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

Panagiotis Patrinos and Alberto Bemporad, *Fellow, IEEE*

***Abstract*—This paper proposes a dual fast gradient-projection method for solving quadratic programming problems that arise in model predictive control of linear systems subject to general polyhedral constraints on inputs and states. The proposed algorithm is well suited 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 tightly estimated; and 3) the computational cost per iteration increases only linearly with the prediction horizon.**

***Index Terms*—Computational methods, optimization algorithms, predictive control for linear systems.**

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]–[3]. The key enabler for the spread of MPC in industry is the availability of algorithms that can solve the quadratic program (QP) associated with MPC online at each sampling step. This means the availability of embedded optimization algorithms for QPs that: 1) can provide a solution within the available sampling interval (e.g., 10 ms in a typical automotive application) on relatively simple control hardware (e.g., a microcontroller or a field-programmable gate array); 2) require a small memory footprint to store the data defining the optimization problem and the code implementing the optimization algorithm itself; 3) lead to a control code that is simple enough to be software-certifiable, especially in safety-critical applications; and 4) have worst-case execution time that is well predictable, in order to satisfy hard real-time system requirements.

During the last years various researchers have spent considerable efforts to develop algorithms that address the above requirements. To date, many good algorithms and packages for QP are available to solve linear MPC problems, mainly active-set methods [4]–[6], interior-point methods [7]–[9], and Newton methods [10].

A different approach to meet the above requirements was taken in [11], where the authors proposed the use of multiparametric quadratic programming to precompute the optimal solution as an explicit piecewise affine function of the state vector. Nonetheless, the approach is limited to relatively small problems (typically: one or two command inputs,short prediction horizons, up to ten states) and to linear time-invariant (LTI) systems, so the quest for adequate QP algorithms to be embedded in control applications for online execution is continuing.

Within the MPC community, fast gradient-projection methods [12], [13] were also proposed very recently in [14]–[18]. In [15], [16] a first step was made towards computational complexity certification for the method of [12] when applied to MPC. The fast gradient-projection method is applied to the primal problem, so the results are limited only to input-constrained MPC problems with simple constraint sets (e.g., a box). In [17] the fast gradient method is applied to the dual of the MPC problem, where the equality constraints corresponding to the state equations are relaxed. This framework can only handle bound input-state constraints with diagonal weight matrices and ellipsoidal constraints on the terminal state, with the Hessian matrix defining the ellipsoid being equal to that of the terminal cost. Complexity certification is provided only for the dual cost (whereas in MPC the main concern is on primal cost and feasibility of primal variables) and the estimated bounds are not tight in general. In [19] a combination of the augmented Lagrangian with the fast gradient-projection method is proposed for linear time-invariant MPC with box constraints. The algorithm involves an outer and inner loop and has no complexity certification.

In this paper we propose an algorithm based on the fast gradient-projection method of [12] for linear MPC problems with *general polyhedral constraints on inputs and states*. Specifically, the fast gradient-projection method is applied to the dual problem resulting by relaxing the inequality constraints. Global convergence rates of $O(1/\nu^2)$ (where ν is the iteration counter) are provided not only for dual optimality but also for primal optimality, primal feasibility, and distance from the primal optimizer, which are all that matter in MPC applications. In addition, practical termination criteria are provided that guarantee a solution of prespecified accuracy. Furthermore, an efficient way of calculating the gradient of the dual function with a complexity that increases only linearly with the prediction horizon is described. Finally, some practical ways for determining the worst-case number of iterations to achieve a certain accuracy are provided for MPC problems.

Manuscript received August 05, 2012; revised February 12, 2013 and June 27, 2013; accepted June 27, 2013. Date of publication July 30, 2013; date of current version December 19, 2013. This work was supported by the HYCON2 Network of Excellence "Highly-Complex and Networked Control Systems" under Grant Agreement 257462 and by the "E-PRICE: Price-based Control of Electrical Power Systems" under Grant Agreement 249096. Preliminary versions and parts of this work were presented in [20] and [21]. Recommended by Associate Editor A. Ferrara.

