51st IEEE Conference on Decision and Control
December 10-13, 2012. Maui, Hawaii, USA

An Accelerated Dual Gradient-Projection Algorithm for Linear Model Predictive Control

Panagiotis Patrinos and Alberto Bemporad

Abstract— **This paper proposes a dual fast gradient-projection method for solving quadratic programming problems that arise in linear model predictive control with general polyhedral constraints on inputs and states. The proposed algorithm is quite suitable for embedded control applications in that: (1) it is extremely simple and easy to code; (2) the number of iterations to reach a given accuracy in terms of optimality and feasibility of the primal solution can be estimated quite tightly; (3) the computational cost per iteration increases only linearly with the prediction horizon; and (4) the algorithm is also applicable to linear time-varying (LTV) model predictive control problems, with an extra on-line computational effort that is still linear with the prediction horizon.**

I. INTRODUCTION

Model Predictive Control (MPC) is continuously gaining popularity in industry to solve a very wide spectrum of control problems, due to its ability to explicitly optimize closed-loop performance and to take into account constraints on command inputs, internal states, and outputs [1]. The key enabler for the spread of MPC in industry is the availability of optimization algorithms, typically quadratic programming (QP) solvers, that (1) can provide a solution within the available sampling interval; (2) require little memory to store the data defining the optimization problem and the code implementing the algorithm itself; (3) the control code is simple enough to be software-certifiable; (4) the worst-case execution time of the algorithm is well predictable to satisfy hard real-time system requirements.

During the last years tremendous efforts were devoted by researchers to develop algorithms that address the above requirements, and to date many good algorithms and packages for QP are available to solve linear MPC problems, mainly active-set methods [2], interior-point methods [3] and Newton methods [4]. A different approach, which is limited to relatively small problems, was taken in [5], where multiparametric quadratic programming is used to pre-solve off-line the QP to obtain the optimal command input as an explicit (continuous and piecewise affine) function of the state vector.

In this paper we propose an algorithm based on the fast gradient method of [6] for linear (possibly time-varying) MPC problems with general polyhedral constraints on inputs and states that satisfies the above requirements. Specifically, the fast gradient method is applied to the dual problem resulting by relaxing the inequality constraints. Global convergence rates of $O(1/\nu^2)$ (ν is the iteration counter) are provided not only for dual optimality but also for primal properties of optimality and feasibility, that is what matters in MPC applications. Two practical ways are provided for certifying computational complexity (i.e., the worst-case number of iterations to achieve certain accuracy). In addition, concrete termination criteria that do not involve unknown quantities such as the optimal value, are given that guarantee a pre-specified accuracy. Furthermore, an efficient way of calculating the gradient of the dual function is described whose complexity grows only linearly with the prediction horizon.

Within the MPC community, fast-gradient methods were also rediscovered very recently in [7], [8]. In [7] it was shown how to certify the computational complexity of the method of [6] when applied to MPC. However, the method is applied to the primal problem, which limits the study to input-constrained MPC and simple enough constraint sets (e.g., a box). In [8] the fast gradient method is applied to the dual of the MPC problem, where the equality constraints corresponding to the state equations have been relaxed. This framework can only handle bound input and state constraints and a particular class of ellipsoidal constraints, and complexity certification guarantees are provided only for the dual cost.

Note: Due to lack of space, the proofs of the results of this paper are given in the internal report [9] which is available upon request from the authors.

II. NOTATION

Let $\mathbb{R}$, $\mathbb{N}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$, $\mathbb{S}^n_+$, $\mathbb{S}^n_{++}$ denote the field of real numbers, the set of non-negative integers, the set

This work was partially supported by the HYCON2: Highly-complex and networked control systems, Network of Excellence, FP7-IST contract no. 257462 and by the European project E-PRICE: Price-based Control of Electrical Power Systems, FP7-IST contract no. 249096.

P. Patrinos & A. Bemporad are with IMT Institute for Advanced Studies Lucca, Piazza San Ponziano 6, 55100 Lucca, Italy. {panagiotis.patrinos, alberto.bemporad}@imtlucca.it

978-1-4673-2066-5/12/$31.00 ©2012 IEEE

of column real vectors of length n, the set of m by n real matrices, the set of symmetric positive semidefinite and positive definite n by n matrices, respectively. The transpose of a matrix $A \in \mathbb{R}^{m\times n}$ is denoted by A'. For any nonnegative integers $k_1 \leq k_2$ the finite set $\{k_1,\dots,k_2\}$ is denoted by $\mathbb{N}_{[k_1,k_2]}$. If $k_1 = 0$ we simply write $\mathbb{N}_{k_2}$. For $z \in \mathbb{R}^n$, $[z]_+$ denotes its Euclidean projection on the nonnegative orthant, i.e., the vector whose i-th coordinate is $\max\{z_i, 0\}$. For a vector $z \in \mathbb{R}^n$, $\|z\|$ denotes the Euclidean norm of z, while if $A \in \mathbb{R}^{m\times n}$, $\|A\|$ denotes the spectral norm of A (unless otherwise stated). For a set $C \subseteq \mathbb{R}^n$, $\operatorname{int} C$, $\operatorname{rint} C$, denote its interior and relative interior, respectively. The graph of a set-valued mapping $\mathcal{S} : \mathbb{R}^d \rightrightarrows \mathbb{R}^n$, is the set $\operatorname{gph}\mathcal{S} = \{(p,z)|z \in \mathcal{S}(p)\}$, while its domain is $\operatorname{dom}\mathcal{S} = \{p|\mathcal{S}(p) \neq \emptyset\} = \{p|\exists\ z \text{ s.t. } (p,z) \in \operatorname{gph}\mathcal{S}\}$.

