G Model
CACE-4761; No. of Pages 10

ARTICLE IN PRESS

Computers and Chemical Engineering xxx (2013) xxx–xxx

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

On the design of exact penalty functions for MPC using mixed integer programming

Morten Hovd*, Florin Stoican

Norwegian University of Science and Technology (NTNU), Department of Engineering Cybernetics, Trondheim, Norway

ARTICLE INFO

Article history:
Received 30 November 2012
Received in revised form 27 June 2013
Accepted 3 July 2013
Available online xxx

Keywords:
Soft constraints
Exact penalty functions
Lagrange multipliers
Mixed integer programming
LICQ

ABSTRACT

Soft constraints and penalty functions are commonly used in MPC to ensure that the optimization problem has a feasible solution, and thereby avoid MPC controller failure. On the other hand, soft constraints may allow for unnecessary violations of the original constraints, *i.e.*, the constraints may be violated even if a valid solution that does not violate any constraints exists.

The paper develops procedures for the minimizing (according to some norm) of the Lagrange multipliers associated with a given mp-QP problem, assumed to originate from an MPC problem formulation. To this end the LICQ condition is exploited in order to efficiently formulate the optimization problem, and thereby improve upon existing mixed integer formulations and enhance the tractability of the problem. The results are used to design penalty functions such that corresponding soft constraints are made exact, that is, the original (hard) constraints are violated only if there exists no solution where all constraints are satisfied.

1. Introduction

Model Predictive Control (MPC) has been a remarkable industrial success, with thousands of installations worldwide (Qin & Badgwell, 2003). A distinguishing feature of MPC controllers is the relative ease with which constraints in both states/outputs and inputs are handled. Nevertheless, such constraints may introduce many complexities that an industrial MPC controller needs to address. There has been particular focus on the effect of hard output constraints on stability (Zafiriou & Marchal, 1991; de Oliveira & Biegler, 1994) as well as the use of soft constraint formulations to ensure a feasible optimization problem, see Scokaert and Rawlings (1999), Vada (2000), Hovd and Braatz (2001a) and references therein.

This paper advances the ideas shown in Hovd (2011). Using the KKT conditions derived from the MPC problem conditions are given which make the penalty function exact for a “soft” set of constraints. These conditions are formulated using polyhedral norms over the Lagrange multipliers of the original problem. This is a step further with respect to the usual approach where only the ℓ_1 and ℓ_∞-norm cases are discussed, see, e.g. Kerrigan and Maciejowski (2000).

Explicit formulations like the one in Bemporad, Morari, Dua, and Pistikopoulos (2002) provide a piecewise description of the Lagrange multipliers but the calculations quickly become intractable for high dimension and/or a large set of constraints defining the feasible region. Here, the implicit description of the problem is kept and the bi-level programming formalism is used to recast it into a mixed integer formulation. This allows a direct search for the worst-case values of the Lagrangian multipliers - albeit a search formulated as a mixed integer linear program.

The Lagrange multipliers might not be uniquely defined for a given combination of states/inputs. In Hovd (2011) this was solved by adding additional minimization subproblems such that the minimum multiplier (after the ℓ_2 nom) is selected from the space of available solutions. Here, the selection of active constraints is restricted such that only linearly independent combinations are considered (the LICQ condition), which in turn means that the Lagrange multipliers are uniquely determined and there is no need to add another level of minimization. The numerical aspects are also analyzed and it is noted how and under which conditions numerical errors can be minimized and the computation time reduced.

The rest of the paper is organized as follows. An introduction to bi-level programming is given in Section 2. Preliminaries on MPC formulation and soft constraints in MPC are discussed in Section 3 and in Section 4 the conditions for exact penalty functions are given. In Section 5 the conditions are posed in the mixed-integer formalism and various optimizations are studied. Lastly, some illustrative examples are shown in Section 6 and conclusions are drawn in Section 7.

1.1. Notation

For a vector $t \in \mathbb{R}^d$, $(t)_i$ denotes the *ith* element. The vector $\mathbf{1}_d \in \mathbb{R}^d$ is defined as $[1 \dots 1]^T$ and the identity matrix of order N

* Corresponding author. Tel.: +47 73591426.
E-mail address: morten.hovd@itk.ntnu.no (M. Hovd).

0098-1354/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.compchemeng.2013.07.001

is denoted as $I_N \in \mathbb{R}^{N\times N}$. The specification of the dimension may be dropped whenever it is clear from the context. The collection of all possible combinations of N binary variables will be noted $\left\{0,1\right\}^N = \left\{(b_1,\ldots,b_N) : b_i \in \left\{0,1\right\}, \forall i = 1,\ldots,N\right\}$. A finite intersection of inequalities which describes a non-empty region is called a polyhedral set. A polytope is a bounded polyhedral set. The cardinality of a set A is denoted as $\# A$.

Throughout the paper, λ represents Lagrangian multipliers for inequality constraints, ν are Lagrangian multipliers for equality constraints, and s are binary variables used to linearize the complementarity constraints of the KKT conditions. Additional nomenclature is introduced when necessary where it is first used. The reader is assumed to be familiar with nomenclature commonly used in systems theory and MPC literature.

2. Bi-level programming

In bi-level programming, the constraints of the main optimization problem involve the solution of another (lower level) optimization problem:

$$\min_{y} V_U(y,z) \tag{1a}$$

$$\text{subject to}\; G_{UI} \leq 0 \tag{1b}$$

$$G_{UE}(y,z) = 0 \tag{1c}$$

$$z = \arg\min_{z} V_L(y,z) \tag{1d}$$

$$\text{subject to}\; G_{LI}(y,z) \leq 0 \tag{1e}$$

$$G_{LE}(y,z) = 0 \tag{1f}$$

Bi-level programming has been addressed since the 1970's, and the survey Colson, Marcotte, and Savard (2005) lists several contributions in the control area going back to the early 1980's, but due to the inherent difficulty of these problem formulations, they have been used rather sporadically since. However, with increasing availability of computing power, interest in these problems is returning (e.g., Kookos & Perkins, 2003; Hovd & Kookos, 2005; Scott et al., 2013). Jones and Morari, (2009) appears to be the first to apply bi-level programming to MPC design, followed by Manum, Jones, Lofberg, Morari, and Skogestad (2009), Hovd (2011), Löfberg (2012).

Assume that the lower level optimization problem admits a unique optimal solution everywhere in its feasible region. Then, it can be replaced by its Karush–Kuhn–Tucker conditions (KKT), resulting in

$$\min_{y,z,\lambda,\nu} V_U(y,z) \tag{2a}$$

$$\text{subject to}\; G_{UI} \leq 0 \tag{2b}$$

$$G_{UE}(y,z) = 0 \tag{2c}$$

$$\lambda \geq 0 \tag{2d}$$

$$G_{LI}(y,z) \leq 0 \tag{2e}$$

$$G_{LE}(y,z) = 0 \tag{2f}$$

$$\lambda \times G_{LI}(y,z) = 0 \tag{2g}$$

$$\nabla_z \mathcal{L}(y,z,\lambda,\nu) = 0 \tag{2h}$$

where the "$\times$" symbol indicates that the kth element of the vector λ of Lagrangian multipliers multiplies the kth constraint in the original lower-level constraints. $\mathcal{L}(y,z,\lambda,\nu) = V_L(y,z) + \lambda^T G_{LI}(y,z) + \nu^T G_{LE}(y,z)$ is the Lagrangian function of the lower-level problem. Notice that there are no nonnegativity constraints for the Lagrangian multipliers ν for the equality constraints.