The authors are with IMT Institute for Advanced Studies Lucca, 55100 Lucca, Italy (e-mail: panagiotis.patrinos@imtlucca.it; alberto.bemporad@imtlucca.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2013.2275667

II. Notation

Let $\mathbb{R}$, $\mathbb{N}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$, $\mathbb{S}^n_+$, $\mathbb{S}^n_{++}$ denote the sets of real numbers, non-negative integers, column real vectors of length n, real matrices of size m by n, 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 $\{k_1, k_1+1, \ldots, k_2\}$ 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 ith coordinate is $\max\{z_i, 0\}$. For $C \subseteq \mathbb{R}^d \times \mathbb{R}^n$, the projection of C onto the p-axis, is denoted by $\mathsf{P}_p C = \{p \in \mathbb{R}^d | \exists z \in \mathbb{R}^n \text{ s.t. } (p,z) \in C\}$. For a vector $z \in \mathbb{R}^n$, $\|z\|$ denotes its Euclidean norm, while if $A \in \mathbb{R}^{m\times n}$, $\|A\|$ denotes its spectral norm (unless otherwise stated). For $A \in \mathbb{R}^{m\times n}$, $A_{i\cdot}$ denotes the ith row of A. The n-dimensional vector whose entries are all equal to one is denoted by $\mathbf{1}$ (the dimension should be clear from the context). The Kronecker product of $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{p\times q}$ is denoted by $A \otimes B$. If $V : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $\|\nabla V(z_1) - \nabla V(z_2)\| \leq L_V \|z_1 - z_2\|$ for all $z_1, z_2 \in \mathbb{R}^n$ then we say that V is L_V-smooth. If there exists a $\mu_V > 0$ such that $(\nabla V(z_1) - \nabla V(z_2))'(z_1 - z_2) \geq \mu_V \|z_1 - z_2\|^2$ for every z_1, z_2 belonging to some convex set C then we say that V is strongly convex on C with convexity parameter μ_V. For a set $C \subseteq \mathbb{R}^n$, $\operatorname{int} C$, $\operatorname{rint} C$, $\operatorname{bdr} C$ denote its interior, its relative interior and its boundary, respectively. The graph of a set-valued mapping $\mathcal{S} : \mathbb{R}^d \rightrightarrows \mathbb{R}^n$, is the set $\operatorname{gph} \mathcal{S} = \{(p,z) | z \in \mathcal{S}(p)\}$, its domain is $\operatorname{dom} \mathcal{S} = \{p | \mathcal{S}(p) \neq \emptyset\} = \mathsf{P}_p(\operatorname{gph} S)$, while its range is $\operatorname{rge} \mathcal{S} = \{z | \exists p \text{ with } z \in \mathcal{S}(p)\}$.

III. Basic Setup

Consider an MPC formulation based on the following constrained finite-horizon linear-quadratic optimal control problem:

$$
\begin{aligned}
V^\star(p) = \min \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N) && \text{(1a)}\\
\text{s.t.} \quad & x_0 = p && \text{(1b)}\\
& x_{k+1} = Ax_k + Bu_k + f,\ k \in \mathbb{N}_{N-1} && \text{(1c)}\\
& Fx_k + Gu_k \leq c, \qquad k \in \mathbb{N}_{N-1} && \text{(1d)}\\
& F_N x_N \leq c_N && \text{(1e)}
\end{aligned}
$$

where given a vector $p \in \mathbb{R}^{n_x}$ (the current state vector) the goal is to compute a state-input sequence $\{x_0, \ldots, x_N, u_0, \ldots, u_{N-1}\}$, $(x_k \in \mathbb{R}^{n_x}, u_k \in \mathbb{R}^{n_u})$ for the LTI model (1c), such that it minimizes the finite-horizon cost (1a) over the prediction horizon of length N, while satisfying initial condition (1b), linear state-input constraints (1d) $(c \in \mathbb{R}^{m_s})$ and linear terminal constraints (1e) $(c_N \in \mathbb{R}^{m_N})$. It is assumed that the stage cost is

$$
\ell(x,u) = \frac{1}{2}\begin{bmatrix} x \\ u \end{bmatrix}' \begin{bmatrix} Q & S' \\ S & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} + \begin{bmatrix} q \\ r \end{bmatrix}' \begin{bmatrix} x \\ u \end{bmatrix} \qquad (2)
$$

and that the terminal cost is

$$
V_f(x) = \frac{1}{2} x' Q_N x + q_N' x \qquad (3)
$$

where $R \in \mathbb{S}^{n_u}_{++}$, $\begin{bmatrix} Q & S' \\ S & R \end{bmatrix} \in \mathbb{S}^{n_x+n_u}_+$, 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, let $V(z) = \sum_{k=0}^{N-1} \ell(x_k, u_k) + V_f(x_N)$, $\mathcal{Z}(p) = \{z \in \mathbb{R}^n | x_0 = p, x_{k+1} = Ax_k + Bu_k + f, k \in \mathbb{N}_{N-1}\}$ and $g : \mathbb{R}^n \to \mathbb{R}^m$ $(m = Nm_s + m_f)$ be 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