III. Basic setup

Consider the Model Predictive Control (MPC) formulation

$$V^\star(p) = \min \sum_{k=0}^{N-1} \ell(x_k,u_k) + V_f(x_N) \tag{1a}$$

$$\text{s.t. } x_0 = p \tag{1b}$$

$$x_{k+1} = Ax_k + Bu_k + f, k \in \mathbb{N}_{N-1} \tag{1c}$$

$$Fx_k + Gu_k \leq c, \qquad k \in \mathbb{N}_{N-1} \tag{1d}$$

$$F_N x_N \leq c_N. \tag{1e}$$

Given the current state vector $p \in \mathbb{R}^{n_x}$, the goal is to compute a state-input sequence $x_0,\dots,x_N$, $u_0\dots,u_{N-1}$, ($x_k \in \mathbb{R}^{n_x}$, $u_k \in \mathbb{R}^{n_u}$) that minimizes the finite-horizon cost (1a) over the prediction horizon N, while satisfying the constraints (1d) ($c \in \mathbb{R}^{m_s}$) and (1e) ($c_N \in \mathbb{R}^{m_N}$), with

$$\ell(x,u) = \tfrac{1}{2}\left[\begin{smallmatrix}x\\u\end{smallmatrix}\right]' \left[\begin{smallmatrix}Q & S'\\ S & R\end{smallmatrix}\right] \left[\begin{smallmatrix}x\\u\end{smallmatrix}\right] + \left[\begin{smallmatrix}q\\r\end{smallmatrix}\right]' \left[\begin{smallmatrix}x\\u\end{smallmatrix}\right], \tag{2}$$

$$V_f(x) = \tfrac{1}{2}x'Q_N x + q_N' x, \tag{3}$$

$R \in \mathbb{S}^{n_u}_{++}$, $\left[\begin{smallmatrix}Q & S'\\ S & R\end{smallmatrix}\right] \in \mathbb{S}^{n_x+n_u}_{+}$, and $Q_N \in \mathbb{S}^{n_x}_{+}$.

Let $x = [x_0' \cdots x_N']'$, $u = [u_0' \cdots u_{N-1}']'$ and $z = [x'\ u']' \in \mathbb{R}^n$ ($n = Nn_u + (N+1)n_x$). Also $V(z) = \sum_{k=0}^{N-1} \ell(x_k,u_k) + V_f(x_N)$, $\mathcal{Z}(p) = \{z \in \mathbb{R}^n | x_0 = p,\ x_{k+1} = Ax_k + Bu_k + f,\ k \in \mathbb{N}_{N-1}\}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ ($m = Nm_s + m_f$) is the affine mapping with $g_k(z) = Fx_k + Gu_k - c$, $k \in \mathbb{N}_{N-1}$, $g_N(z) = F_N x_N - c_N$. Then (1) can be expressed as

$$\mathbb{P}(p): \qquad V^\star(p) = \min_{z \in \mathcal{Z}(p)} \{V(z) | g(z) \leq 0\} \tag{4}$$

By relaxing the inequality constraints, we obtain the following problem which is dual to (4)

$$\mathbb{D}(p): \qquad \Psi^\star(p) = \max_{y \geq 0} \Psi(p,y), \tag{5}$$

where the dual function is

$$\Psi(p,y) = \min_{z \in \mathcal{Z}(p)} V(z) + y'g(z). \tag{6}$$

For $p \in \mathbb{R}^{n_x}$ let $\mathcal{Z}^\star(p)$, $\mathcal{Y}^\star(p)$ denote the set of optimal solutions of (4), (5), respectively.

Since (4) is a convex quadratic program, strong duality always holds (without the need of any constraint qualification). Therefore $V^\star(p) = \Psi^\star(p)$, and one can solve $\mathbb{D}(p)$ to obtain a dual optimal vector $y^\star(p) \in \mathcal{Y}(p)$ and then calculate a primal optimal vector $z^\star(p) \in \mathcal{Z}^\star(p)$ by solving $\min_{z \in \mathcal{Z}(p)} V(z) + (y^\star)'g(z)$. The dual function (6) can be written as

$$\begin{aligned} \Psi(p,y) = \min_{z} & \sum_{k=0}^{N-1} \bar{\ell}(x_k,u_k,y_k) + \bar{V}_f(x_N,y_N) \qquad (7) \\ \text{s.t. } & x_0 = p \\ & x_{k+1} = Ax_k + Bu_k + f, k \in \mathbb{N}_{N-1} \end{aligned}$$

where $\bar{\ell}(x,u,y_k) \triangleq \ell(x,u) + y_k' g(x,u)$, $\bar{V}_f(x,y_N) \triangleq V_f(x) + y_N' g_f(x)$. Given $p \in \mathbb{R}^{n_x}$, $y \in \mathbb{R}^m$, problem (7) is a standard finite-horizon linear-quadratic optimal control problem with positive definite input weight matrix, therefore it is well known that there exist unique sequences $x_0,\dots,x_N$, $u_0\dots,u_{N-1}$, that attain the minimum [10, Section 1.9]. Furthermore, through Danskin's theorem [10, Prop. B.25], this means that the dual function is real-valued, continuously differentiable with its gradient given by $\nabla_y\Psi(p,y) = g(z^y)$, where z^y solves (7). Finally, (7) can be seen as a parametric equality-constrained quadratic program with the parameter vector being y, therefore, z^y is a linear mapping, while $\Psi(p,\cdot) : \mathbb{R}^m \to \mathbb{R}$ is convex quadratic, thus its gradient, being an affine mapping, is Lipschitz continuous on $\mathbb{R}^m$.