The difficulty resides in the lower-level problem where the complementarity condition (2g) induces nonlinearity. The technique proposed in Fortuny-Amat and McCarl (1981) is used here to reformulate (2) using auxiliary binary variables $s \in \{0,1\}^N$ (where N denotes the number of inequalities in (2e)):

$$\min_{y,z,\lambda,\nu} V_U(y,z) \tag{3a}$$

$$\text{subject to}\; G_{UI} \leq 0 \tag{3b}$$

$$G_{UE}(y,z) = 0 \tag{3c}$$

$$\lambda \geq 0 \tag{3d}$$

$$\lambda \leq M^{\lambda} s \tag{3e}$$

$$G_{LI}(y,z) \leq 0 \tag{3f}$$

$$G_{LE}(y,z) = 0 \tag{3g}$$

$$G_{LI}(y,z) \geq -M^{u}(1-s) \tag{3h}$$

$$\nabla_z \mathcal{L}(y,z,\lambda,\nu) = 0 \tag{3i}$$

where M^{λ} and M^{u} are diagonal matrices of appropriate dimensions and with sufficiently large values for the diagonal elements. The nonlinear condition (2g) is replaced by conditions (3e) and (3h) in the sense that they provide the same result as the original complementarity condition through suitable combinations of binary variables $s \in \{0,1\}^N$: whenever $s_i = 1$, the set of inequalities (3f)–(3h) degenerates to an equality (i.e., $(G_{LI}(y,z))_i = 0$) and, correspondingly, when $s_i = 0$, the inequality (3h) becomes redundant.

In the following sections this approach is applied to a particular case with an MPC problem as lower level optimization and the Lagrangian multipliers maximization as the upper level optimization. It will be shown how these results can be used for the design of an exact penalty function and what improvements in the bilevel optimization problem can be obtained by considering the special structure of the constraints.

3. Problem preliminaries

Consider a fairly typical (i.e., LTI dynamics, quadratic cost function and linear constraints) MPC formulation:

$$\min_{u_0,\ldots,u_{N-1}} \sum_{k=0}^{N-1} \left(u_k^T R u_k + x_k^T Q x_k\right) + x_N^T Q_f x_N \tag{4a}$$

$$\text{s.t.}\quad G_k x_k + H_k u_k \leq b_k,\; k = 0,\ldots,N \tag{4b}$$

$$x_{k+1} = Ax_k + Bu_k \tag{4c}$$

where $Q \succeq 0$, $Q_f \succeq 0$, $R \succ 0$ are the cost matrices, x_0 is the current state and N is the length of the prediction horizon.

For compactness of notation, (4) is recast as

$$\min_{u} \frac{1}{2} u^T H u + x_0^T F u \tag{5a}$$

$$\text{s.t.}\quad Gu \leq W + Ex_0, \tag{5b}$$

Please cite this article in press as: Hovd, M., & Stoican, F. On the design of exact penalty functions for MPC using mixed integer programming. *Computers and Chemical Engineering* (2013), http://dx.doi.org/10.1016/j.compchemeng.2013.07.001

where $u = [u_0^T \dots u_{N-1}^T]^T$ and matrices H, F, W and E are constructed accordingly. Assume for further use that the optimization problem resides in the $\mathbb{R}^{N \cdot m}$ space (where m denotes the dimension of the input) and that the set of constraints has q inequalities.

Note that the optimization problem (5) is convex and regular, admitting a unique optimal solution everywhere in the feasible domain. Thus, it can be replaced by its Karush–Kuhn–Tucker (KKT) conditions, resulting in:

$$Hu + F^T x_0 + G^T \lambda = 0 \tag{6a}$$

$$\lambda \geq 0 \tag{6b}$$

$$Gu - W - Ex_0 \leq 0 \tag{6c}$$

$$\lambda \times (Gu - W - Ex_0) = 0 \tag{6d}$$

where the "×" symbol indicates that the *kth* element of the vector λ of Lagrangian multipliers multiplies the *kth* constraint from (5b).

The MPC formulation shown above is a so called *hard constrained* problem. There may be initial states x_0 for which there exists no input sequence u for which the constraints are fulfilled. In such a situation the optimization solver will find no solution and consequently will not provide any input for the plant. This is in general considered unacceptable in industrial practice. Practical MPC implementations therefore include some way of relaxing the constraints to ensure that the optimization problem is always feasible and that the input to the plant is always well defined. There are several ways of doing this (Scokaert & Rawlings, 1999), one of the simplest and most common is to use *soft constraints*, as detailed in the next section.

4. Soft constraints in MPC

When using soft constraints, the MPC formulation adds variables in the constraint equations which allow relaxing (some of) the constraints, while the optimization cost function includes terms which penalize the constraint violation. Thus, with a soft constraint formulation, (4) is replaced with:

$$\min_{u,\epsilon} \frac{1}{2} u^T Hu + x_0^T Fu + \phi(\epsilon) \tag{7a}$$

$$\text{s.t.} \quad Gu \leq W + Ex_0 + \theta(\epsilon) \tag{7b}$$

$$\epsilon \geq 0 \tag{7c}$$

where $\phi(\epsilon) : \mathbb{R}^{d_\epsilon} \to \mathbb{R}$ and $\theta(\epsilon) : \mathbb{R}^{d_\epsilon} \to \mathbb{R}^N$ represent the penalty function and the constraint function, respectively.

Remark 1. Naturally, constraints can be softened only if this is physically meaningful and safe to do so. Input constraints are typically hard constraints given by the physics of the process, and it would be absurd to soften such constraints. However, many state/output constraints represent operational desirables (product quality specifications, comfort of operators, etc.), and violating such constraints for some period may be acceptable.

4.1. Equivalence between "hard" and "soft" constraints

The conditions of exact correspondence between "hard" and "soft" constraints follow classic optimization results (Fletcher, 1987) and are detailed in the next proposition.

Proposition 1. *Functions $\phi(\epsilon)$ and $\theta(\epsilon)$ assure an exact correspondence between (5) and (7) whenever $\epsilon = 0$ if the next conditions are fulfilled:*

(i) $\left.\frac{\partial \phi(\epsilon)}{\partial u}\right|_{\epsilon=0} = 0$,
(ii) $\theta(0) = 0$,
(iii) $\left.\frac{\partial \phi(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} \geq \left.\frac{\partial \theta(\epsilon)}{\partial \epsilon}\right|_{\epsilon=0} \lambda$.

for any λ verifying (6).

Proof. Write[1] the KKT conditions for (7)

$$Hu + F^T x_0 + G^T \lambda + \frac{\partial \phi(\epsilon)}{\partial u} = 0 \tag{8a}$$

$$\frac{\partial \phi(\epsilon)}{\partial \epsilon} - \frac{\partial \theta(\epsilon)}{\partial \epsilon} \lambda - \mu = 0 \tag{8b}$$

$$\lambda \geq 0 \tag{8c}$$

$$\mu \geq 0 \tag{8d}$$

$$Gu - W - Ex_0 - \theta(\epsilon) \leq 0 \tag{8e}$$

$$\epsilon \geq 0 \tag{8f}$$

$$\lambda \times \left(Gu - W - Ex_0 - \theta(\epsilon)\right) = 0 \tag{8g}$$

$$\mu \times \epsilon = 0 \tag{8h}$$

and observe that conditions (8a), (8c), (8e) and (8g) are equal with (6a)–(6d) at $\epsilon = 0$ if and only if items (i) and (ii) are verified. In (8b) an additional condition on λ with respect to (6) appears. In order to allow the same values as in (6) for the Lagrange multipliers of (7b) it suffices to verify[2] item (iii), thus concluding the proof. □

Obviously, one would like to keep the extended problem (7) in a tractable formulation. Therefore, one can choose

$$\phi(\epsilon) = F_\epsilon^T \epsilon + \epsilon^T H_\epsilon \epsilon \tag{9a}$$

$$\theta(\epsilon) = G_\epsilon \epsilon \tag{9b}$$

in order to keep the cost function quadratic and the set of constraints linear. This choice of functions leads to the following corollary.

