Preprints, 5th IFAC Conference on Nonlinear Model Predictive Control
September 17-20, 2015. Seville, Spain

# Solving Mixed-Integer Quadratic Programs via Nonnegative Least Squares

**Alberto Bemporad** *

* *IMT Institute for Advanced Studies Lucca, Italy (email: alberto.bemporad@imtlucca.it).*

**Abstract:** This paper proposes a new algorithm for solving Mixed-Integer Quadratic Programming (MIQP) problems. The algorithm is particularly tailored to solving small-scale MIQPs such as those that arise in embedded hybrid Model Predictive Control (MPC) applications. The approach combines branch and bound (B&B) with nonnegative least squares (NNLS), that are used to solve Quadratic Programming (QP) relaxations. The QP algorithm extends a method recently proposed by the author for solving strictly convex QP's, by (*i*) handling equality and bilateral inequality constraints, (*ii*) warm starting, and (*iii*) exploiting easy-to-compute lower bounds on the optimal cost to reduce the number of QP iterations required to solve the relaxed problems. The proposed MIQP algorithm has a speed of execution that is comparable to state-of-the-art commercial MIQP solvers and is relatively simple to code, as it requires only basic arithmetic operations to solve least-square problems.

*Keywords:* Mixed-integer quadratic programming, Quadratic Programming, Active set methods, Nonnegative least squares, Model predictive control, Hybrid systems.

## 1. INTRODUCTION

After the first paper (Bemporad and Morari, 1999), hybrid Model Predictive Control (MPC) has received tremendous attention, both by academic researchers and industrial engineers. The main reason is the large variety of complex control problems that the approach can handle, due to the ability of capturing in the same model multiple linear dynamics, logic variables and states, mixed linear and logical constraints, meeting closed-loop performance and constraint satisfaction requirements in a rather direct, effective, and systematic way (Bemporad and Morari, 1999). As for linear and nonlinear MPC, this success would not have been possible if good numerical solvers were not available to solve Mixed-Integer Quadratic Programming (MIQP) or mixed-integer linear programming problems, and to automatize the translation of the hybrid control problem into a computationally-tractable form (Torrisi and Bemporad, 2004). In fact, evaluating the hybrid MPC control decision on line requires solving an MIQP at each time step. While to date very efficient commercial solvers exist to solve MIQPs (Gurobi Optimization, Inc., 2014; IBM, Inc., 2014; Fair Isaac Corporation, 2015), these are not tailored to embedded applications on low-cost/low-power control boards. Efforts in this direction were recently proposed by Frick et al. (2015), extending an embedded convex programming solver based on interior-point methods to a Branch and Bound (B&B) setting (Floudas, 1995). Another approach for B&B-based MIQP tailored to MPC problems was proposed by Axehill and Hansson (2006), where a dual QP method is employed and warm-started from parent-node solutions, and optimality conditions are solved using Riccati recursions. The method does not exploit however dual lower bounds on the optimal cost, that is instead particularly useful to reduce the number of solved QP relaxations (Fletcher and Leyffer, 1998).

In this paper, we propose a B&B method to solve MIQPs that leverages on a novel solution algorithm for strictly convex QPs recently developed by the author (Bemporad, 2015b). Such a QP solver is an active set method based on a nonnegative least squares (NNLS) reformulation of the quadratic optimization problem, is quite fast and simple to code, and, contrary to iterative methods, converges after a finite number of iterations to the solution, rather insensitively with respect to preconditioning of problem matrices. The advantage of the method is that it relies on solving least-squares problems, probably one of the most studied problems in numerical linear algebra, so that an abundance of fast and robust numerical techniques are available for its implementation. The benefits of using nonnegative least squares in MPC was recently investigated by the author, both for embedded linear MPC based on QP (Bemporad, 2015b) and for solving the multiparametric quadratic programming (mpQP) problems that arise in explicit MPC (Bemporad, 2015a).

To be able to use the QP solver of (Bemporad, 2015b) as the core engine for solving QP relaxations during branching, this paper first extends the algorithm to the case of equality and bilateral inequality constraints, to warm starting from previous solutions, and to compute dual lower bounds on the optimal cost.

We show in examples that the resulting MIQP solver is quite competitive with respect to commercial packages, at least to solve small-scale MIQPs that arise in typical

⋆ This work was partially supported by the H2020 European project DISIRE, Grant Agreement No. 636834, http://spire2030.eu/disire/.

embedded hybrid Model Predictive Control (MPC) applications.

### 1.1 Notation

Let $\mathbb{R}^n$ denote the set of real vectors of dimension $n$ and $\mathbb{N}$ the set of natural integers, respectively, and let $\mathcal{I} \subset \mathbb{N}$ be a finite set of integers. For a vector $a \in \mathbb{R}^n$, $a_i$ denotes the $i$-th entry of $a$, $a_{\mathcal{I}}$ the subvector obtained by collecting the entries $a_i$ for all $i \in \mathcal{I}$, $\|a\|_2$ the Euclidean norm of $a$, $\|a\|_1 = \sum_{i=1}^n |a_i|$ the 1-norm of $a$, the condition $a > 0$ is equivalent to $a_i > 0$, $\forall i = 1, \ldots, n$ (and similarly for $\geq$, $\leq$, $<$), and $\mathrm{diag}(a)$ is the diagonal matrix whose $(i,i)$-th element is $a_i$. We denote by $0_n$ the vector of $\mathbb{R}^n$ with all zero components, with the subscript $_n$ dropped whenever the dimension is clear from the context. For a matrix $A \in \mathbb{R}^{n\times m}$, $A'$ denotes its transpose, $A_i$ denotes the $i$-th row of $A$, $A_{\mathcal{I}}$ the submatrix of $A$ obtained by collecting the rows $A_i$ for all $i \in \mathcal{I}$, and $A_{\mathcal{I}\mathcal{J}}$ the submatrix of $A$ obtained by collecting the rows and columns of $A$ indexed by $i \in \mathcal{I}$ and $j \in \mathcal{J}$, respectively. Matrix $A^{\#} \in \mathbb{R}^{m\times n}$ denotes the pseudoinverse matrix of $A$, namely $AA^{\#}A = A$, $A^{\#}AA^{\#} = A^{\#}$, $AA^{\#} = (AA^{\#})'$, $A^{\#}A = (A^{\#}A)'$ (if $A$ is full column rank, $A^{\#} \triangleq (A'A)^{-1}A'$). For a square matrix $A \in \mathbb{R}^{n\times n}$, $A^{-1}$ denotes the inverse of $A$ (if it exists) and $A^{-T}$ its transpose, $A \succ 0$ ($A \succeq 0$) denotes positive definiteness (semidefiniteness) of $A$. Matrix $I_n$ denotes the identity matrix of order $n$, where sometimes the subscript $n$ is dropped if the dimension is clear from the context.