IV. Accelerated dual gradient projection

In this section we will describe the Accelerated Dual Gradient Projection (GPAD for short) scheme for solving (4). The goal is to compute an $(\varepsilon_V,\varepsilon_g)$-optimal solution for (4), defined as follows.

Definition 1: Consider two nonnegative constants ε_V, ε_g. We say that $z \in \mathbb{R}^n$ is an $(\varepsilon_V,\varepsilon_g)$-optimal solution for (4) if $z \in \mathcal{Z}(p)$ and

$$V(z,p) - V^\star(p) \leq \varepsilon_V \tag{8a}$$

$$\max_{i \in \mathbb{N}_{[1,m]}} [g_i(z)]_+ \leq \varepsilon_g. \tag{8b}$$

To simplify notation, we omit the dependence on the parameter p for the rest of this section:

$$V^\star = \min_{z \in \mathcal{Z}} \{V(z) | g(z) \leq 0\}, \tag{9}$$

The dual function is $\Psi(y) = \min_{z \in \mathcal{Z}}\{V(z) + y'g(z)\}$, while the dual problem is $\Psi^\star = \max_{y \geq 0} \Psi(y)$.

The results presented in this section are valid under less stringent assumptions regarding V, g, $\mathcal{Z}$ and Ψ. Specifically, we only assume that $V : \mathbb{R}^n \to \mathbb{R}$ is a convex function, each component of $g : \mathbb{R}^n \to \mathbb{R}^m$ is convex, $\mathcal{Z} \subset \mathbb{R}^n$ is a convex set, strong duality holds and that Ψ is continuously differentiable with Lipschitz continuous gradient, i.e., there exists a $L_\Psi > 0$ such that $\|\nabla\Psi(y^1) - \nabla\Psi(y^2)\| \leq L_\Psi \|y^1 - y^2\|$. Let $\Phi(y) = -\Psi(y)$. Then $\Psi^\star = -\min_{y \geq 0} \Phi(y)$, and $\nabla\Phi(y) = -g(z^y)$, where $z^y = \arg\min_{z \in \mathcal{Z}}\{V(z) + y'g(z)\}$. GPAD (summarized in Algorithm 1) is the first fast gradient method of Nesterov [6] (see also [11], [12, Section 6.9], [13, Alg. 2], [14]) applied to the convex minimization problem $\min_{y \geq 0} \Phi(y)$.

Algorithm 1 Accelerated Dual Gradient Projection (GPAD)

```
$y_{(0)} = y_{(-1)} \in \mathbb{R}^m_+$. $\theta_0 = \theta_{-1} = 1$. $\nu \leftarrow 0$
Step 1. $w_{(\nu)} = y_{(\nu)} + \theta_\nu(\theta_{\nu-1}^{-1} - 1)(y_{(\nu)} - y_{(\nu-1)})$
Step 2. $z_{(\nu)} = \arg\min_{z \in \mathcal{Z}}\{V(z) + w'_{(\nu)} g(z)\}$
Step 3. $y_{(\nu+1)} = \left[w_{(\nu)} + \frac{1}{L_\Psi} g(z_{(\nu)})\right]_+$
Step 4. $\theta_{\nu+1} = \frac{\sqrt{\theta_\nu^4 + 4\theta_\nu^2} - \theta_\nu^2}{2}$. Set $\nu \leftarrow \nu + 1$ and go to Step 1.
```

Step 2 is the only complicated part of Algorithm 1. However, in many interesting cases $z_{(\nu)}$ can be computed exactly. For example, if (9) is a strictly convex quadratic program (V is strictly convex quadratic, $\mathcal{Z} = \mathbb{R}^n$ and g is affine), then the dual function can be computed explicitly by inverting the Hessian of the primal. In that case, computing $z_{(\nu)}$ involves just a matrix-vector product. In an MPC setting (eq. (1)), one can eliminate the state sequence, converting the problem to a strictly convex quadratic program. However, this would require the inversion of a large-scale matrix, which can be performed *offline* only once in the case of an LTI system, but may become prohibitive for LTV systems. Furthermore, the calculation of $z_{(\nu)}$ would require a matrix-vector product whose cost is of order $O(N^2)$. In Section V we show how we can implement step 2 much more efficiently.

The following theorem gives the $O(1/\nu^2)$ global convergence rate for the dual cost.

Theorem 1 ([13, Cor. 2]): Let $\{y_{(\nu)}, w_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1. Then for any $k \in \mathbb{N}_+$

$$\Phi(y_{(\nu+1)}) - \Phi(y^\star) \leq \frac{2L_\Psi}{(\nu+2)^2}\|y_{(0)} - y^\star\|^2. \qquad (10)$$

In Theorem 1 and in what follows, $y^\star$ is any element of the set of dual optimal solutions $\mathcal{Y}^\star$.