Corollary 1. *Functions $\phi(\epsilon)$ and $\theta(\epsilon)$ written as in (9) assure an exact correspondence between (5) and (7) whenever $\epsilon = 0$ if the condition*

$$F_\epsilon \geq G_\epsilon^T \lambda \tag{10}$$

is fulfilled for any λ verifying (6).

Proof. The proof is immediate and follows from Proposition 1. By replacing (9) into Proposition 1 we observe that items (i) and (ii) are always verified and that item (iii) becomes (10) and remains thus the only condition which needs to be checked. □

Remark 2. Needless to say, for a fixed G_ϵ^T there exists a matrix F_ϵ verifying (10) only if the feasible region in the space of x_0 is bounded (which implies that the Lagrange multipliers are also bounded).

Remark 3. Note that in condition (10) only the linear term F_ϵ in the penalty function (9) determines whether the soft constraints are exact. The quadratic term H_ϵ should ensure that the modified problem is a standard QP, but otherwise is held to be of less importance. Typically, the elements of H_ϵ are therefore small, although some careful choices of the matrix components may influence the

[1] As for (5), we assume that (7) admits a KKT formulation.
[2] Note that we took into account that all the Lagrange multipliers attached to (7c) are greater than or equal with zero since their constraints are active ($\epsilon_i = 0 \to \mu_i \geq 0$).

trade-off between constraint violations in different variables. This issue will not be pursued any further here. Ensuring that the soft constraints are exact is considered to be of primary importance, and the focus in this paper is therefore on the linear term.

In the following the general definition (9) is kept, both for its flexibility (a compromise between the extremes illustrated by the ℓ_1 and ℓ_∞ norms – as shown in the next subsection) and for treating with only one optimization problem (instead of having separate cases for each of the usual norms).

4.2. Polyhedral norm formulation

In literature (e.g., Kerrigan and Maciejowski (2000)), the ℓ_1 and ℓ_∞ norms are frequently used as penalty functions. The condition (10) can be written[3] easily for both of these by using the concept of a polyhedral norm.

Definition 1. [Blanchini (1995)] Having a polyhedral set given in half-space representation $\{x : (Gx)_i \leq 1,\ i = 1 \ldots n\}$, its associated polyhedral norm is defined as $\Psi(G, x) = \max\limits_{i=1\ldots n}\{(Gx)_i\}$ where n denotes the number of rows in matrix G.

Additionally, note that F_ϵ is elementwise positive (since there exists a solution of (6) for which no constraints are saturated we have that $\lambda = 0$ has to verify (10)) which, without any loss of generality, allows to write

$$F_\epsilon = k\mathbf{1}_{d_\epsilon}. \tag{11}$$

This, together with Definition 1 permits to reformulate Corollary 1 as follows.

Corollary 2. *Functions $\phi(\epsilon)$ and $\theta(\epsilon)$ written as in (9) assure an exact correspondence between (5) and (7) whenever $\epsilon = 0$ if the weight k appearing in (11) respects*

$$k \geq \max_{\lambda} \Psi(G_\epsilon^T, \lambda) \tag{12}$$

for any λ verifying (6).

Proof. Introducing (11) into (10) results in $G_\epsilon^T \lambda \leq k\mathbf{1}_{d_\epsilon}$ which, using Definition 1, gives condition (12). □

Remark 4. Condition (12) encompasses the classical ℓ_1 and ℓ_∞ conditions for exact hard constraints:

- by taking $\theta(\epsilon) = \epsilon I_N$ one has that $\phi(\epsilon) = k||\epsilon||_1$ where $k \geq \max\limits_{\lambda} ||\lambda||_\infty$;
- by taking $\theta(\epsilon) = \epsilon\mathbf{1}_N$ one has that $\phi(\epsilon) = k||\epsilon||_\infty$ where $k \geq \max\limits_{\lambda} ||\lambda||_1$.

Remark 5. As it can be seen from Remark 4 and as stated in Hovd (2011), the weight k on the linear term of the penalty function has to be larger than the maximal value of the dual norm of the Lagrangian multipliers of the corresponding hard-constrained optimization problem. This holds for the more general case of the polyhedral norm, since the norm $\Psi(G_\epsilon^T, x)$ is dual to the norm associated with the constraint function ($\theta(\epsilon) = \Psi(G_\epsilon, \epsilon)$ – Ziegler (1995)).

The ℓ_1 norm penalty function increases the number of decision variables in the optimization problem by the number of constraints that are relaxed. In contrast, the ℓ_∞ norm penalty norm only increases the number of decision variables in the optimization problem by 1 – since the same slack variable can be used for all relaxed constraints. For this reason, ℓ_∞ norm penalty functions are often preferred, although it is shown in Rao, Wright, and Rawlings (1998) that the addition of the ℓ_1 norm optimization variables can be handled at virtually no additional computational cost if problem structure is utilized in the QP solver.

On the other hand, the ℓ_∞ norm can result in unexpected behavior and poor performance if it is used to soften an output constraint for which there is an inverse response. In Hovd and Braatz (2001a) it was shown how to minimize this problem by using time-dependent weights in the optimization criterion.

A sufficiently high value of the linear term in the penalty function will ensure that the soft constraints are exact (see Corollary 2). However, a too large term is generally not desirable, since it may lead to unnecessarily violent control should the plant for some reason be outside of the (hard constrained) feasible region. Therefore, one would wish to find the minimal values for which (12) is still satisfied with λ verifying (6). This is a non-convex optimization problem which in general has been considered intractable. In the next section we will reformulate conditions (6) in order to recast the optimization problem into a tractable MI(L)P problem.

5. Reformulation of the KKT conditions

From Corollary 2, the aim is to solve the problem

$$\max_{x_0} \Psi(G_\epsilon^T, \lambda) \tag{13a}$$

$$\min_{u} \frac{1}{2} u^T H u + x_0^T F u \tag{13b}$$

$$\text{s.t.} \quad Gu \leq W + Ex_0. \tag{13c}$$

Problem (13) is a bi-level programming problem where the constraints of the main optimization problem involve the solution of another (lower level) optimization problem. However, it differs from standard bi-level optimization problems in that the upper level objective function is not determined before the lower-level problem is replaced by its KKT conditions (i.e., finding the maximum of the polyhedral norm $\Psi(G_\epsilon^T, \lambda)$ in (13a) requires finding first the multipliers which satisfy (13b and c).