## 2. PROBLEM FORMULATION

We want to solve the following class of Mixed-Integer Quadratic Programming (MIQP) problems

$$\begin{aligned} \min_z \quad & V(z) \triangleq \frac{1}{2}z'Qz + c'z && \text{(1a)}\\ \text{s.t.} \quad & \ell \leq Az \leq u && \text{(1b)}\\ & Gz = g && \text{(1c)}\\ & \bar{A}_i z \in \{\bar{\ell}_i, \bar{u}_i\},\ i = 1, \ldots, q && \text{(1d)} \end{aligned}$$

where $Q \succ 0$ is the Hessian matrix, $Q \in \mathbb{R}^{n\times n}$, $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $\ell, u \in \mathbb{R}^m$, $\ell \leq u$, $G \in \mathbb{R}^{p\times n}$, $g \in \mathbb{R}^p$, $\bar{A} \in \mathbb{R}^{q\times n}$, $\bar{\ell}, \bar{u} \in \mathbb{R}^q$. Binary constraints $z_i \in \{0,1\}$ are a special case of (1d), obtained by setting $\bar{A}_i$ as the $i$-th row of the identity matrix, $\bar{\ell}_i = 0$, and $\bar{u}_i = 1$.

MIQP problems of the form (1) arise when formulating hybrid MPC controllers based on the following Mixed Logical Dynamical (MLD)

$$\begin{aligned} x_{k+1} &= \mathcal{A}x_k + \mathcal{B}_1 v_k + \mathcal{B}_2\delta_k + \mathcal{B}_3\zeta_k + \mathcal{B}_5 && \text{(2a)}\\ y_k &= \mathcal{C}x_k + \mathcal{D}_1 v_k + \mathcal{D}_2\delta_k + \mathcal{D}_3\zeta_k + \mathcal{D}_5 && \text{(2b)}\\ \mathcal{E}_2\delta_k + \mathcal{E}_3\zeta_k &\leq \mathcal{E}_1 v_k + \mathcal{E}_4 x_k + \mathcal{E}_5, && \text{(2c)} \end{aligned}$$

model representation, where $x_k$ is the state vector, $y_k$ is the output vector, $v_k$ is the input vector, $\zeta_k$ and $\delta_k$ are auxiliary vectors. Vector $\zeta_k$ is real-valued, $\delta_k$ is binary, $x_k$, $v_k$, $y_k$ can contain both real and binary components. Matrices $\mathcal{A}$, $\mathcal{B}_i$, $\mathcal{C}$, $\mathcal{D}_i$ and $\mathcal{E}_i$ are constant and determine the hybrid dynamics. They can be obtained automatically by high-level descriptions of the hybrid dynamics and constraints, for example by using the translation tool HYSDEL (Torrisi and Bemporad, 2004). The MLD model (2) is used to formulate the following hybrid MPC problem

$$\begin{aligned} \min_{\{v_k,\delta_k,\zeta_k\}_{k=0}^{T-1}} \sum_{k=0}^{T-1} & \|L_x(x_k - r_k^x)\|_2^2 + \|L_v(v_k - r_k^v)\|_2^2 + && \text{(3a)}\\ & \|L_\zeta(\zeta_k - r_k^\zeta)\|_2^2 + \|L_\delta(\delta_k - r_k^\delta)\|_2^2 \\ \text{s.t.} \quad & \text{MLD model (2)} && \text{(3b)}\\ & x_0 = x(t) \end{aligned}$$

that can be mapped into a MIQP problem of the form (1), with $z = \left[\, v_0' \cdots v_{N-1}'\ \delta_0' \cdots \delta_{N-1}'\ \zeta_0' \cdots \zeta_{N-1}' \,\right]' \in \mathbb{R}^n$, $\bar{\ell} = 0$, $\bar{u} = 1$, and $\bar{A}$ containing the rows of the identity matrix $I_n$ corresponding to the binary components of vector $z$.

We have assumed that matrix $Q$ in (1a) is positive definite. In many hybrid MPC formulations this is not the case, as some of the weight matrices $L_x$, $L_v$, $L_\zeta$, $L_\delta$ may not be full-rank and lead to a resulting Hessian matrix $Q \succeq 0$. In such cases, we assume the problem gets modified by adding a regularization term $\rho I_n$ to $Q$, $0 < \rho \ll 1$. We will show in Section 5.2 that this does not change the computed hybrid MPC action significantly.

## 3. EXTENDED QP SOLVER BASED ON NNLS

The core ingredient of the MIQP algorithm proposed in this paper is the QP solver developed in (Bemporad, 2015b) for minimizing strictly convex quadratic functions subject to inequality constraints. Such a QP solver is based on the idea of rephrasing a strictly convex QP problem as a Least Distance Problem (LDP) that is solved via a NNLS algorithm, and was shown very efficient in (Bemporad, 2015b) compared to existing state-of-the-art QP algorithms. In this section we extend the algorithm to handle bilateral inequalities (1b), equality constraints (1c), warm starting, and early stopping, so that it can be efficiently exploited within a branch & bound (B&B) framework. The resulting extended method to solve the QP problem (1a)–(1c) is described in Algorithm 1.

We justify the various steps of Algorithm 1 in the following sections.

### 3.1 Bilateral inequalities and equalities

The dual QP problem of (1a)–(1c) is the following convex QP

$$\begin{aligned} \max_{\lambda_\ell,\lambda_u,\mu} \quad & \Psi(\lambda_\ell,\lambda_u,\mu) \triangleq -\frac{1}{2}\begin{bmatrix}\lambda_\ell\\ \lambda_u\\ \mu\end{bmatrix}' \begin{bmatrix}-A\\ A\\ G\end{bmatrix} Q^{-1} \cdot \\ & \begin{bmatrix}-A' & A' & G'\end{bmatrix}\begin{bmatrix}\lambda_\ell\\ \lambda_u\\ \mu\end{bmatrix} - \begin{bmatrix}d_\ell\\ d_u\\ f\end{bmatrix}'\begin{bmatrix}\lambda_\ell\\ \lambda_u\\ \mu\end{bmatrix} - \frac{1}{2}c'Q^{-1}c && \text{(8a)}\\ \text{s.t.} \quad & \lambda_\ell, \lambda_u \geq 0,\ \mu \text{ free}, && \text{(8b)} \end{aligned}$$

where $\lambda_\ell, \lambda_u \in \mathbb{R}^m$, $\mu \in \mathbb{R}^p$, and $d_\ell, d_u \in \mathbb{R}^m$, $f \in \mathbb{R}^p$ are defined as follows:

$$d_\ell \triangleq -\ell - AQ^{-1}c,\ d_u \triangleq u + AQ^{-1}c,\ f \triangleq g + GQ^{-1}c. \quad \text{(8c)}$$

The following theorem shows how the QP problem (1a)–(1c) is equivalent to a least squares problem in which some of the variables are constrained to be nonnegative, extending (Bemporad, 2015b, Th. 1) to the case of equality and bilateral inequality constraints.

*Theorem 1.* Consider the QP (1a)–(1c) and let $Q \succ 0$. Let $L'L$ be a Cholesky factorization of $Q$ and define

**Algorithm 1** QP solver based on NNLS with equality constraints, bilateral inequalities, warm start, and early stop

**Input**: Inverse Cholesky factor $L^{-1}$ of $Q$; matrices $M = AL^{-1}$, $N = GL^{-1}$; vectors $c$, $v = L^{-T}c$, $d_\ell = -(\ell + Mv)$, $d_u = u + Mv$, $f = g + NL^{-T}c$; initial guess $\mathcal{P}_u, \mathcal{P}_\ell$, vectors $y_\ell, y_u, w_\ell, w_u, \nu, a$, scalar $\gamma$, $\delta$, $V$ satisfying (13), (14), (15); max tolerable value $V_0$ for the optimal cost; feasibility tolerance $\epsilon \geq 0$.