A. Primal infeasibility bound

While Algorithm 1 solves the dual of (9), we are mainly interested in finding a solution to the primal problem (9) that is $(\varepsilon_V, \varepsilon_g)$-optimal and a bound on the number of iterations it takes to find it. For the averaged primal sequence

$$\bar{z}_{(\nu)} = \left(\textstyle\sum_{i=0}^{\nu} \theta_i^{-1}\right)^{-1} \textstyle\sum_{i=0}^{\nu} \theta_i^{-1} z_{(i)} \qquad (11)$$

convergence results were reported in [15, Theorem 3], [13, Cor. 2], and [16]. However these results assume compact dual and/or primal constraint sets, and cannot be applied to convex optimization problems in which the dual constraint set is the nonnegative orthant. The next theorem provides bounds on the primal infeasibility for the averaged primal iterates (11).

Theorem 2: Let $\{y_{(\nu)}, w_{(\nu)}, z_{(\nu)}, \theta_\nu\}$ be generated by algorithm 1. Then for any $\nu \in \mathbb{N}_+$

$$\max_{i \in \mathbb{N}_{[1,m]}} [g_i(\bar{z}_{(\nu)})]_+ \leq \frac{8L_\Psi}{(\nu+2)^2}\|y_{(0)} - y^\star\|. \qquad (12)$$

B. Primal suboptimality bounds

The next theorem gives global convergence rate of $V(\bar{z}_{(\nu)})$ to $V^\star$. Notice that a lower bound on $V(\bar{z}_{(\nu)}) - V^\star$ is also relevant since $\bar{z}_{(\nu)}$ could be infeasible, therefore one may have $V(\bar{z}_{(\nu)}) \leq V^\star$ as well.

Theorem 3: Let $\{y_{(\nu)}, w_{(\nu)}, z_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1. Then for any $\nu \in \mathbb{N}_+$

$$V(\bar{z}_{(\nu)}) - V^\star \leq \frac{2L_\Psi}{(\nu+2)^2}(\|y^\star\|^2 + \|y_{(0)}\|^2), \qquad (13a)$$

$$V(\bar{z}_{(\nu)}) - V^\star \geq -\frac{8L_\Psi}{(\nu+2)^2}\|y_{(0)} - y^\star\| \, \|y^\star\|. \qquad (13b)$$

The following corollary provides a practical termination criterion for deciding $(\varepsilon_V, \varepsilon_g)$-optimality of $\bar{z}_{(\nu)}$ *without knowing* $V^\star$. Its proof follows directly from Theorems 2, 3.

Corollary 1: Let $\{y_{(\nu)}, w_{(\nu)}, z_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1. If

$$\Phi(y_{(\nu+1)}) + V(\bar{z}_{(\nu)}) \leq \varepsilon_V \qquad (14a)$$

$$\max_{i \in \mathbb{N}_{[1,m]}} [g_i(\bar{z}_{(\nu)})]_+ \leq \varepsilon_g \qquad (14b)$$

then $\bar{z}_{(\nu)}$ is an $(\varepsilon_V, \varepsilon_g)$-optimal solution for problem (9). Furthermore $V(\bar{z}_{(\nu)}) - V^\star \geq \varepsilon_g \|y^\star\|$.

Notice that in equations (12), (13a), quantities $\max_{i \in \mathbb{N}_{[1,m]}}[g_i(\bar{z}_{(\nu)})]_+$, $V(\bar{z}_{(\nu)})$ can be replaced by $\min\{\max_{i \in \mathbb{N}_{[1,m]}}[g_i(\bar{z}_{(\nu)})]_+, \max_{i \in \mathbb{N}_{[1,m]}}[g_i(z_{(\nu)})]_+\}$, $\min\{V(\bar{z}_{(\nu)}), V(z_{(\nu)})\}$ without changing their validity. This simple observation leads to Algorithm 2, which is very effective in practice according to our experience.

Algorithm 2 GPAD with termination criterion

```
**Input**: $\varepsilon_V > 0$, $\varepsilon_g > 0$
**Iterate** steps 1–4 of Algorithm 1 **until** (14) or
    $\Phi(y_{(\nu+1)}) + V(z_{(\nu)}) \le \varepsilon_V$    (15a)
    $\max_{i\in\mathbb{N}_{[1,m]}} [g_i(z_{(\nu)})]_+ \le \varepsilon_g$    (15b)
is satisfied.
```

V. Efficient calculation of the gradient

The main computational burden of GPAD lies in solving problem $\min_{z\in\mathcal{Z}(p)}\{V(p,z)+y'g(z)\}$ (Step 3 in Algorithm 1). The next proposition describes how to perform this operation very efficiently.

Proposition 1: Given $y \in \mathbb{R}^m$, the unique optimal sequence z^y for (7) is given by Algorithm 4, where K_k, D_k, M_k, d_k, L_k, C_k, s_k, $k \in \mathbb{N}_{N-1}$ can be calculated by Algorithm 3.