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

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

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

where the dual function is

$$
\Psi(p,y) = \min_{z \in \mathcal{Z}(p)} \mathcal{L}(z,y) \qquad (6)
$$

and $\mathcal{L}(z,y) = V(z) + y'g(z)$ is the Lagrangian function for (4). For $p \in \mathbb{R}^{n_x}$ let $\mathcal{Z}^\star(p)$, $\mathcal{Y}^\star(p)$ denote the set of optimal solutions of (4) and (5), respectively.

Since (4) is a convex quadratic program, strong duality holds as long as (4) is feasible (without the need of any constraint qualification). Therefore $V^\star(p) = \Psi^\star(p)$, and in principle one could solve $\mathbb{D}(p)$ to obtain a dual optimal vector $y^\star(p) \in \mathcal{Y}^\star(p)$ and then calculate a primal optimal vector $z^\star(p) \in \mathcal{Z}^\star(p)$ by solving $\min_{z \in \mathcal{Z}(p)} \mathcal{L}(z, y^\star)$.

IV. Accelerated Dual Gradient Projection

This section describes 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 $\varepsilon_V, \varepsilon_g$. 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

$$
\begin{aligned}
V(z) - V^\star &\leq \varepsilon_V && \text{(7a)}\\
\left\|[g(z)]_+\right\|_\infty &\leq \varepsilon_g. && \text{(7b)}
\end{aligned}
$$

Equation (7a) bounds the distance of the corresponding primal cost from the optimal value, while (7b) bounds the maximum primal infeasibility for the primal suboptimal solution. Note that although in most dual methods (e.g., [17]) the goal is to find an approximate *dual* solution (i.e., $\Psi^\star - \Psi(p,y) \leq \varepsilon_\Psi$), our aim is to find an approximate *primal* solution in the sense of Definition 1. This is very important in MPC applications where the goal is to compute the optimal input sequence for (1).

To simplify notation, we omit the dependence on the parameter p appearing in (4), (5), and (6). The results presented in this

section are valid under less stringent assumptions than the ones satisfied by (1), regarding V, g, $\mathcal{Z}$ and Ψ. Specifically, we assume that $V : \mathbb{R}^n \to \mathbb{R}$ is strongly convex and continuous over $\mathcal{Z}$, $g : \mathbb{R}^n \to \mathbb{R}^m$ is affine, and $\mathcal{Z} \subseteq \mathbb{R}^n$ is closed, convex. Then Ψ is L_Ψ-smooth and its gradient is given by $\nabla_y \Psi(y) = g(z^y)$ with $z^y = \arg\min_{z\in\mathcal{Z}} \mathcal{L}(z, y)$, e.g., see [23]. Furthermore, we assume that there exists a $z \in \mathcal{Z}$ with $g(z) < 0$, or that $\mathcal{Z}$ is polyhedral and there exists a $z \in \mathcal{Z}$ with $g(z) \leq 0$. Then strong duality holds [24].

GPAD (summarized in Algorithm 1) is the first fast gradient-projection method of Nesterov [12] (see also [13], [25, Sec. 6.9,], [26, Alg. 2], and [27]) applied to the concave maximization problem (5).

Algorithm 1: Accelerated Dual Gradient-Projection

Input: $y_{(0)} = y_{(-1)} \in \mathbb{R}^m_+$. $\theta_0 = \theta_{-1} = 1$. $\nu \leftarrow 0$

1 $w_{(\nu)} = y_{(\nu)} + \theta_\nu(\theta_{\nu-1}^{-1} - 1)(y_{(\nu)} - y_{(\nu-1)})$

2 $z_{(\nu)} = \arg\min_{z\in\mathcal{Z}} \mathcal{L}(z, w_{(\nu)})$

3 $y_{(\nu+1)} = [w_{(\nu)} + (1/L_\Psi)g(z_{(\nu)})]_+$

4 $\theta_{\nu+1} = (\sqrt{\theta_\nu^4 + 4\theta_\nu^2} - \theta_\nu^2)/2$. Set $\nu \leftarrow \nu + 1$ and go to Step 1.