5.1. MILP formulation

Proceed by replacing the lower-level problem in (13) by its KKT condition, and from (6d) realize that the complementarity condition induces nonlinearity. Recall here (3) and replace the complementarity conditions with the equivalent linear formulation using the auxiliary binary variables $s \in \{0, 1\}^q$:

$$\max_{\lambda, u, x_0, s} \Psi(G_\epsilon^T, \lambda) \tag{14a}$$

$$Hu + F^T x_0 + G^T \lambda = 0 \tag{14b}$$

$$\lambda \geq 0 \tag{14c}$$

$$\lambda \leq M^\lambda s \tag{14d}$$

$$Gu - W - Ex_0 \leq 0 \tag{14e}$$

$$Gu - W - Ex_0 \geq -M^u(1 - s) \tag{14f}$$

where M^λ and M^u are diagonal matrices of appropriate dimensions and with sufficiently large values for the diagonal elements.

[3] One can make use of the fact that the Lagrange multipliers are positive and ignore the $|\cdot|$ operator which appears in the norm definition:

$$||a||_p = \left(\sum_i |a_i|^p\right)^{\frac{1}{p}}$$

.

Note that the constraints (14e) are the constraints of the original MPC problem. Their presence means that one will not have to calculate the feasible region explicitly. This is a major advantage, since the projection operation involved in calculating explicitly the feasible region can be computationally very demanding for large systems.

However, the direct inclusion of λ as free variables in (14) allows for unnecessarily large solutions – bounded only by M^{λ} from (14d). In order to deal with this issue, in Hovd (2011), an additional minimization level was introduced in (14). This approach has the drawback that it increases the number of auxiliary variables (due to the additional layer of minimization) and thus, adds to the computational difficulty:

$$Gu - W - Ex_0 \leq 0 \tag{15a}$$

$$Gu - W - Ex_0 \geq -M^u(1-s) \tag{15b}$$

$$\min_{\lambda} \Psi(G_{\epsilon}^T, \lambda) \tag{15c}$$

$$s.t. \quad \lambda \geq 0 \tag{15d}$$

$$\lambda \leq M^{\lambda} s \tag{15e}$$

$$Hu + F^T x_0 + G^T \lambda = 0. \tag{15f}$$

Note that the norm appearing in (15c) should be the same as the norm which is maximized at the upper level (14a), to ensure the accuracy of the solution.

To further reduce the computational effort, it is advantageous to explicitly impose that the optimization problem takes into account only linearly independent combinations of constraints (as it will be detailed in the next subsection). This makes the addition of another level of optimization, as in Hovd (2011) superfluous and permits to significantly reduce the computation time.

5.2. Explicit imposition of the LICQ condition

Recall the following notion characterizing an active set of constraints.

Definition 2. [Tøndel, Johansen, and Bemporad (2003)] For an active set, it is said that the linear independence constraint qualification (LICQ) holds if the set of active constraint gradients are linearly independent.

As long as the set of active constraint gradients (i.e., the subset of rows from G saturated by the optimum u corresponding to a given x_0) respects LICQ, the KKT problem (6) is well-posed in the sense that both the primal solution u and the dual solution λ are uniquely defined.

In Bemporad et al. (2002), the construction of the explicit PWA control (and thus the determination of the Lagrangian multipliers) is considered only for selections which respect LICQ since only the constraints respecting this qualification determine a full-dimensional region in the state-space. On the other hand, (14) considers all feasible combinations of constraints regardless of their linear independence. Therefore, the Lagrangian multipliers will not be always uniquely determined which in turn leads to unnecessarily large solutions.

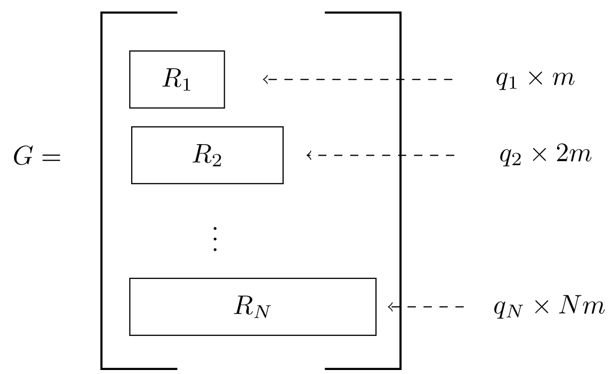
A first solution was proposed in (15) but at the cost of increased complexity. Instead, the aim here is to exploit the structure of the constraint matrix G and use the information in order to impose that only subsets of constraints which respect LICQ are chosen. For future use, define this collection of subsets as $\mathcal{I}_N^{\circ}$. The cardinality of this collection and its subsequent refinements give a theoretical upper bound for the number of subproblems that the MI optimization problem may need to solve in a worst-case scenario.

Since one is interested only in candidate sets which respect LICQ it is clear that one can select at most $N \cdot m$ constraints from the existing q without violating the linear independence requirement in the $\mathbb{R}^{N \cdot m}$ space. This leads to an upper bound of

$$\#\mathcal{I}_N^{\circ} = \sum_{j=0}^{N \cdot m} \binom{q}{j} \tag{16}$$

candidate subsets. The bound is obtained by selecting from the available q constraints $0, 1, \ldots, N \cdot m$ active constraints successively.

However, the resulting bound is conservative, and a better bound can be computed by making use of the special structure characterizing matrix G: as seen in (4b), up to the time instant k only the initial state x_0 and the sequence of inputs $u_0, \ldots, u_k$ appear in the constraints. Algebraically, this means that matrix G has a lower block-triangular structure:



(17)

where each block R_k describing the kth order constraints has q_k rows and km columns (note that $q_1 + q_2 + \ldots q_N = q$).

In the following result conditions are given which restrict the number of row selections from the constraint matrix:

Proposition 2. *Let i_k denote the number of constraints selected from each block R_k. Then the conditions*

$$0 \leq i_k \leq \min(q_k, km), \ \sum_{j=0}^{k} i_j \leq km, \quad \forall k = 1, \ldots, N \tag{18}$$

define all the selections of constraints which can be LICQ and the collection $\mathcal{I}_N^{\circ}$ of candidate active sets is bounded by

$$\#\mathcal{I}_N^{\circ} = \sum_{(i_1, \ldots, i_N)\,\text{verifies}\,(18)} \left(\prod_{j=1}^{N} \binom{q_j}{i_j} \right). \tag{19}$$

Proof. The reasoning for the conditions in (18) is based upon the fact that in a collection of 'y' rows where only the first 'x' elements in each row are non-zero, at most '$\min(x, y)$' rows can be selected and still be linearly independent. This property leads directly to the first inequality in (18) whereas for the second one can observe that the following selection rule applies: from the first q_1 rows one can select at most m, from the first $q_1 + q_2$ rows we can select at most $2 \cdot m$ and so forth until, from the first $q_1 + \ldots + q_N = q$ rows we can select at most $N \cdot m$ rows.

The bound (19) can be computed by simply counting the candidate sets of active constraints. For a selection $(i_1, \ldots, i_N)$ take each block R_j and extract all the combinations of i_j rows from it, which means $\binom{q_j}{i_j}$ possibilities. Each of these combinations is then considered with each of the other combinations from the other blocks. □

The construction of $\mathcal{I}_N^\circ$ was made assuming that any i_j subset of rows from the block R_j is linearly independent, assumption which may not hold in practical cases (e.g., when the bounds upon states/inputs are symmetrical). In this sense, consider the next definition:

Definition 3. We refer to an mpQP problem of the form (4) as input-symmetric if the involved constraints can be rewritten as

$$\begin{bmatrix} G_1 \\ G_2 \end{bmatrix} u - \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} - \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} x_0 \le 0 \tag{20}$$

where $G_2 = -G_1$. Consequently, each block-matrix R_j with q_j rows given as in (17) can be written as $R_j = \begin{bmatrix} \overline{R}_j \\ -\overline{R}_j \end{bmatrix}$ where $\overline{R}_j$ has $\frac{q_j}{2}$ rows.