Algorithm 3 Factor step

```
**0:** $P_N = Q_N$.
**for** $k = N-1,\dots,0$ **do**
    **1:** $\bar{R}_k = R + B'P_{k+1}B, \bar{S}_k = S + B'P_{k+1}A,$    (16a)
        $\bar{r}_k = r + B'P_{k+1}f,$    (16b)
    **2:** $K_k = -\bar{R}_k^{-1}\bar{S}_k, D_k = -\bar{R}_k^{-1}G',$    (17a)
        $M_k = -\bar{R}_k^{-1}B', d_k = -\bar{R}_k^{-1}\bar{r}_k,$    (17b)
    **3:** $L_k = (A+BK_k)', C_k = (F_k + GK_k)',$    (18a)
        $s_k = q + K_k\bar{r}_k,$    (18b)
    **4:** $P_k = Q + A'P_{k+1}A - \bar{S}_k'\bar{R}^{-1}\bar{S}_k$    (19)
**end for**
```

Algorithm 4 Solve step

```
**0:** $e_N = F_N'y_N + q_N$    (20)
**for** $k = N-1,\dots,1$ **do**
    **1:** $e_k = L_k e_{k+1} + C_k y_k + s_k$    (21)
**end for**
**2:** $x_0 = p$
**for** $k = 0,\dots,N-1$ **do**
    **3:** $u_k = K_k x_k + D_k y_k + M_k e_{k+1} + d_k,$    (22a)
    **4:** $x_{k+1} = Ax_k + Bu_k,$    (22b)
**end for**
```

A rough flop count of Algorithm 3 (considering only operations of cubic order, i.e., matrix-matrix products, factorizations and forward-backward substitutions for linear systems with matrix-valued right hand-sides) gives $N(3n_x^3 + 6n_x^2 n_u + 6n_u^2 n_x + \frac{1}{3}n_u^3 + 2m_s n_x n_u + 2n_u^2 m_s)$ flops, which increases only linearly with the prediction horizon. Having performed the factor step, calculating the gradient of (7) is very cheap. Specifically, computing z^y takes $N(4n_x^2 + 6n_x n_u + 2m_s(n_x+n_u)) + 2m_N n_x$ flops, which again increases only linearly with the prediction horizon. Note that in the case of MPC for Linear Time-Invariant (LTI) systems, Algorithm 3 can be performed *off-line*.

VI. Calculating a tight Lipschitz constant

Calculating a tight Lipschitz constant for the gradient of the dual is important, since both the theoretical and practical convergence rate of GPAD depends on it. Utilizing [15, Theorem 1], if $\left[\begin{smallmatrix} Q & S' \\ S & R \end{smallmatrix}\right] \in \mathbb{S}_{++}^{n_x+n_u}$, then $L_\Psi = \frac{\|[\tilde{F}\ \tilde{G}]\|^2}{\mu}$ is a Lipschitz constant for $\nabla_y\Psi(p,\cdot)$. Here $\mu = \lambda_{\min}(\tilde{H})$, $\tilde{H} = \left[\begin{smallmatrix} \tilde{Q} & \tilde{S}' \\ \tilde{S} & \tilde{R} \end{smallmatrix}\right]$, $\tilde{Q} = \left[\begin{smallmatrix} I_N\otimes Q & 0 \\ 0 & Q_N \end{smallmatrix}\right]$, $\tilde{R} = I_N \otimes R$, $\tilde{S} = \left[\begin{smallmatrix} I_N\otimes S \\ 0 \end{smallmatrix}\right]$, and $\tilde{F}$, $\tilde{G}$ are given in the appendix. In [8] this result was improved in the case of $\Psi(p,\cdot)$ being strictly convex quadratic. Translating the latter result to the present setting, $L'_\Psi = \|\begin{bmatrix}\tilde{F} & \tilde{G}\end{bmatrix}\tilde{H}^{-1/2}\|^2$ is smaller than L_Ψ unless $\tilde{H}$ is a positive multiple of the identity matrix.

However, in the present work it is assumed that $\left[\begin{smallmatrix} Q & S' \\ S & R \end{smallmatrix}\right]$, and Q_N are merely positive semidefinite as it often happens in MPC applications, when for example the system's output is weighted in the stage cost. Since for a given $p \in \mathbb{R}^{n_x}$, Problem (7) is an equality constrained convex quadratic program, its value function $\Psi(p,\cdot)$ is convex quadratic, therefore the spectral norm of its Hessian matrix is the tightest Lipschitz constant for $\nabla_y\Psi(p,\cdot)$. The next proposition gives a formula for the Hessian of $\Psi(p,\cdot)$, using Proposition 1.

Proposition 2: Consider problem (7). Then $x = \tilde{X}y + \tilde{t}(p)$, $u = \tilde{U}y + \tilde{v}(p)$, $\nabla_y^2\Psi(p,y) = \tilde{F}\tilde{X} + \tilde{G}\tilde{U}$, where

$$\begin{aligned} \tilde{X} &= \tilde{L}'\tilde{B}\tilde{E}, \qquad \tilde{t}(p) = \tilde{L}'(\tilde{B}\tilde{d} + \tilde{f}(p)) \\ \tilde{U} &= \tilde{K}\tilde{X} + \tilde{E},\ \tilde{v}(p) = \tilde{K}\tilde{t}(p) + \tilde{d}, \end{aligned}$$

$\tilde{E} = \tilde{D} + \tilde{M}\tilde{L}\tilde{C}$, and the matrices involved are given in the appendix. Therefore $L_\Psi = \|\tilde{F}\tilde{X} + \tilde{G}\tilde{U}\|$ is the tightest Lipschitz constant for $\Psi(p,\cdot)$.