1. $k \leftarrow 0$;
2. **if** $(\min\{w_\ell, w_u\} \geq -\delta\epsilon$ **or** $\mathcal{P}_\ell \cup \mathcal{P}_u = \{1, \ldots, m\}$ **or** $a'a + \delta^2 = 0$ **or** $V > V_0)$ **and** $k > 0$ **then go to** Step 8;
3. $k \leftarrow k + 1$; $i_\ell \leftarrow \arg\min_{i \in \{1,\ldots,m\} \setminus \mathcal{P}_\ell} w_{\ell i}$, $i_u \leftarrow \arg\min_{i \in \{1,\ldots,m\} \setminus \mathcal{P}_u} w_{ui}$;
4. **if** $w_{\ell i_\ell} \leq w_{u i_u}$ **then** $\mathcal{P}_\ell \leftarrow \mathcal{P}_\ell \cup \{i_\ell\}$; $\gamma \leftarrow \gamma + |d_{\ell i_\ell}|$; **otherwise** $\mathcal{P}_u \leftarrow \mathcal{P}_u \cup \{i_u\}$; $\gamma \leftarrow \gamma + |d_{u i_u}|$;
5. $s_\ell, s_u \leftarrow 0_m$;
6. **solve** the least squares (LS) problem
$$\begin{bmatrix} s_{\ell\mathcal{P}_\ell} \\ s_{u\mathcal{P}_u} \\ s_\nu \end{bmatrix} \leftarrow \arg\min_z \left\| \begin{bmatrix} -M'_{\mathcal{P}_\ell} & M'_{\mathcal{P}_u} & N' \\ d'_{\ell\mathcal{P}_\ell} & d'_{u\mathcal{P}_u} & f' \end{bmatrix} z + \begin{bmatrix} 0_n \\ \gamma \end{bmatrix} \right\|_2^2; \tag{4}$$
7. **if** $s_{\ell\mathcal{P}_\ell}, s_{u\mathcal{P}_u} \geq 0$ **then**
$$\begin{bmatrix} y_\ell \\ y_u \\ \nu \end{bmatrix} \leftarrow \begin{bmatrix} s_\ell \\ s_u \\ s_\nu \end{bmatrix}; \; a \leftarrow M'(y_u - y_\ell) + N'\nu; \tag{5a}$$
$$\delta \leftarrow \gamma + d'_\ell y_\ell + d'_u y_u + f'\nu; \tag{5b}$$
$$\begin{bmatrix} w_\ell \\ w_u \end{bmatrix} \leftarrow Ma \begin{bmatrix} -I \\ I \end{bmatrix} + \delta \begin{bmatrix} d_\ell \\ d_u \end{bmatrix}; \tag{5c}$$
$$V \leftarrow \text{as in (14b)}; \tag{5d}$$
**go to** Step 2;
**otherwise**
$$\alpha_\ell \leftarrow \min_{h \in \mathcal{P}_\ell:\; s_{\ell h} \leq 0} \left\{ \frac{y_{\ell h}}{y_{\ell h} - s_{\ell h}} \right\};$$
$$\alpha_u \leftarrow \min_{h \in \mathcal{P}_u:\; s_{uh} \leq 0} \left\{ \frac{y_{uh}}{y_{uh} - s_{uh}} \right\};$$
$$\begin{bmatrix} y_\ell \\ y_u \\ \nu \end{bmatrix} \leftarrow \begin{bmatrix} y_\ell \\ y_u \\ \nu \end{bmatrix} + \left( \begin{bmatrix} s_\ell \\ s_u \\ s_\nu \end{bmatrix} - \begin{bmatrix} y_\ell \\ y_u \\ \nu \end{bmatrix} \right) \cdot \min\{\alpha_\ell, \alpha_u\};$$
$\mathcal{I}_\ell \leftarrow \{h \in \mathcal{P}_\ell : y_{\ell h} = 0\}, \mathcal{P}_\ell \leftarrow \mathcal{P}_\ell \setminus \mathcal{I}_\ell$;
$\gamma \leftarrow \gamma - \|d_{\ell\mathcal{I}_\ell}\|_1$;
$\mathcal{I}_u \leftarrow \{h \in \mathcal{P}_u : y_{uh} = 0\}; \mathcal{P}_u \leftarrow \mathcal{P}_u \setminus \mathcal{I}_u$;
$\gamma \leftarrow \gamma - \|d_{u\mathcal{I}_u}\|_1$;
**go to** Step 5;
8. **if** the residual in Step 6 is nonzero **then**
**if** $V \leq V_0$ **then**
$$z^* \leftarrow -L^{-1}\left(\frac{1}{\delta}a + v\right); \; \begin{bmatrix} \lambda_\ell^* \\ \lambda_u^* \\ \mu^* \end{bmatrix} \leftarrow \frac{1}{\delta} \begin{bmatrix} y_\ell \\ y_u \\ \nu \end{bmatrix}; \tag{7a}$$
$$V^* \leftarrow V; \tag{7b}$$
**otherwise** $V^* > V_0$;
**otherwise** QP problem is infeasible;
9. **end**.

**Output**: Primal solution $z^*$ of the QP problem (1a)–(1c); optimal Lagrange multipliers $\lambda_\ell^*, \lambda_u^* \geq 0$ of lower and upper bound constraints (1b) and $\mu^*$ of equality constraints (1c); optimal cost $V^*$, or infeasibility status, or proof that $V^* > V_0$; number $k$ of iterations.

$M \triangleq AL^{-1}$, $N \triangleq GL^{-1}$. Let $\gamma$ be any positive scalar. Consider the Partially Nonnegative Least Squares (PNNLS) problem

$$\min_y \quad \frac{1}{2} \left\| \begin{bmatrix} M' \\ -d'_\ell \end{bmatrix} y_\ell - \begin{bmatrix} M' \\ d'_u \end{bmatrix} y_u - \begin{bmatrix} N' \\ f' \end{bmatrix} \nu - \begin{bmatrix} 0 \\ \gamma \end{bmatrix} \right\|_2^2 \tag{9a}$$
$$\text{s.t.} \quad y_\ell, y_u \geq 0, \; \nu \text{ free} \tag{9b}$$

with $y_\ell, y_u \in \mathbb{R}^m$, $\nu \in \mathbb{R}^p$, and let

$$r^* \triangleq \begin{bmatrix} M'(y_\ell^* - y_u^*) - N'\nu^* \\ -\gamma - d'_\ell y_\ell^* - d'_u y_u^* - f'\nu^* \end{bmatrix}, \tag{10}$$

be the residual obtained at the optimal solution $(y_\ell^*, y_u^*, \nu^*)$ of (9), where $y_\ell^*, y_u^* \in \mathbb{R}^m$, $\nu^* \in \mathbb{R}^p$, and $r^* \in \mathbb{R}^{n+1}$. The following statements hold:

*i)* If $r^* = 0$ then QP (1a)–(1c) is infeasible;
*ii)* If $r^* \neq 0$ then
$$z^* \triangleq -Q^{-1}\left( c + \frac{A'(y_u^* - y_\ell^*) + G'\nu^*}{\gamma + d'_\ell y_\ell^* + d'_u y_u^* + f'\nu^*} \right) \tag{11}$$
solves QP (1a)–(1c).

*Proof.* See Appendix A. ■

### *3.2 Properties of the solution*

The following Lemma 1 shows the properties of the primal and dual solutions that one could reconstruct during the iterations of Algorithm 1 as in (7a) below, as well as the corresponding primal and dual objective functions.