Under these circumstances, the following corollary applies.

Corollary 3. *Consider that (4) is input-symmetrical and let i_k denote the number of constraints selected from each block $\overline{R}_k$. Then, the LICQ constraint selections have to respect relations*

$$0 \le i_k \le \min\left(\frac{q_k}{2}, km\right), \sum_{j=0}^{k} i_j \le km, \ \forall k = 1, \ldots, N \tag{21}$$

and the collection $\mathcal{I}_N^\circ$ of candidate active sets is bounded by

$$\#\mathcal{I}_N^\circ = \sum_{(i_1,\ldots,i_N)\,\text{verifies}\,(21)} \left(\prod_{j=1}^{N} \binom{\frac{q_j}{2}}{i_j} 2^{i_j} \right). \tag{22}$$

Proof. The proof is constructive. Relations (21) describe the same conditions as in (18) with the difference that at most $\frac{q_j}{2}$ rows can be selected from a block R_j.

For the bound (22) the reasoning is as follows: select i_j rows from the $\frac{q_j}{2}$ rows which define the sub-block $\overline{R}_j$; subsequently, consider all the possible sign permutations for the selected constraints (i.e., 2^{i_j}); the rest of the procedure follows the proof of Proposition 2. □

Remark 6. The conditions (18) and (21) describe an integral polytope (i.e., a polytope whose extreme vertices are integers) and the selections $(i_1, \ldots, i_N)$ are points contained in this polytope. These notions can be described formally using the Ehrhart polynomial and the associated theory (Diaz & Robins, 1997). Using these elements it is possible to determine the number of integer points which respect (18) or (21) and to enumerate them. The actual computation of the Ehrhart polynomial coefficients and the enumeration of the selections $(i_1, \ldots, i_N)$ can be done with the Ehrhart routines of the PolyLib library (see, Clauss (1996), Clauss, Loechner, and Wilde (1997) for details).

Finally, using Proposition 2 one can reformulate (14) more efficiently by adding constraints which force the selection of only LICQ sets of constraints:

$$Hu + F^T x_0 + G^T \lambda = 0 \tag{23a}$$

$$\lambda \ge 0 \tag{23b}$$

$$\lambda \le M^\lambda s \tag{23c}$$

$$Gu - W - Ex_0 \le 0 \tag{23d}$$

$$Gu - W - Ex_0 \ge -M^u(1-s) \tag{23e}$$

$$0 \le \sum_{i=\tau_{k-1}+1}^{\tau_k} s_i \le \min(q_k, km), \sum_{i=1}^{\tau_k} s_i \le km \tag{23f}$$

where $\tau_k = q_1 + \ldots + q_k$ for any $k = 1, \ldots, N$ (with the convention that $\tau_0 = 0$).

Thus, the computation time is significantly reduced for (23) with respect to the original formulation (14).

Remark 7. Evidently, the construction (23) can be adapted for the case of input-symmetric constraints. By using the first part of Corollary 3 and the notation of Definition 3 we observe that to the constraints of (23) an additional condition can be added:

$$s_1 + s_2 \le 1 \tag{24}$$

where s_1 and s_2 denote respectively the binary variables associated with (G_1, E_1, W_1) and (G_2, E_2, W_2). The condition simply states that any two binary variables which correspond to the same row from G_1, respectively G_2 cannot be "1" simultaneously.

Remark 8. Note that even if structurally a set of constraints can be linearly independent (i.e., it respects the selection constraints given in Proposition 2 and Corollary 3), it may happen that the actual constraints selected do not respect LICQ. In this case, the formulations (23) and (24) will fail. Whenever the constraints are input-symmetric and conditions (23) are applied, the problem still gives a correct solution since even if non-LICQ sets of constraints are possible, it is not possible that both can be saturated for the same input and initial state.

Finally, one can apply formulation (23) into the maximization problem (12):

$$k^* = \max_{\lambda,u,x_0,s} \max_{j=1\ldots d_\epsilon} (G_\epsilon^T \lambda)_j \tag{25a}$$

$$\text{s.t.} \quad Hu + F^T x_0 + G^T \lambda = 0 \tag{25b}$$

$$\lambda \ge 0 \tag{25c}$$

$$\lambda \le M^\lambda s \tag{25d}$$

$$Gu - W - Ex_0 \le 0 \tag{25e}$$

$$Gu - W - Ex_0 \ge -M^u(1-s) \tag{25f}$$

$$0 \le \sum_{i=\tau_{k-1}+1}^{\tau_k} s_i \le \min(q_k, km), \sum_{i=1}^{\tau_k} s_i \le km \tag{25g}$$

with d_ϵ denoting the number of rows in G_ϵ^T. The second "max" operator comes from the way the polyhedral norm $\Psi(\cdot, \cdot)$ is defined.

In the general case when G_ϵ is not a vector (e.g., for the ℓ_1 norm we have $G_\epsilon = \mathbf{1}_N$, as seen in Remark 4) one will have to recast (25) as follows:

$$k^* = \max_{j=1\ldots d_\epsilon} \max_{\lambda,u,x_0,s} (G_\epsilon^T \lambda)_j \tag{26a}$$

$$\text{s.t.} \quad Hu + F^T x_0 + G^T \lambda = 0 \tag{26b}$$

$$\lambda \ge 0 \tag{26c}$$

$$\lambda \le M^\lambda s \tag{26d}$$

$$Gu - W - Ex_0 \le 0 \tag{26e}$$

$$Gu - W - Ex_0 \ge -M^u(1-s) \tag{26f}$$

$$0 \le \sum_{i=\tau_{k-1}+1}^{\tau_k} s_i \le \min(q_k, km), \sum_{i=1}^{\tau_k} s_i \le km \tag{26g}$$

Please cite this article in press as: Hovd, M., & Stoican, F. On the design of exact penalty functions for MPC using mixed integer programming. *Computers and Chemical Engineering* (2013), http://dx.doi.org/10.1016/j.compchemeng.2013.07.001

where the order of the "max" operators is switched and one basically has to compute d_ϵ lower level maximizations in order to solve the overall problem.

Lastly, recall the relations (10) and (11). Note that the maximization in (12) is a linear program and can thus be replaced by its dual. It follows then that an alternative formulation of Corollary 2 exists:

$$k^* \leq \min_{G_\epsilon^T \lambda \leq k\mathbf{1}} \lambda \tag{27}$$

for any λ verifying (6).