Based on Proposition 2, matrices $\tilde{U}$, $\tilde{X}$, can be calculated efficiently by utilizing the upper triangularity of $\tilde{L}$, without requiring the factorization of a large matrix. Having calculated matrices $\tilde{X}$, $\tilde{U}$ one can compute the spectral norm of the Hessian, i.e., $L_\Psi = \|\tilde{F}\tilde{X} + \tilde{G}\tilde{U}\|$. Alternatively, for LPV systems where the calculation of the spectral norm is not affordable in real-time, one can compute an upper bound on L_Ψ, such as the Frobenius norm or the induced 1-norm, or perform a backtracking line search during the course of GPAD [13], [14].

VII. Certification of complexity

Certifying computational complexity amounts to finding a bound on the number of iterations required by

GPAD to solve $\mathbb{P}(p)$ (cf. Eq. (4), equivalently Eq. (1)) that is independent on p, as defined below.

Definition 2: Given $\mathcal{P} \subset \operatorname{dom}\mathcal{S}$, where $\mathcal{S}(p) = \{z \in \mathcal{Z}(p) | g(z) \leq 0\}$, $\nu^\star$ is called a *uniform iteration bound* (UIB) for GPAD on $\mathcal{P}$ if for every $p \in \mathcal{P}$ and every $\nu \geq \nu^\star$, $\bar{z}_{(\nu)}$ is an $(\varepsilon_V, \varepsilon_g)$-optimal solution for $\mathbb{P}(p)$.

For simplicity we will assume that GPAD is started from $y_{(0)} = y_{(-1)} = 0$. As it is clear from Theorems 2 and 3, obtaining such a UIB requires determining a uniform bound on the norm of some dual optimal solution, defined below.

Definition 3: For a $\mathcal{P} \subset \operatorname{dom}\mathcal{S}$, we say that $\Delta_y(\mathcal{P})$ is a *uniform dual bound (UDB) for* $\mathbb{P}(p)$ *on* $\mathcal{P}$ if for any $p \in \mathcal{P}$ there exists a $y^\star(p) \in \mathcal{Y}^\star(p)$ such that

$$\|y^\star(p)\| \leq \Delta_y(\mathcal{P}) < \infty. \tag{24}$$

The calculation of a UDB immediately leads to the establishment of a UIB as it is seen by the following corollary, which is a direct consequence of Theorems 2 and 3.

Corollary 2: Let $\mathcal{P} \subset \operatorname{dom}\mathcal{S}$ and $\Delta_y(\mathcal{P})$ be a UDB on $\mathcal{P}$. Then

$$\nu^\star = \left\lceil \sqrt{2L_\Psi \Delta_y(\mathcal{P})} \max\left\{ \sqrt{\tfrac{\Delta_y(\mathcal{P})}{\varepsilon_V}}, \tfrac{2}{\sqrt{\varepsilon_g}} \right\} \right\rceil - 2,$$

is a UIB on $\mathcal{P}$.

Since Theorems 2 and 3 are valid for any $y^\star(p) \in \mathcal{Y}^\star(p)$, the tightest UDB on $\operatorname{dom}\mathcal{S}$ is $\max_{p \in \operatorname{dom}\mathcal{S}} \min_{y^\star(p) \in \mathcal{Y}^\star(p)} \|y^\star(p)\|$. In principle, one can determine $\mathcal{Y}^\star(p)$ explicitly (by solving the parametric optimization problem $\mathbb{D}(p)$ using the algorithm of [17]) and then calculate the minimum norm selection. However, this procedure would vanish the use of on-line optimization, in favor of the explicit solution. Finding an upper bound for $\max_{y^\star(p) \in \mathcal{Y}^\star(p)} \|y^\star(p)\|$ on some compact subset of $\operatorname{int}(\operatorname{dom}\mathcal{S})$ (assuming that $\operatorname{dom}\mathcal{S}$ has nonempty interior) is a much easier problem. The need to confine the search for such a bound only on a $\mathcal{P} \subset \operatorname{int}(\operatorname{dom}\mathcal{S})$ is dictated by the following lemma.

Definition 4: We say that $\mathbb{P}(\bar{p})$ satisfies the *Slater condition* (at $\bar{p} \in \mathbb{R}^d$) if there exists a $\bar{z} \in \mathcal{Z}(\bar{p})$ such that $g(z) < 0$. We call any such vector $\bar{z}$ a *Slater vector* for $\mathbb{P}(\bar{p})$.

Lemma 1: Consider $\mathbb{P}(p)$. If $\operatorname{int}(\operatorname{dom} S) \neq \emptyset$, then $\mathbb{P}(\bar{p})$ satisfies the Slater condition if and only if $\bar{p} \in \operatorname{int}(\operatorname{dom} S)$, in which case $\mathcal{Y}^\star(\bar{p})$ is bounded and

$$\|y^\star(\bar{p})\| \leq \gamma^{-1}(V(\bar{p}, \bar{z}) - V^\star(\bar{p})), \tag{25}$$

for all $y^\star(\bar{p}) \in \mathcal{Y}^\star(\bar{p})$, where $\gamma = \min_{i \in \mathbb{N}_{[1,m]}}\{-g_i(\bar{z})\}$ and $\bar{z} \in \mathbb{R}^n$ is a Slater point for $\mathbb{P}(\bar{p})$.