*Lemma 1.* Let $y_\ell, y_u, \nu$, $a$ and $\delta$ be defined as in (5a)–(5b) for a given $\gamma > 0$, with $s_\ell, s_u \geq 0$ and $s_\nu$ obtained by solving the LS problem (4), and let $\delta > 0$. The primal cost associated with $z = -L^{-1}\left(\frac{1}{\delta}a + v\right)$, $v = L^{-T}c$, and the dual cost associated with vectors $\lambda_\ell = \frac{1}{\delta}y_\ell$, $\lambda_u = \frac{1}{\delta}y_u$, and $\mu = \frac{1}{\delta}\nu$ are such that

$$\Psi(\lambda_\ell, \lambda_u, \mu) = -\frac{1}{2}\frac{a'a}{\delta^2} - \frac{\delta - \gamma}{\delta} - \frac{1}{2}v'v \tag{12a}$$
$$V(z) = \frac{1}{2}\left(\frac{a'a}{\delta^2} - v'v\right) \tag{12b}$$
$$\Psi(\lambda_\ell, \lambda_u, \mu) = V(z) \leq V(z^*). \tag{12c}$$

*Proof.* Since $\delta > 0$ and $y_\ell = s_\ell \geq 0$, $y_u = s_u \geq 0$, we have that the triplet $(\lambda_\ell, \lambda_u, \mu)$ is feasible for the dual QP problem (8), and therefore the inequality $\Psi(\lambda_\ell, \lambda_u, \mu) \leq V(z^*)$ in (12c) is satisfied. Equality (12a) follows by (5a)–(5b) and substituting $\lambda_\ell, \lambda_u, \mu$ in the dual cost (8a). By substituting the expression for $z$ in (1a) we obtain the second equality (12b). Equality (12c) follows from (5) and from the conditions of optimality of (9) related to complimentary slackness, that is $y'_\ell w_\ell + y'_u w_u = 0$, and of $\nu$ as a function of $y_\ell, y_u$ given by (Bemporad, 2015a, Lemma 1), that is $(NN' + ff')\nu - NM'y_\ell + NMy_u + fd'_\ell y_\ell + fd'_u y_u^* + \gamma f = 0$ (cf. the proof of Theorem 1 in Appendix A). ■

In case $\delta > 0$, by (12c) of Lemma 1 the quantity $z$ as in (7a) is always super-optimal during the iterations of Algorithm 1 (and, if it is strictly super-optimal, is also necessarily infeasible); it only becomes optimal (and therefore feasible) when the algorithm terminates with a nonzero residual.

### 3.3 Warm starting

We consider as a valid warm start the initial active sets $\mathcal{P}_u, \mathcal{P}_\ell \subseteq \{1,\dots,m\}$, initial guess $y_\ell, y_u, w_\ell, w_u \in \mathbb{R}^m$ and $\nu \in \mathbb{R}^p$ for the PNNLS problem (9) satisfying the following conditions:

$$\mathcal{P}_u \cap \mathcal{P}_\ell = \emptyset \tag{13a}$$
$$y_\ell, y_u \geq 0, \tag{13b}$$
$$\nu = -\begin{bmatrix} N' \\ f' \end{bmatrix}^{\#} \begin{bmatrix} M'(y_u - y_\ell) \\ \delta \end{bmatrix} \tag{13c}$$
$$w_\ell = -MM'(y_u - y_\ell) + \delta d_\ell \tag{13d}$$
$$w_u = MM'(y_u - y_\ell) + \delta d_u \tag{13e}$$
$$y_\ell' w_\ell = y_u' w_u = 0 \tag{13f}$$
$$y_{\ell i} \geq 0,\ w_{\ell i} = 0,\ \forall i \in \mathcal{P}_\ell, \tag{13g}$$
$$y_{ui} \geq 0,\ w_{ui} = 0,\ \forall i \in \mathcal{P}_u \tag{13h}$$
$$y_{\ell i} = 0,\ \forall i \in \{1,\dots,m\} \setminus \mathcal{P}_\ell \tag{13i}$$
$$y_{ui} = 0,\ \forall i \in \{1,\dots,m\} \setminus \mathcal{P}_u, \tag{13j}$$

where

$$\delta = d_\ell' y_\ell + d_u' y_u + \gamma \tag{13k}$$

along with the following quantities

$$a = M'(y_u - y_\ell) + N'\nu \tag{14a}$$
$$V = \frac{1}{2}\left(\frac{a'a}{\delta^2} - v'v\right). \tag{14b}$$

See the proof of Theorem 1 for a justification of conditions (13) and of Lemma 1 for conditions (14).

### 3.4 Early stopping criteria

By exploiting the result of Lemma 1, Algorithm 1 has been formulated for solving the QP problem (1a)–(1c) only if the optimal solution $V(z^*) \leq V_0$, where $V_0$ is a given value (possibly $V_0 = +\infty$). This is of particular importance, in that it allows to halt immediately Algorithm 1 at Step 2 after computing (5) at Step 7, in case the quantity $V$ in (14b) is greater than $V_0$. This feature will be particularly useful in the B&B setting described in next Section 4.

The following Corollary 1 of Theorem 1 (that extends (Bemporad, 2015b, Corollary 2) to the equality constrained case) justifies the stopping condition $a'a + \delta^2 = 0$ at Step 2, providing a simple yet very effective criterion to early detecting the infeasibility of the QP problem (1a)–(1c). In the numerical implementation of Algorithm 1, the condition $a'a + \delta^2 = 0$ is replaced by $a'a + \delta^2 \leq \epsilon_{\text{infeas}}$, where $\epsilon_{\text{infeas}} > 0$ is a small tolerance.

*Corollary 1.* Let $s_\ell, s_u$ a solution of problem (4), let $s_\ell, s_u$, and let $a, \delta$ be defined as in (5). If $a'a + \delta^2 = 0$ then the QP problem (1a)–(1c) is infeasible.

*Proof.* The quantity $a'a + \delta^2$ is equal to the residual of problem (4). As proved in part $i$) of Theorem 1, if such a residual is zero then the polyhedron $\mathcal{C} \triangleq \{u \in \mathbb{R}^n : M_{\mathcal{P}_u} u \leq u, M_{\mathcal{P}_\ell} u \geq \ell, Nu = f\}$ is empty. Hence, also the polyhedron $\{u \in \mathbb{R}^n :\ Mu \leq u, Mu \geq \ell, Nu = f\} = \mathcal{C} \cap \{u \in \mathbb{R}^n :\ M_{\{1,\dots,m\}\setminus\mathcal{P}_u} u \leq u, M_{\{1,\dots,m\}\setminus\mathcal{P}_\ell} u \geq \ell, Nu = f\}$ is empty, and therefore problem (1a)–(1c) is infeasible. ■

### 3.5 Improving numerical robustness

The basic NNLS algorithm of Lawson and Hanson (1974) is formulated for $\gamma = 1$. As suggested in (Bemporad, 2015b), we choose here to adapt $\gamma$ during iterations to the following value

$$\gamma = 1 + \|f\|_1 + \|d_{\ell\mathcal{P}_\ell}\|_1 + \|d_{u\mathcal{P}_u}\|_1 \tag{15}$$

which provides better numerical conditioning.