Applying the same line of reasoning as for (25) one obtains the following "max-min" problem:

$$k^* = \max_{u,x_0,s} k^\circ \tag{28a}$$

$$\text{s.t.} \quad Hu + F^T x_0 + G^T \lambda = 0 \tag{28b}$$

$$\lambda \geq 0 \tag{28c}$$

$$\lambda \leq M^\lambda s \tag{28d}$$

$$Gu - W - Ex_0 \leq 0 \tag{28e}$$

$$Gu - W - Ex_0 \geq -M^u(1-s) \tag{28f}$$

$$0 \leq \sum_{i=\tau_{k-1}+1}^{\tau_k} s_i \leq \min(q_k, km), \sum_{i=1}^{\tau_k} s_i \leq km \tag{28g}$$

$$k^\circ = \min_k k \tag{28h}$$

$$\text{s.t.}\, G_\epsilon^T \lambda \leq k\mathbf{1}. \tag{28i}$$

Note that in the lower-order problem (28h)–(28i) the only variable is 'k' (a slack variable) whereas 'λ' is only a parameter, since the upper-order problem (28a)–(28g) uniquely defines for a given triplet (u, x_0, s) the associate Lagrangian multipliers (due to the presence of the LICQ conditions). Consequently, one can reformulate (28) where the optimization (28h)–(28i) is replaced by its KKT conditions (where the complementarity condition is again linearized through the addition of auxiliary binary variables $w \in \{0, 1\}^{d_\epsilon}$):

$$k^* = \max_{u,x_0,s,\delta,w,k^\circ} k^\circ \tag{29a}$$

$$\text{s.t.} \quad Hu + F^T x_0 + G^T \lambda = 0 \tag{29b}$$

$$\lambda \geq 0 \tag{29c}$$

$$\lambda \leq M^\lambda s \tag{29d}$$

$$Gu - W - Ex_0 \leq 0 \tag{29e}$$

$$Gu - W - Ex_0 \geq -M^u(1-s) \tag{29f}$$

$$0 \leq \sum_{i=\tau_{k-1}+1}^{\tau_k} s_i \leq \min(q_k, km), \sum_{i=1}^{\tau_k} s_i \leq km \tag{29g}$$

$$1 - \mathbf{1}^T \delta = 0 \tag{29h}$$

$$\delta \geq 0 \tag{29i}$$

$$\delta \leq M^\delta w \tag{29j}$$

$$G_\epsilon^T \lambda - \mathbf{1}k \leq 0 \tag{29k}$$

$$G_\epsilon^T \lambda - \mathbf{1}k \geq -M^\delta(\mathbf{1} - w). \tag{29l}$$

With this last formulation there are two alternatives, (26) and (29), for solving the original optimization problem (13). It remains at the latitude of the reader to choose between these representations, depending on the specifics of the problem at hand. For example, in the case of the ℓ_1 norm, the formulation (26) has a slightly more compact form whereas for the ℓ_∞ norm, the formulation (29) proves superior.

5.3. Numerical considerations

Some conditions need to be imposed on the numerical values of parameters M^λ and M^u such as not to alter the original result (here (23) is used as basis for analysis without any loss of generality since the other variants can be treated similarly). These conditions can be summarized as follows:

(i) M^u sufficiently large such that the discarded inequalities from (23e) do not influence the ones in (23d);
(ii) M^λ sufficiently large such that there exists a λ permitting an optimum solution in (23) identical to one the which can be obtained in (6);
(iii) M^u and M^λ small enough such that numerical problems are minimized and/or avoided.

Item (iii) implies the need to take the minimal values of parameters M^u and M^λ which respect the requirements given in items (ii) and (iii).

Item (i) has a simple geometrical meaning: whenever an inequality from (23e) is discarded (the associated binary variable is $s_i = 0$), it should be redundant with respect to the feasible region defined by the constraints in (23d). Without entering into details, this can be tested using a variant of the Farkas lemma (as explained in Vassilaki, Hennet, and Bitsoris (1988)) which leads to a LP problem whose solution gives the minimal M^u.

For item (ii) the reasoning applied to the first item cannot be applied. The solution is to solve (23) iteratively: as long as the λ found as result of the optimization problem saturates any of the constraints (23c) the corresponding element of M^λ is increased and the computation of (23) repeated.

It is worth noting that by adding constraint (23f), the complexity of the problem is reduced. A MI problem might, in the worst case scenario, pass through all the admissible combinations of binary variables. For conditions (14) this means that at most 2^N sub-problems may need to be solved. For comparison, in the case of (23), the maximal number of sub-problems is less than $\sum_{i=0}^{d} \binom{N}{i}$.

6. Illustrative examples

Three examples are considered, from Hovd, Scibilia, Maciejowski, and Olaru (2009), Hovd and Braatz (2001a) and Hovd and Braatz (2001b) respectively.

Consider first a double integrator, described by

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0.3 \end{bmatrix}$$

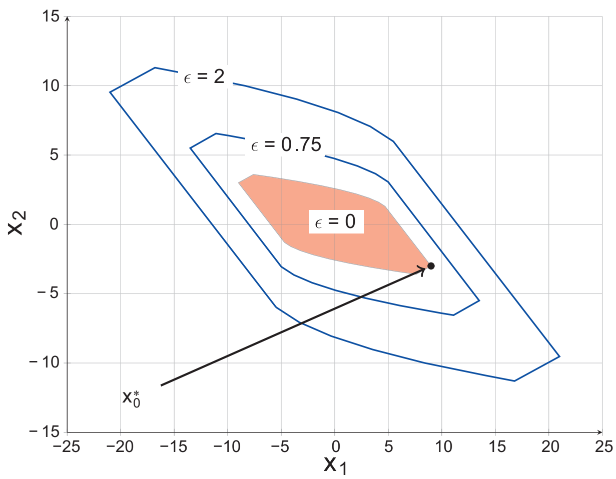


Fig. 1. Feasible region of problem (5) and the point where the ℓ_1-norm of the Lagrangian multipliers is maximized. (For interpretation of references to colour in the text, the reader is referred to the web version of this article.)

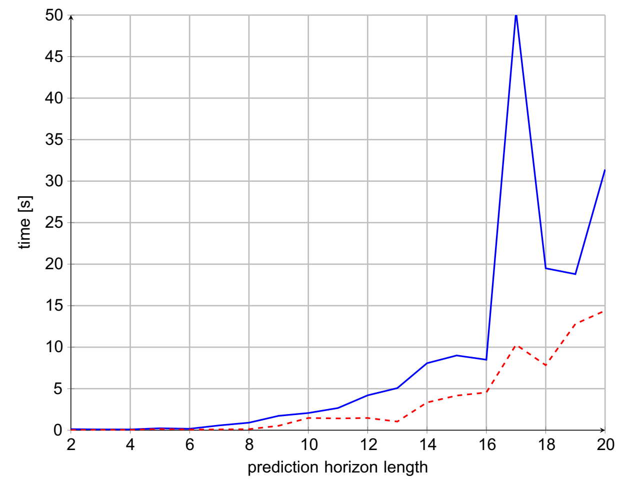


Fig. 2. Time necessary to solve problem (15) – solid, vs. for solving problem (23) – dashed. (For interpretation of references to colour in this figure legend, the reader is referred to the web version of this article.)

with constraints

$$-1 \le u_k \le 1, \begin{bmatrix} -5 \\ -5 \end{bmatrix} \le x_k \le \begin{bmatrix} 5 \\ 5 \end{bmatrix}.$$

The weight matrices used are $Q = I_2$ and $R = 1$, whereas the prediction horizon $N = 15$ is used, resulting in 58 constraints in the MPC formulation (5) (and the same number of binary variables in the optimization formulation (23)).

MPC with an ℓ_∞-norm penalty function is considered, as shown in the second item of Remark 4. That is, $\theta(\epsilon) = \epsilon \mathbf{1}_N$ and $\psi(\epsilon) = k\epsilon$ with $\epsilon \in \mathbb{R}$. Maximizing the ℓ_1-norm of the Lagrangian multipliers in order to ensure exact soft constraints, it is found that the maximum is achieved at $x_0^* = [9 \ -3]^T$ (as shown in Fig. 1) and the corresponding value of the ℓ_1-norm and thus of the penalty function weight is $k = ||\lambda||_1 = 911.2465$. This value, and the location of the maximum is verified by solving the MPC problem at all vertices of the feasible region.