Remark 1: For $\bar{p} \in \operatorname{int}(\operatorname{dom}\mathcal{S})$, the tightest possible bound of the form (25) can be obtained by solving $\min_{z,\gamma}\{\gamma^{-1}V(\bar{p}, z) | z \in \mathcal{Z}(p), g(z) + \gamma \leq 0, \gamma \geq 0\}$. Although this problem is not convex, it is equivalent to $\Gamma(\alpha) = 0$, where

$$\Gamma(\alpha) = \min_{z \in \mathcal{Z}(p), \gamma \geq 0} \{V(\bar{p}, z) - \alpha\gamma | g(z) + \gamma \leq 0\}. \tag{26}$$

Notice that Γ is a univariate function whose root can be found by employing a generalized Newton method with respect to α, where at each iteration we solve (26), which is a convex QP [18].

Next, two practical ways of calculating a UDB for $\mathbb{P}(p)$ on a subset of $\operatorname{int}(\operatorname{dom}\mathcal{S}) \neq \emptyset$, are presented.

Proposition 3: Let $P_S = \{p^s\}_{s \in \mathbb{N}_{[1,S]}}$, with $p^s \in \operatorname{int}(\operatorname{dom}\mathcal{S})$, and

$$\gamma = \min\{-g_i(z^s) | s \in \mathbb{N}_{[1,S]}, i \in \mathbb{N}_{[1,m]}\}, \tag{27a}$$

$$\beta = \max\{V(p^s, z^s) | s \in \mathbb{N}_{[1,S]}\}, \tag{27b}$$

where z^s is a Slater vector for $\mathbb{P}(p^s)$, $s \in \mathbb{N}_{[1,S]}$. If $V^\star \geq 0$, then $\Delta_y(\mathcal{P}_S) = \gamma^{-1}\beta$ is a UDB for $\mathbb{P}(p)$ on $\mathcal{P}_S \triangleq \operatorname{conv} P_S$.

A second way of deriving a UDB is described next.

Proposition 4: Let $\operatorname{int}(\operatorname{dom}\mathcal{S}) \neq \emptyset$, $\varepsilon > 0$ and $\mathcal{P}_\varepsilon = \{p \in \mathbb{R}^{n_x} | \exists z \in \mathcal{Z}(p), g(z) + \epsilon \leq 0\}$. Then

$$\Delta_y(\mathcal{P}_\varepsilon) = \max_{p,y}\{\textstyle\sum_{i=1}^m y_i | y \in \mathcal{Y}^\star(p), p \in \mathcal{P}_\varepsilon\} \tag{28}$$

is a UDB for $\mathbb{P}(p)$ on $\mathcal{P}_\varepsilon$.

Remark 2: By writing down the optimality conditions for $\mathbb{P}(p)$, problem (28) can be formulated as a *Linear Program with Linear Complementarity Constraints* (LPCC), which in turn can be transformed into mixed-integer linear inequalities by adopting the so-called "big-M" technique (see e.g. [19, Prop. 2]), and using bounds obtained by e.g., Proposition 3. Obviously, bound (28) is tighter than the one of Proposition 3, and by choosing an arbitrarily small ε it can be made valid for $\mathcal{P} = \operatorname{int}(\operatorname{dom}\mathcal{S})$.

VIII. Simulations

Consider a two dimensional system with matrices $A = \left[\begin{smallmatrix}1 & 1\\0 & 1\end{smallmatrix}\right]$, $B = \left[\begin{smallmatrix}1\\0.5\end{smallmatrix}\right]$, and constraints $-1 \leq u \leq 1$, $-\left[\begin{smallmatrix}2\\5\end{smallmatrix}\right] \leq x \leq \left[\begin{smallmatrix}2\\5\end{smallmatrix}\right]$. The weight matrices are $Q = I_2$, $R = 0.1$, while S, q, r, q_N, f are equal to zero. The terminal weight Q_N is chosen as the solution of the infinite-horizon unconstrained LQR problem, and F_N, g_N represent the maximal positively invariant set of the system in closed-loop with the LQR feedback law. The certification analysis will be carried out for GPAD and Problem (1) with prediction horizon ranging from $N = 5$ to $N = 15$ with step 2. For each N, we apply the following steps: (i) calculate the Lipchitz constant using Proposition 2; (ii) compute the UDB according to Proposition 4 by solving problem (28) ($\varepsilon = 0.1$); (iii) compute the UDB according to Proposition 3, with the collection of points being the vertices of $\mathcal{P}_\varepsilon$ (to make the

TABLE I: Complexity certification analysis

N	Uniform iteration bound			GPAD	
	Prop. 3	Prop. 4	Sampled	Cor. 1	Alg. 2
5	6188	1692	1282	570	164
7	8716	1960	1392	607	166
9	8758	1972	1398	610	167
11	8780	2012	1402	611	167
13	8794	2010	1404	612	167
15	8802	2006	1406	613	167

comparison with Proposition 4 fair); (iv) compute the UIBs according to Corollary 2; (v) form a uniform grid on $\mathcal{P}_\varepsilon$ (gridding each axis with 10^3 points) and for each point of the grid compute an $(\varepsilon_V, \varepsilon_g)$-solution ($\varepsilon_V = 10^{-2}$, $\varepsilon_g = 10^{-3}$) stopping GPAD according with the termination criterion of Corollary 1 and of Algorithm 2; (vi) for validation purposes, for each point of the grid we compute the optimal solution of (1) using CPLEX, and then the minimum norm dual optimal solution. The maximum of the norm of the dual optimal solutions along the grid serves as a benchmark to compare against our bounds. Notice that this is just a *lower bound* to the tightest UDB.