Although less critical than with iterative methods like accelerated gradient projection (Patrinos and Bemporad, 2014) and ADMM (Boyd et al., 2011), preconditioning the MIQP problem (1) sometimes ensures a better numerical robustness of Algorithm 1. However, contrary to iterative methods, in active set methods the effect of scaling the variables of the problem is much less critical, as the total number of iterations may decrease, remain constant, or even increase sometimes. In case preconditioning is needed, we suggest here to use Jacobi diagonal scaling of the inequality constraints (1b) defined as follows (Bertsekas, 2009):

$$\theta_i \triangleq \frac{1}{\|M_i\|_2} \tag{16a}$$
$$M \leftarrow \text{diag}(\theta_1,\dots,\theta_m)M \tag{16b}$$
$$\ell_i \leftarrow \theta_i \ell_i \tag{16c}$$
$$u_i \leftarrow \theta_i u_i,\ i = 1,\dots,m. \tag{16d}$$

## 4. MIQP SOLVER BASED ON NNLS

The B&B Algorithm 2 solves the convex MIQP problem (1) by exploiting the features offered by Algorithm 1.

The sets $\mathcal{Q}_\ell, \mathcal{Q}_u$ initialized at Step 2 represent the sets of indices corresponding to the equality constraints induced by setting, respectively, $\bar{A}_i z = \bar{\ell}_i$ or $\bar{A}_i z = \bar{u}_i$ during branching. The tuple $\mathcal{A}$ collects the input arguments to Algorithm 1, while $\mathcal{S}$ is an ordered list of tuples, corresponding to the equality-constrained QP's that remain to be solved. At Step 3.1 the last element $\mathcal{A}$ of $\mathcal{S}$ is extracted to solve the corresponding MIQP relaxation, corresponding to a depth-first search.

After executing Algorithm 1 at Step 3.2, the final values of $\mathcal{P}_\ell, \mathcal{P}_u, M, y_\ell, y_u, w_\ell, w_u, \nu, d_\ell, d_u, \delta, \gamma, a, V^*$ are kept and used (if needed) to warm start the subsequent QP relaxations. Note that in this case $V^*$ becomes a lower bound for all children QP problems, as these include additional equality constraints. Step 3.3.1 is only executed if the QP relaxation was feasible and did not halt because the condition $V > V_0$ was satisfied (that is, the relaxation was proven to a worse cost than the cost of the best integer-feasible solution found so far). In this case, Step 3.3.1 checks whether all integrality constraints are satisfied, where the quantity $t_i \triangleq -\frac{1}{\delta}\bar{M}_i(a - L^{-T}c) = \bar{A}_i z^*$, and eventually updates the best known integer-feasible solution $\zeta^*$ and its corresponding cost $V_0$. Otherwise, branching is executed at Steps 3.3.3.1–3.3.3.8 by picking up the index $i_b$ corresponding to the constraint $\bar{A}_i z$ that is most distant from $\bar{\ell}_i, \bar{u}_i$ (Step 3.3.3.1). Such a constraint is moved from the set of inequality constraints to the set of equality constraints at Step 3.3.3.2, and two new MIQP relaxations $\mathcal{A}_0$, $\mathcal{A}_1$ are formed at Step 3.3.3.7.

To prove that the warm-starting vectors in $\mathcal{A}_0$, $\mathcal{A}_1$ satisfy (13)–(15), note that $y_\ell, y_u, \nu$ are generated at Step 7 of Algorithm 1 after solving the LS problem (4). Hence, $(y_\ell, y_u, \nu_0)$ and $(y_\ell, y_u, \nu_1)$ satisfy (13c). In addition, $y_\ell$, $y_u$, $w_\ell$, $w_u$ are an output of Algorithm 1, so they satisfy the conditions of QP optimality, and in particular (13b), that remains satisfied after executing Step 3.3.3.4, and (13d)–(13i). Clearly, the warm-start combination of vectors may not be optimal, due to forcing the equality constraint $\bar{A}_{i_b} z = \bar{\ell}_i$ (or $\bar{A}_{i_b} z = \bar{u}_i$) in the active set of the new relaxed QP problem.

Step 3.3.3.8 chooses to solve first the MIQP relaxation corresponding to the quantity $t_{i_b}$ that is closest to the lower value $\ell_{i_b}$ or upper value $u_{i_b}$. When no relaxations are left to solve, Step 4 checks whether the initial upper-bound $V_0$ has remained $+\infty$, in which case no integer feasible solution was found.

### *4.1 Double-inequality formulation*

In Steps 3.3.3.2–3.3.3.3, Algorithm 2 fixes binary constraints (1d) as either the *equality* constraint $\bar{A}_i z = \bar{\ell}_i$ or $\bar{A}_i z = \bar{u}_i$. An alternative approach to equivalently fix $\bar{A}_i z = \bar{\ell}_i$ is to transform the relaxed inequality $\bar{A}_i z \leq \bar{u}_i$ into the *inequality* $\bar{A}_i z \leq \bar{\ell}_i$ (and, similarly $\bar{A}_i z \geq \bar{\ell}_i$ into $\bar{A}_i z \geq \bar{u}_i$ to fix $\bar{A}_i z = \bar{u}_i$). To change the lower bound from $\bar{\ell}_i$ to $\bar{u}_i$ one simply subtracts the quantity $\bar{\ell}_i - \bar{u}_i$ from $d_{\ell i}$ (and, similarly $\bar{u}_i - \bar{\ell}_i$ from $d_{ui}$) before creating the children relaxed QP problems at Step 3.3.3.7. In this way, matrices $N_b \equiv N_0$, $M_b \equiv M$, $f_0 = f_1 \equiv f$ remain constant during the execution of the algorithm, as it is enough to update only vectors $d_{\ell i}$, $d_{ui}$. While the approach is appealing for its simplicity, it may lead to a possible increased number of iterations when solving the QP relaxation via Algorithm 1. Moreover, the unavoidable introduction of a tolerance $\varepsilon$ when imposing the upper bound $\bar{A}_i z \leq \bar{\ell}_i + \varepsilon$ (and, similarly, $\bar{A}_i z \geq \bar{u}_i - \varepsilon$) must be taken care of accurately to avoid numerical issues.

## 5. NUMERICAL RESULTS

In this section we report numerical experiments obtained on a Macbook Pro 3GHz Intel Core i7 with 16GB RAM running MATLAB R2014b.

### *5.1 Random mixed-integer quadratic programs*

We compare the performance of the MIQP solver developed in the previous sections (labeled as NNLS) against the commercial state-of-the-art MIQP solver of GUROBI v6.0 (Gurobi Optimization, Inc., 2014) and of CPLEX (IBM, Inc., 2014) with default options. Presolvers were disabled for fairness of comparison, although they do not contribute significantly to change computation time (sometimes even worsen it). Algorithm 1 has been implemented in Embedded MATLAB code and compiled, Algorithm 2 is run in interpreted MATLAB code.

Table 1 shows the CPU time obtained for solving feasible MIQP problems with $n$ variables, $m$ bilateral linear inequality constraints, $p$ binary constraints of the form $z_i \in \{0, 1\}$, and condition number $\kappa = 10^4$ of the primal Hessian $Q$ [1]. The reported time is the worst-case obtained over 20 instances for each triplet $n$, $m$, $p$. Problem (4) in Algorithm 1 is solved by updating the LDL$^T$ decomposition

**Algorithm 2** MIQP solver based on NNLS

**Input**: MIQP problem matrices $Q = Q' \succ 0$, $A$, $G$, $\bar{A}$ and vectors $\ell$, $u$, $g$, $\bar{\ell}$, $\bar{u}$; feasibility tolerance $\epsilon \geq 0$.