The feasible domain of $x_0 \in \mathbb{R}^2$ for problem (5) is illustrated in Fig. 1 – the region filled region, and contrasted with projections of the unbounded feasible domain characterizing the soft problem (7) – contours drawn for different bounds upon ϵ. For a meaningful comparison we have projected along $\epsilon = 0.75$ and $\epsilon = 2$ values (e.g., inside the contour defined by $\epsilon = 2$, the penalty cost is less than $911.2465 \cdot 2$).

Fig. 2 depicts the cost difference between solving optimization problems (15) and (23), respectively, for a prediction horizon ranging from $N = 2$ to $N = 20$. As expected, it can be observed that the presence of the constraints (23f) in (23) reduces the computation time significantly.

Lastly, Fig. 3 compares the trajectories generated by the soft/hard constrained MPC. As expected, it is found that inside the feasible region the trajectories coincide and that in the case of an infeasible starting point, one can still compute a trajectory.

The second example is taken from Hovd and Braatz (2001b). The discrete-time model is given by

$$A = \begin{bmatrix} 2 & -1.45 & 0.35 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$C = \begin{bmatrix} -1 & 0 & 2 \end{bmatrix}, D = 0$$

and the constraints are given by

$$-1 \le Cx \le 1, \quad -1 \le u \le 1.$$

The state weight is given by $Q = C^T C$, the input weight is $R = 10$, and a prediction horizon $N = 5$ with a terminal set is used.

Soft-constrained MPC controllers, one using the ℓ_1 formulation and the other using the ℓ_∞ formulation, are simulated. Both controllers start from the same infeasible state ($x_0 = [1.5\ 1.5\ 21.5]^T$). For illustration, Fig. 4 depicts the resulting output and corresponding input values for both norm implementations. As it can be seen, for these particular dynamics, the use of the ℓ_1 norm (solid blue line) in the penalty function results in a large overshoot whereas the ℓ_∞ norm (dashed red line) has a smaller overshoot but takes longer to converge to the origin. For illustration purposes different scales are used in Fig. 4(b) and (c) but it can be seen that the ℓ_∞ input has a significantly smaller magnitude – but displays strong oscillations. A technique to mitigate such oscillations was proposed in Hovd and Braatz (2001b). Finally, the inputs resulting from the use of the ℓ_1 norm behave better, but this is to be traded with the additional difficulty of solving online a larger MPC problem. The issue can be alleviated by the use of special-purpose qp solvers as the one described in Rao et al. (1998) which is able to efficiently utilize the problem structure.

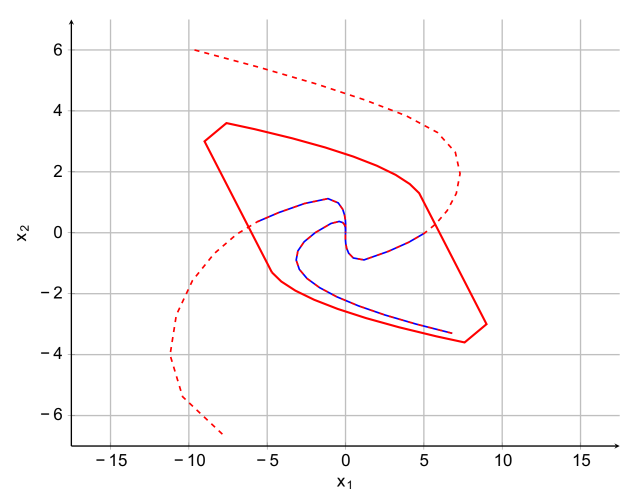
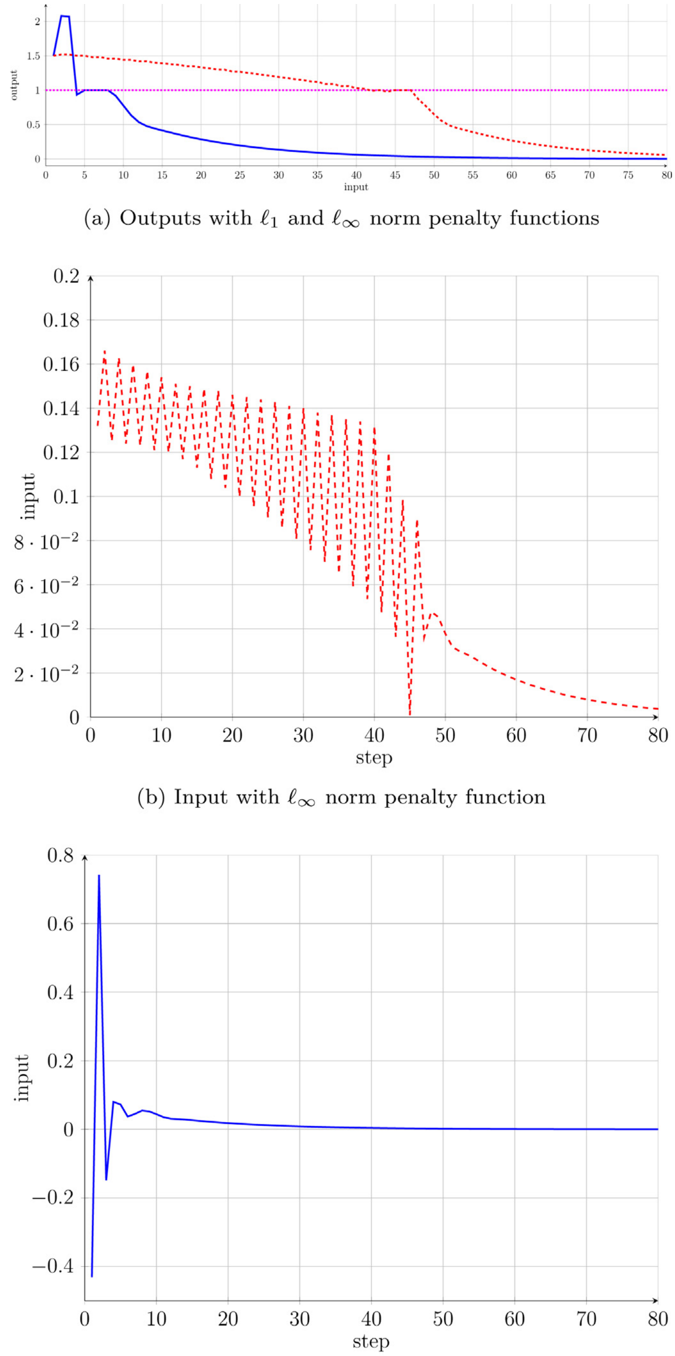


Fig. 3. Depiction of trajectories generated with/out soft constraints (dashed red and solid blue respectively). (For interpretation of references to colour in this figure legend, the reader is referred to the web version of this article.)

Please cite this article in press as: Hovd, M., & Stoican, F. On the design of exact penalty functions for MPC using mixed integer programming. *Computers and Chemical Engineering* (2013), http://dx.doi.org/10.1016/j.compchemeng.2013.07.001



(a) Outputs with ℓ_1 and ℓ_∞ norm penalty functions

(b) Input with ℓ_∞ norm penalty function

(c) Input with ℓ_1 norm penalty function

Fig. 4. Depiction of inputs/outputs generated with soft constraints for a ℓ_1 and ℓ_∞-norm penalty functions when starting from an infeasible point.