When Algorithm 1 is applied to solve (1), a Lipschitz constant for $\nabla\Psi$ can be computed by forming explicitly the Hessian of Ψ as in [21] and calculating its spectral norm. We remark that Algorithm 1 can also be applied to solve problem (1), in the case where the dynamics (1c) are linear time-varying (LTV). For LTV 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 (see, e.g., [26] and [27]).

Remark 1: Convergence of Algorithm 1 can be shown for any stepsize satisfying

$$\frac{1-\theta_{\nu+1}}{\theta_{\nu+1}^2} \leq \frac{1}{\theta_\nu^2} \tag{8}$$

e.g., $\theta_\nu = 2/(\nu+2)$ (see [26]). The stepsize in Step 4 satisfies (8) as an equality. It also satisfies the inequality $\theta_\nu \leq (2/(\nu+2))$. More importantly, it possesses a crucial property which will be heavily exploited later [26], i.e.,

$$\vartheta_\nu \triangleq \sum_{i=0}^{\nu} \theta_i^{-1} = \frac{1}{\theta_\nu^2}. \tag{9}$$

Denote by $\ell_\Psi(\cdot; y) : \mathbb{R}^m \to \mathbb{R}$ the first-order Taylor expansion of Ψ at $y \in \mathbb{R}^m$, i.e., $\ell_\Psi(v; y) = \Psi(y) + \nabla\Psi(y)'(v-y)$, and $\Delta_\Psi(v; y) = \ell_\Psi(v; y) - \Psi(v)$. Since $\Psi : \mathbb{R}^m \to \mathbb{R}$ is concave, with Lipschitz continuous gradient, one has $0 \leq \Delta_\Psi(v; y) \leq (L_\Psi/2)\|v - y\|^2$ [13, Th. 2.1.5]. Let

$$v_{(\nu)} = y_{(\nu-1)} + \theta_{\nu-1}^{-1}\left(y_{(\nu)} - y_{(\nu-1)}\right) \tag{10a}$$
$$= y_{(\nu)} + \theta_\nu^{-1}\left(w_{(\nu)} - y_{(\nu)}\right) \tag{10b}$$

and notice that $v_{(0)} = y_{(0)}$, since $y_{(0)} = y_{(-1)}$. The following lemma will be instrumental for showing global convergence rates for the primal problem (4).

Lemma 2 ([26, Proof of Cor. 2]): Let $\{y_{(\nu)}, w_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1. For any $\nu \in \mathbb{N}_+$ and $y \geq 0$,

$$\theta_\nu^{-2}\left(\Psi(y) - \Psi\left(y_{(\nu+1)}\right)\right) + \sum_{i=0}^{\nu} \theta_i^{-1}\Delta_\Psi\left(y; w_{(i)}\right) + \frac{L_\Psi}{2}\left\|y - v_{(\nu+1)}\right\|^2 \leq \frac{L_\Psi}{2}\left\|y - y_{(0)}\right\|^2. \tag{11}$$

The main consequence of Lemma 2 is the $O(1/\nu^2)$ global convergence rate of the dual cost to the optimal value, stated in the following corollary.

Corollary 3 ([26, Cor. 2]): Let $\{y_{(\nu)}, w_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1. Then for any $\nu \in \mathbb{N}_+$

$$\Psi(y^\star) - \Psi\left(y_{(\nu+1)}\right) \leq \frac{2L_\Psi}{(\nu+2)^2}\left\|y_{(0)} - y^\star\right\|^2. \tag{12}$$

In Corollary 3 and in what follows, $y^\star$ is any element of the set $\mathcal{Y}^\star$ of dual optimal solutions.