1. Compute inverse Cholesky factorization $Q^{-1} = L^{-1}L^{-T}$; $v \leftarrow L^{-T}c$;
2. **set** $M_0 \leftarrow \left[\begin{smallmatrix} \bar{A} \\ A \end{smallmatrix}\right] L^{-1}$, $d_\ell = -\left[\begin{smallmatrix} \bar{\ell} \\ \ell \end{smallmatrix}\right] - M_0 L^{-T}c$, $d_u = \left[\begin{smallmatrix} \bar{u} \\ u \end{smallmatrix}\right] + M_0 L^{-T}c$;
   $N_0 \leftarrow GL^{-1}$, $f \leftarrow g + N_0 L^{-T}c$;
   $\mathcal{P}_\ell, \mathcal{P}_u \leftarrow \emptyset$, $y_\ell, y_u, w_\ell, w_u \leftarrow 0_{q+m}$;
   $\delta, \gamma \leftarrow 1 + \|f\|_1$, $\nu \leftarrow -(N_0 N_0' + ff')^{-1} f\gamma$;
   $a \leftarrow M'(y_u - y_\ell) + N'\nu$;
   $V \leftarrow \frac{1}{2}\left(\frac{a'a}{\delta^2} - v'v\right)$; $V_0 \leftarrow +\infty$;
   $\zeta^* \leftarrow \emptyset$; $\mathcal{Q}_\ell, \mathcal{Q}_u \leftarrow \emptyset$;
   $\mathcal{A} \leftarrow \{\mathcal{P}_\ell, \mathcal{P}_u, M_0, N_0, f, y_\ell, y_u, w_\ell, w_u, \nu, d_\ell, d_u, \delta, \gamma, a, V, \mathcal{Q}_\ell, \mathcal{Q}_u\}$;
   $\mathcal{S} \leftarrow \{\mathcal{A}\}$;
3. **while** $\mathcal{S} \neq \emptyset$ **do**:
   3.1. **get** last element $\mathcal{A} \in \mathcal{S}$; **set** $\mathcal{S} \leftarrow \mathcal{S} \setminus \{\mathcal{A}\}$;
   3.2. **execute** Algorithm 1 with input from $\mathcal{A}$ and **get** $z^*, V^*, \mathcal{P}_\ell, \mathcal{P}_u, y_\ell, y_u, w_\ell, w_u, \nu, d_\ell, d_u, \delta, \gamma, a$;
   3.3. **if** QP problem feasible **and** $V^* \leq V_0$ **then**
      3.3.1. **if** $(\mathcal{Q}_\ell \cup \mathcal{Q}_u = \{1, \ldots, q\}$ **or** $t_i \triangleq -\frac{1}{\delta}\bar{M}_i(a - L^{-T}c) \in \{\ell_i, u_i\}$, $\forall i = 1, \ldots, q)$ **then** $V_0 \leftarrow V^*$, $\zeta^* \leftarrow z^*$; **otherwise**
         3.3.3.1. $i_b \leftarrow \arg\min_{i \in \{1,\ldots,q\} \setminus (\mathcal{Q}_\ell \cup \mathcal{Q}_u)} \left|t_i - \frac{\ell_i + u_i}{2}\right|$;
         3.3.3.2. $\mathcal{J} \leftarrow \mathcal{Q}_\ell \cup \mathcal{Q}_u \cup \{i_b\}$; $\mathcal{I} \leftarrow \{1, \ldots, q\} \setminus \mathcal{J}$;
         3.3.3.3. $N_b \leftarrow \left[\begin{smallmatrix} N \\ \bar{M}_\mathcal{J} \end{smallmatrix}\right]$; $M_b \leftarrow \left[\begin{smallmatrix} \bar{M}_\mathcal{I} \\ M \end{smallmatrix}\right]$;
         3.3.3.4. $y_\ell \leftarrow$ vector of components of $y_\ell$ after eliminating the component $y_{\ell b}$ corresponding to $i_b$ (same for $y_u, w_\ell, w_u$);
         3.3.3.5. $f_0 \leftarrow \begin{bmatrix} f \\ -\bar{\ell}_{\mathcal{Q}_\ell} \\ \bar{u}_{\mathcal{Q}_u} \\ -\bar{\ell}_{i_b} \end{bmatrix} + N_b L^{-T}c$;
                  $f_1 \leftarrow \begin{bmatrix} f \\ -\bar{\ell}_{\mathcal{Q}_\ell} \\ \bar{u}_{\mathcal{Q}_u} \\ \bar{u}_{i_b} \end{bmatrix} + N_b L^{-T}c$;
         3.3.3.6. $\nu_0 \leftarrow \left[\begin{smallmatrix} \nu \\ y_{\ell b} \end{smallmatrix}\right]$; $\nu_1 \leftarrow \left[\begin{smallmatrix} \nu \\ y_{ub} \end{smallmatrix}\right]$;
         3.3.3.7. $\mathcal{A}_0 \leftarrow \{\mathcal{P}_\ell, \mathcal{P}_u, M_b, N_b, f_0, y_\ell, y_u, w_\ell, w_u, \nu_0, d_\ell, d_u, \delta, \gamma, a, V^*, \mathcal{Q}_\ell \cup \{i_b\}, \mathcal{Q}_u\}$;
                  $\mathcal{A}_1 \leftarrow \{\mathcal{P}_\ell, \mathcal{P}_u, M_b, N_b, f_1, y_\ell, y_u, w_\ell, w_u, \nu_1, d_\ell, d_u, \delta, \gamma, a, V^*, \mathcal{Q}_\ell, \mathcal{Q}_u \cup \{i_b\}\}$;
         3.3.3.8. **if** $t_{i_b} \leq \frac{\ell_i + u_i}{2}$ **then** $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{A}_1, \mathcal{A}_0\}$; **otherwise** $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{A}_0, \mathcal{A}_1\}$;
4. **if** $V_0 = +\infty$ **then** (1) is infeasible **otherwise** $\mathcal{V}^* \leftarrow V_0$;
5. **end**.

**Output**: Solution $\zeta^*$ of the MIQP problem (1), optimal cost $\mathcal{V}^*$, or infeasibility status.

[1] The entries of matrix $A$ are generated from the normal distribution $\mathcal{N}(0, 0.0025)$, $\ell$, $u$ from the uniform distribution $\mathcal{U}(0, 1)$, $c$ from $\mathcal{N}(0, 100)$; matrix $Q = U\Sigma V'$, where $U, V$ are orthogonal matrices generated by QR decomposition of random $n \times n$ matrices, and $\Sigma$ is diagonal with nonzero entries having logarithms equally spaced between $\pm \log(\kappa)/4$ (Bierlaire et al., 1991).