The results of the analysis are summarized in Table I, showing that Proposition 4 leads to much tighter bounds than Proposition 3, and very close to the ones obtained by sampling (40% larger), and obtained by GPAD with the termination criterion of Corollary 1. Finally, GPAD with the termination criterion of Algorithm 2 greatly outperforms the one with termination criterion given by Corollary 1.

In terms of speed of execution of the GPAD algorithm, a comparison with other state-of-the-art active set and interior point solvers has been reported in [20] and [9], respectively, showing comparable (and often superior) performance.

References

[1] A. Bemporad, "Model-based predictive control design: New trends and tools," in *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, CA, 2006, pp. 6678–6683.

[2] H. Ferreau, H. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 816–830, 2008.

[3] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optimization and Engineering*, pp. 1–27, 2010.

[4] P. Patrinos, P. Sopasakis, and H. Sarimveis, "A global piecewise smooth Newton method for fast large-scale model predictive control," *Automatica*, vol. 47, no. 9, pp. 2016–2022, 2011.

[5] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[6] Y. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, vol. 27, no. 2, pp. 372–376, 1983.

[7] S. Richter, C. Jones, and M. Morari, "Real-time input-constrained MPC using fast gradient methods," in *Proc. 48th IEEE Conf. on Decision and Control*, 2009, pp. 7387–7393.

[8] S. Richter, M. Morari, and C. Jones, "Towards computational complexity certification for constrained MPC based on lagrange relaxation and the fast gradient method," in *Proc. 50th IEEE Conf. on Decision and Control and European Control Conf.*, Orlando, USA, 2011, pp. 5223–5229.

[9] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control (extended version)," IMT Lucca, Italy, Internal report, 2012, submitted for publication.

[10] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Athena Scientific, 1999.

[11] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers, 2004.

[12] D. Bertsekas, *Convex optimization theory*. Athena Scientific, 2009.

[13] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," Department of Mathematics, University of Washington, Tech. Rep., 2008.

[14] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[15] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[16] I. Necoara and J. Suykens, "Application of a smoothing technique to decomposition in convex optimization," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2674–2679, 2008.

[17] P. Patrinos and H. Sarimveis, "A new algorithm for solving convex parametric quadratic programs based on graphical derivatives of solution mappings," *Automatica*, vol. 46, no. 9, pp. 1405–1418, 2010.

[18] A. Zhang and S. Hayashi, "Celis-Dennis-Tapia based approach to quadratic fractional programming problems with two quadratic constraints," *Numerical Algebra, Control and Optimization*, vol. 1, no. 1, pp. 83–98, 2011.

[19] W. Heemels, B. D. Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," *Automatica*, vol. 37, no. 7, pp. 1085–1091, Jul. 2001.

[20] A. Bemporad and P. Patrinos, "Simple and certifiable quadratic programming algorithms for embedded linear model predictive control," in *Proc. 4th IFAC Nonlinear Model Predictive Control Conference*, F. A. M. Lazar, Ed., 2012, pp. 14–20.

Appendix

$$\tilde{F} = \begin{bmatrix} I_N \otimes F & 0 \\ 0 & F_N \end{bmatrix}, \tilde{G} = \begin{bmatrix} I_N \otimes G \\ 0 \end{bmatrix},$$

$$\tilde{C} = \begin{bmatrix} 0 & 0 \\ 0 & \oplus_{k=1}^{N} C_k \end{bmatrix}, \tilde{M} = \begin{bmatrix} 0 & \oplus_{k=0}^{N-1} M_k \end{bmatrix},$$

$$\tilde{L} = \begin{bmatrix} I & L_0 & L_0^1 & \cdots & L_0^{N-1} & L_0^{N-1} \\ 0 & I & L_1 & \cdots & L_1^{N-2} & L_1^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & I & L_{N-1}^{N-1} \\ 0 & 0 & 0 & \cdots & 0 & I \end{bmatrix}$$

$$\tilde{d} = d + \tilde{L}s, d = \begin{bmatrix} d_0' & \cdots & d_{N-1}' \end{bmatrix}', s = \begin{bmatrix} 0 & s_1' & \cdots & s_N' \end{bmatrix}'$$

$$\tilde{D} = \begin{bmatrix} \oplus_{k=0}^{N-1} D_k & 0 \end{bmatrix}, \ \tilde{K} = \begin{bmatrix} \oplus_{k=0}^{N-1} K_k & 0 \end{bmatrix}$$

$$\tilde{A} = \begin{bmatrix} -I & 0 & \cdots & 0 & 0 \\ A & -I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & A & -I \end{bmatrix}, \tilde{B} = \begin{bmatrix} 0 \\ I_N \otimes B \end{bmatrix}, \tilde{f}(p) = \begin{bmatrix} p \\ f \\ \vdots \\ f \end{bmatrix}$$