The third and last example is taken from Hovd and Braatz (2001a). The discrete-time model is given by

$$A = \begin{bmatrix} .928 & .002 & -.003 & -.004 \\ .041 & .954 & .012 & .006 \\ -.052 & -.046 & .896 & -.003 \\ -.069 & .051 & .032 & .935 \end{bmatrix}, \quad B = \begin{bmatrix} 000 & .336 \\ .183 & .007 \\ .090 & -.009 \\ .042 & .012 \end{bmatrix}$$

$$C = \begin{bmatrix} .000 & .000 & -.098 & .269 \\ .000 & .000 & .080 & .327 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

and the constraints are given by

$$\begin{bmatrix} -1 \\ -1 \end{bmatrix} \le Cx \le \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} -1 \\ -1 \end{bmatrix} \le u \le \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

The state weight is given by $Q = C^TC$, the input weight is $R = I_2$, and a prediction horizon $N = 10$ is used. This problem has 116 constraints in the MPC formulation, and hence requires 116 binary variables in the MILP formulation for calculating $||\lambda||_1$. The maximum value of the norm is found at $x = [25.5724\ 25.3546\ 9.7892\ 0.2448]^T$, and has the value $||\lambda||_1 = 38907$. For this example, calculating the feasible region is subject to numerical errors, and the result has therefore not been verified by checking the vertices of the feasible region. A partial verification has been done by finding the active constraints which correspond to the λ given by the LICQ problem and using them to describe the Lagrangian multipliers over the current critical region. Then, all the vertices of this critical region were investigated, and the results were consistent with the results of the optimization.

7. Conclusions

In this paper, procedures for calculating the maximum values of the polyhedral norm of the Lagrangian multipliers of standard QP problems have been developed. The procedures are intended for designing penalty functions for soft constraints in MPC, to find the required weights for making the constraints exact. The calculation procedures are formulated as MILP problems, which are known in general to be NP-hard and thus very computationally demanding to solve. By forcing the selection of active constraints which are LICQ the problem is simplified and the computation times reduced.

Section 5.3 provides a discussion of some relevant numerical issues in solving the MILP problems resulting from the formulation developed in this paper. Although the examples illustrate that some problems of industrial-size scale can be solved (Example 3 has 116 constraints in the MPC formulation), it is clear that further improvements to MILP solvers and more efficient problem formulations will be of interest.

References

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

Blanchini, F. (1995). Nonquadratic Lyapunov functions for robust control. *Automatica*, *31*(3), 451–461.

Clauss, P. (1996, May 25-28). Counting solutions to linear and nonlinear constraints through Ehrhart polynomials: Applications to analyze and transform scientific programs. In *Proceedings of the 10th international conference on supercomputing* (pp. 278–285). Philadelphia, PA, USA: ACM.

Clauss, P., Loechner, V., & Wilde, D. (1997). Deriving formulae to count solutions to parameterized linear systems using Ehrhart polynomials: Applications to the analysis of nested-loop programs. In *Tech. rep., Informatique et Calcul Paralléle Scientifique (ICPS)*. Strasbourg, France: Universite Louis Pasteur.

Colson, B., Marcotte, P., & Savard, G. (2005). Bilevel programming: A survey. *4OR: A Quarterly Journal of Operations Research*, *3*(2), 87–107.

de Oliveira, N., & Biegler, L. (1994). Constraint handing and stability properties of model-predictive control. *AIChE Journal*, *40*(7), 1138–1155.

Diaz, R., & Robins, S. (1997). The Ehrhart polynomial of a lattice polytope. *The Annals of Mathematics Journal*, *145*(3), 503–518.

Fletcher, R. (1987). *Practical methods of optimization*. Chichester UK: John Wiley & Sons Ltd.

Fortuny-Amat, J., & McCarl, B. (1981). A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, *32*(9), 783–792.

Hovd, M. (2011, 28 August–2 September). Multi-level programming for designing penalty functions for MPC controllers. In *Proceedings of the 18th IFAC World Congress* Milano, Italy, (pp. 6098–6103).

Hovd, M., & Braatz, R. (2001a, 25–27 June). Handling state and output constraints in MPC using time-dependent weights. In *Proceedings of the 20th American Control Conference, Vol. 3* Arlington, Virginia, USA, (pp. 2418–2423).

Hovd, M., & Braatz, R. D. (2001b, 4–6 June). *Proceedings of the 6th IFAC symposium on dynamics and control of process systems (DYCOPS)* Jejudo Island, Korea, (pp. 251–256).

Hovd, M., & Kookos, I. (2005, 4–8 July). Calculating dynamic disturbance rejection measures. In *Proceedings of the 16th IFAC World Congress* Prague, Czech Republic.

Hovd, M., Scibilia, F., Maciejowski, J., & Olaru, S. (2009, 16-18 December). Verifying stability of approximate explicit MPC. In *Proceedings of joint 48th IEEE conference on decision and control and 28th Chinese control conference* Shanghai, PR China, (pp. 6345–6350).

Jones, C., & Morari, M. (2009, 23–26 August). Approximate explicit MPC using bilevel optimization. In *Proceedings of the 10th IEEE European Control Conference* Budapest, Hungary, (pp. 2396–2401).

Kerrigan, E., & Maciejowski, J. (2000, 4-7 September). Soft constraints and exact penalty functions in model predictive control. In *Proceedings of the UKACC international conference on control* Cambridge, UK.

Kookos, I., & Perkins, J. (2003). On the efficient computation of disturbance rejection measures. *Computers & Chemical Engineering Journal*, *27*(1), 95–99.

Löfberg, J. (2012). Oops! I cannot do it again: Testing for recursive feasibility in MPC. *Automatica*, *48*(3), 550–555.

Manum, H., Jones, C., Lofberg, J., Morari, M., & Skogestad, S. (2009, 16-18 December). Bilevel programming for analysis of low-complexity control of linear systems with constraints. In *Proceedings of Joint 48th IEEE conference on decision and control and 28th Chinese control conference* Shanghai, PR China, (pp. 946–951).

Qin, S., & Badgwell, T. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice Journal*, *11*(7), 733–764.

Rao, C., Wright, S., & Rawlings, J. (1998). Application of interior-point methods to model predictive control. *Journal of Optimization Theory and Applications*, *99*(3), 723–757.

Scokaert, P., & Rawlings, J. (1999). Feasibility issues in linear model predictive control. *AIChE Journal*, *45*(8), 1649–1659.

Scott, J. K., Findeisen, R., Braatz, R. D., & Raimondo, D. M. (2013, June 17–19). Design of active inputs for set-based fault diagnosis. In *Proceedings of the 2013 American control conference* Washington, DC, USA, (pp. 3567–3572).

Tøndel, P., Johansen, T., & Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, *39*(3), 489–497.

Vada, J. (2000). *Prioritized infeasibility handling in linear model predictive control: Optimality and efficiency*. Engineering Cybernetics Department, NTNU. Ph.D. thesis.

Vassilaki, M., Hennet, J. C., & Bitsoris, G. (1988). Feedback control of linear discrete-time systems under state and control constraints. *International Journal of Control*, *47*(6), 1727–1735.

Zafiriou, E., & Marchal, A. (1991). Stability of SISO quadratic dynamic matrix control with hard output constraints. *AIChE Journal*, *37*(10), 1550–1560.

Ziegler, G. M. (1995). *Lectures on polytopes* (Vol. 152) Heidelberg, Germany: Springer.