(NNLS$_{LDL}$ in the table) of $\begin{bmatrix} -M_{\mathcal{P}_\ell} & d_{\ell\mathcal{P}_\ell} \\ M_{\mathcal{P}_u} & d_{u\mathcal{P}_u} \\ N & f \end{bmatrix} \begin{bmatrix} -M'_{\mathcal{P}_\ell} & M'_{\mathcal{P}_u} & N' \\ d'_{\ell\mathcal{P}_\ell} & d'_{u\mathcal{P}_u} & f' \end{bmatrix}$ recursively as described in (Bemporad, 2015b) when the dimension of vector $\begin{bmatrix} y_\ell \\ y_u \\ \nu \end{bmatrix}$ is smaller or equal than $n$, and, for improved numerical robustness, QR factorization in case more than $n$ elements must be optimized in Problem (4). As an alternative, we purely updated the QR factorization of the same matrix (NNLS$_{QR}$ in the table) recursively as described in (Lawson and Hanson, 1974, Chap. 24, Method 1).

| $n$ | $m$ | $q$ | NNLS$_{LDL}$ | NNLS$_{QR}$ | GUROBI | CPLEX |
|---|---|---|---|---|---|---|
| 10 | 5 | 2 | 2.3 | 1.2 | 1.4 | 8.0 |
| 10 | 100 | 2 | 5.7 | 3.3 | 6.1 | 31.4 |
| 50 | 25 | 5 | 4.2 | 6.1 | 14.1 | 30.1 |
| 50 | 200 | 10 | 68.8 | 104.4 | 114.6 | 294.1 |
| 100 | 50 | 2 | 4.6 | 10.2 | 37.2 | 69.2 |
| 100 | 200 | 15 | 137.5 | 365.7 | 259.8 | 547.8 |
| 150 | 100 | 5 | 15.6 | 49.2 | 157.2 | 260.1 |
| 150 | 300 | 20 | 1174.4 | 3970.4 | 1296.1 | 2123.9 |

Table 1. Worst-case CPU time (ms) on random MIQP problems over 20 instances for each combination of $n$, $m$, $q$.

It is apparent that on such a set of random MIQP problems, Algorithm 2 performs comparably well with respect to the commercial solvers GUROBI and CPLEX, especially when the number $q$ of binary constraints is small compared to $n$ and $m$, probably due to the pure B&B nature of Algorithm 2.

The results shown in Table 2 are obtained, under the same conditions, on purely binary quadratic programs ($n = q$, $m = 5n$). When turning the presolver on, in GUROBI and CPLEX the results remain rather similar.

| $n$ | $m$ | $q$ | NNLS$_{LDL}$ | NNLS$_{QR}$ | GUROBI | CPLEX |
|---|---|---|---|---|---|---|
| 2 | 10 | 2 | 5.1 | 4.0 | 0.7 | 8.4 |
| 4 | 20 | 4 | 8.9 | 4.3 | 4.5 | 16.7 |
| 8 | 40 | 8 | 19.2 | 18.0 | 37.1 | 14.7 |
| 12 | 60 | 12 | 59.7 | 57.8 | 82.3 | 47.9 |
| 20 | 100 | 20 | 483.5 | 457.7 | 566.8 | 99.6 |
| 25 | 250 | 25 | 110.4 | 93.3 | 1054.4 | 169.4 |
| 30 | 150 | 30 | 1645.4 | 1415.8 | 2156.2 | 184.5 |

Table 2. Worst-case CPU time (ms) for random binary QP problems with $n$ variables and $5m$ constraints, over 20 instances for each value of $n$ and the corresponding $m$, $q$.

### *5.2 Hybrid MPC problem*

In order to test Algorithm 2 in a hybrid MPC problem (2)–(3), we consider the hybrid control problem described in (Bemporad and Morari, 1999, Example 5.1) with all zero weights except a unit weight on the output of the system (these are the settings of the demo `bm99sim.m` in the Hybrid Toolbox for MATLAB (Bemporad, 2003)) and a prediction horizon $T$ between 2 and 10.

The regularization term $10^{-4}I$ was added on the resulting Hessian matrix $Q$ to make the resulting MIQP's positive definite. This induces a small difference in the input and output trajectories, however the norm of the difference between the entire trajectories smaller than 0.001. We compare Algorithm 2 with preconditioning (16) against GUROBI and CPLEX with presolvers enabled. The results are reported in Table 3. For $T = 10$, the MIQP problem has $n = 40$, $q = 10$ (i.e., 30 continuous variables and 10 binary variables) and $m = 160$ linear inequalities. We observed that disabling presolvers in GUROBI and CPLEX sometimes speeds up sometimes slows down the solver.

| $N$ | NNLS$_{LDL}$ | NNLS$_{QR}$ | GUROBI | CPLEX |
|---|---|---|---|---|
| 2 | 2.2 | 2.3 | 1.2 | 3.0 |
| 3 | 3.4 | 3.9 | 2.0 | 6.5 |
| 4 | 5.0 | 6.5 | 2.6 | 8.1 |
| 5 | 7.6 | 9.8 | 3.7 | 9.0 |
| 6 | 12.3 | 17.7 | 4.3 | 11.0 |
| 7 | 20.5 | 30.5 | 5.8 | 13.1 |
| 8 | 28.9 | 47.1 | 7.3 | 17.3 |
| 9 | 38.8 | 62.5 | 9.5 | 18.9 |
| 10 | 55.4 | 98.2 | 10.9 | 22.4 |

Table 3. Hybrid MPC problem: CPU time (ms) per sampling step for different prediction horizons $N$

## 6. CONCLUSIONS

In this paper we have proposed a new MIQP solver based on B&B that is tailored to embedded hybrid MPC applications. The approach extends an active set method recently developed by the author to solve QP relaxations as nonnegative least-squares problems. While the presented approach was shown effective in simulations compared to reference commercial solvers, current research is devoted to combine hybrid models and MIQP solution methods for reaching even higher degrees of effectiveness.

## REFERENCES

Axehill, D. and Hansson, A. (2006). A mixed integer dual quadratic programming algorithm tailored for MPC. In *Proc. 45th IEEE Conference on Decision and Control*, 5693–5698. San Diego, CA, USA.

Bemporad, A. (2003). *Hybrid Toolbox – User's Guide*. `http://cse.lab.imtlucca.it/~bemporad/hybrid/toolbox`.

Bemporad, A. (2015a). A multiparametric quadratic programming algorithm with polyhedral computations based on nonnegative least squares. *IEEE Trans. Automatic Control*. In press.

Bemporad, A. (2015b). A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Trans. Automatic Control*. Conditionally accepted for publication.

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

Bertsekas, D. (2009). *Convex Optimization Theory*. Athena Scientific.

Bierlaire, M., Toint, P., and Tuyttens, D. (1991). On iterative algorithms for linear ls problems with bound constraints. *Linear Algebra and Its Applications*, 143, 111–143.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), 1–122.

Fair Isaac Corporation (2015). *FICO Xpress Optimization Suite*. `http://www.fico.com/`.

Fletcher, R. and Leyffer, S. (1998). Numerical experience with lower bounds for MIQP branch-and-bound. *SIAM J. Optim.*, 8(2), 604–616.

Floudas, C.A. (1995). *Nonlinear and Mixed-Integer Optimization*. Oxford University Press.

Frick, D., Domahidi, A., and Morari, M. (2015). Embedded optimization for mixed logical dynamical systems. *Computers & Chemical Engineering*, 72, 21–33.

Gurobi Optimization, Inc. (2014). *Gurobi Optimizer Reference Manual*. URL `http://www.gurobi.com`.

IBM, Inc. (2014). *IBM ILOG CPLEX Optimization Studio 12.6 – User Manual*.

Lawson, C. and Hanson, R. (1974). *Solving least squares problems*, volume 161. SIAM.

