronoi diagrams - A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys*, vol. 23, no. 3, September 1991.

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

[12] S. Rakoviç, E. Kerrigan, K. Kouramas, and D. Mayne, "Invariant approximations of the minimal robust positively invariant set," *IEEE Transactions on Automatic Control*, vol. 50, no. 3, pp. 406–410, March 2005.