Patrinos, P. and Bemporad, A. (2014). An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Trans. Automatic Control*, 59(1), 18–33.

Rockafellar, R. (1970). *Convex Analysis*. Princeton University Press.

Torrisi, F. and Bemporad, A. (2004). HYSDEL — A tool for generating computational hybrid models. *IEEE Trans. Contr. Systems Technology*, 12(2), 235–249.

## APPENDIX A: PROOF OF THEOREM 1

The proof extends the proof reported in (Bemporad, 2015b) to the case of equality bilateral inequality constraints. First, by defining $u \triangleq Lz + L^{-T}c$, we complete the squares in (1a) by substituting $z = L^{-1}u - Q^{-1}c$ and recast (1) into the equivalent constrained Least Distance Problem (LDP)

$$\min_u \quad \frac{1}{2}\|u\|^2 \tag{17a}$$
$$\text{s.t.} \quad \mathcal{M}u \le d \tag{17b}$$
$$Nu = f \tag{17c}$$

where $\mathcal{M} \triangleq \left[\begin{smallmatrix} -M \\ M \end{smallmatrix}\right]$, $d \triangleq \left[\begin{smallmatrix} d_\ell \\ d_u \end{smallmatrix}\right]$.

*i*) Assume the optimal residual $r^* = 0$ in (10). Let $\nu_+^* \triangleq \max\{\nu^*, 0\}$, $\nu_-^* \triangleq \max\{-\nu^*, 0\}$ be the positive and negative parts of $\nu^*$, $\nu^* = \nu_+^* - \nu_-^*$, $\nu_+^*, \nu_-^* \ge 0$. Then, by (10) we get

$$\begin{aligned} \mathcal{M}'y^* + N'\nu_+^* - N'\nu_-^* &= 0 \\ d'y^* + f'\nu_+^* - f'\nu_-^* &= -\gamma \\ y^*, \nu_+^*, \nu_-^* &\ge 0 \end{aligned} \tag{18}$$

where $y^* \triangleq \left[\begin{smallmatrix} y_\ell^* \\ y_u^* \end{smallmatrix}\right]$. By Farkas's Lemma(Rockafellar, 1970, p. 201), for any $\gamma > 0$ (18) is equivalent to infeasibility of

$$\begin{aligned} \mathcal{M}u &\le d \\ Nu &\le f \\ -Nu &\le -f \end{aligned} \tag{19}$$

which is obviously equivalent to (17b)–(17c). Therefore the LDP problem (17) does not admit a solution, and consequently (1).

*ii*) Consider the KKT conditions for problem (9)

$$-\left[\begin{smallmatrix} \mathcal{M} & d \\ N & f \end{smallmatrix}\right]\left[\begin{smallmatrix} -\mathcal{M}'y^* - N'\nu^* \\ -d'y^* - f'\nu^* - \gamma \end{smallmatrix}\right] - \left[\begin{smallmatrix} I \\ 0 \end{smallmatrix}\right] w^* = 0 \tag{20a}$$
$$(y^*)'w^* = 0 \tag{20b}$$
$$\nu^* \text{ free}, \; w^* \ge 0, y^* \ge 0 \tag{20c}$$

where $w^* \triangleq \left[\begin{smallmatrix} w_\ell^* \\ w_u^* \end{smallmatrix}\right]$ is the optimal dual variable for problem (9). From (20a) we get

$$-[\mathcal{M} \; d]\, r^* - w^* = 0 \tag{21a}$$
$$-[N \; f]\, r^* = 0 \tag{21b}$$

and hence the condition $r^* \ne 0$, (20a)–(20b) and (21) imply that

$$\begin{aligned} 0 < (r^*)'r^* &= (r^*)'\left[\begin{smallmatrix} -\mathcal{M}' \\ -d' \end{smallmatrix}\right] y^* + (r^*)'\left[\begin{smallmatrix} N' \\ f' \end{smallmatrix}\right]\nu^* - \gamma r_{n+1}^* \\ &= (w^*)'y^* - \gamma r_{n+1}^* = -\gamma r_{n+1}^*, \end{aligned}$$

i.e., $r_{n+1}^* = -d'y^* - f'\nu^* - \gamma < 0$. By letting

$$u^* \triangleq -\frac{1}{r_{n+1}^*} r_{\{1,\dots,n\}}^* = -\frac{\mathcal{M}'y^* + N'\nu^*}{\gamma + d'y^* + f'\nu^*}, \tag{22}$$

from (20c) and (21a) we obtain

$$0 \le w^* = -[\mathcal{M} \; d]\, r^* = -r_{n+1}^* [\mathcal{M} \; d] \begin{bmatrix} \frac{r_{\{1,\dots,n\}}^*}{r_{n+1}^*} \\ 1 \end{bmatrix}$$

and hence $-\mathcal{M}u^* + d \ge 0$, or equivalently $u^*$ satisfies (17b). Moreover,

$$Nu^* - f = -N\frac{\mathcal{M}'y^* + N'\nu^*}{\gamma + d'y^* + f'\nu^*} - f = 0$$

iff $0 = N\mathcal{M}'y^* + NN'\nu^* + \gamma f + fd'y^* + ff'\nu^* = (NN' + ff')\nu^* + (N\mathcal{M}' + fd')y^* + \gamma f$, or equivalently iff

$$\nu^* = -\begin{bmatrix} N' \\ f' \end{bmatrix}^{\#} \left( \begin{bmatrix} \mathcal{M}' \\ d' \end{bmatrix} y^* + \begin{bmatrix} 0 \\ \gamma \end{bmatrix} \right). \tag{23}$$

Since by (Bemporad, 2015a, Lemma 1) condition (23) is always satisfied at optimality of (9), we have proved that $u^*$ also satisfies (17c) and therefore is a feasible candidate to solve (17). It remains to prove that $u^*$ is also optimal for (17). To this end, consider the remaining KKT conditions of optimality for problem (17)

$$u^* + \mathcal{M}'\lambda^* + N'\mu^* = 0 \tag{24a}$$
$$(\lambda^*)'(\mathcal{M}u^* - d) = 0 \tag{24b}$$
$$\lambda^* \ge 0, \; \nu^* \text{ free}. \tag{24c}$$

Let

$$\lambda^* \triangleq -\frac{1}{r_{n+1}^*} y^*, \; \mu^* \triangleq -\frac{1}{r_{n+1}^*}\nu^*. \tag{25}$$

By negativity of $r_{n+1}^*$ and nonnegativity of $y^*$ we get $\lambda^* \ge 0$. Moreover, by recalling (22), we get

$$u^* = \frac{1}{r_{n+1}^*}(\mathcal{M}'y^* + N'\nu^*) = -\mathcal{M}'\lambda^* - N'\nu^*$$

so that also (24a) is satisfied. To prove (24b) we observe that $(\lambda^*)'(\mathcal{M}u^* - d) = -\frac{1}{r_{n+1}^*}(\lambda^*)'(\mathcal{M}r_{\{1,\dots,n\}}^* + dr_{n+1}^*)$ $= \frac{1}{(r_{n+1}^*)^2}(y^*)'[\mathcal{M} \; d]\, r^* = -\frac{1}{(r_{n+1}^*)^2}(y^*)'w^* = 0$ because of (21a) and (20b). In conclusion, $u^*$ is the optimal solution of problem (17), and hence the vector $z^*$ defined in (11) solves (1a)–(1c). ■