A. Primal Infeasibility Bound

Algorithm 1 solves the dual problem (5). However, in applications such as MPC were the task is to compute an optimal input sequence, the main concern is to determine properties of the algorithm regarding the primal problem (4). In this subsection we will provide bounds for the convergence rate of the *averaged primal sequence*

$$\bar{z}_{(\nu)} = \vartheta_\nu^{-1}\sum_{i=0}^{\nu} \theta_i^{-1} z_{(i)} = (1-\theta_\nu)\bar{z}_{(\nu-1)} + \theta_\nu z_{(\nu)} \tag{13}$$

where $\bar{z}_{(-1)} = 0$. Convergence results for the averaged primal sequence (13) have been reported in [23, Th. 3], for Nesterov's third accelerated gradient scheme which uses two projections and a weighted sum of all the previous gradients, and in [26, Cor. 2] for a method similar to the one presented in this paper. A similar approach for Nesterov's third method coupled with the smoothing technique of [23] and dual decomposition for distributed optimization was followed by [14]. However all these results concern saddle problems with compact dual and/or primal constraint sets. In contrast, we do not require any boundedness assumption on $\mathcal{Z}$ or the dual optimal set $\mathcal{Y}^\star$.

The next theorem provides bounds on the maximum primal infeasibility for the averaged primal sequence (13).

Theorem 4: Let $\{y_{(\nu)}, w_{(\nu)}, z_{(\nu)}, \theta_\nu\}$ be generated by algorithm 1. Then for any $\nu \in \mathbb{N}_+$

$$\left\|\left[g\left(\bar{z}_{(\nu)}\right)\right]_+\right\|_\infty \leq \frac{8L_\Psi}{(\nu+2)^2}\left\|y_{(0)} - y^\star\right\|. \tag{14}$$

Proof: For any $\nu \in \mathbb{N}_+$ we have $w_{(\nu)} + (1/L_\Psi)g(z_{(\nu)}) \leq [w_{(\nu)} + (1/L_\Psi)g(z_{(\nu)})]_+ = y_{(\nu+1)}$. Multiplying by θ_ν^{-1} and rearranging terms $\theta_\nu^{-1}g(z_{(\nu)}) \leq L_\Psi\theta_\nu^{-1}(y_{(\nu+1)} - w_{(\nu)}) = L_\Psi(v_{(\nu+1)} - v_{(\nu)})$, where the equality follows by subtracting

(10b) from (10a) and the latter is used to express $v_{(\nu+1)}$. Since $v_{(0)} = y_{(0)}$, this implies

$$\sum_{i=0}^{\nu} \theta_i^{-1} g\left(z_{(i)}\right) \leq L_\Psi \left(v_{(\nu+1)} - y_{(0)}\right).$$

Multiplying by ϑ_ν^{-1} and using the convexity of each component of g

$$g\left(\bar{z}_{(\nu)}\right) \leq \vartheta_\nu^{-1} \sum_{i=0}^{\nu} \theta_i^{-1} g\left(z_{(i)}\right) \leq \vartheta_\nu^{-1} L_\Psi \left(v_{(\nu+1)} - y_{(0)}\right).$$

From (9) it holds $\vartheta_\nu^{-1} = \theta_\nu^2$. Hence, using the fact that $a \leq b \Rightarrow [a]_+ \leq [b]_+$ and that $\theta_\nu \leq (2/(\nu+2))$ (see Remark 1), we get

$$\left\| \left[g\left(\bar{z}_{(\nu)}\right)\right]_+ \right\| \leq \frac{4L_\Psi}{(\nu+2)^2} \left\| \left[v_{(\nu+1)} - y_{(0)}\right]_+ \right\|. \tag{15}$$

Using

$$y = y^\star \in \mathcal{Y}^\star$$

in (11) and dropping $\sum_{i=0}^{\nu} \theta_i^{-1} \Delta_\Psi(y^\star; w_{(i)})$ and $\Psi(y^\star) - \Psi(y_{(\nu+1)})$ since they are nonnegative, we obtain $\|y^\star - v_{(\nu+1)}\| \leq \|y^\star - y_{(0)}\|$. Therefore $\|[v_{(\nu+1)} - y_{(0)}]_+\| \leq \|v_{(\nu+1)} - y_{(0)}\| \leq \|(v_{(\nu+1)} - y^\star) - (y_{(0)} - y^\star)\| \leq \|v_{(\nu+1)} - y^\star\| + \|y_{(0)} - y^\star\| \leq 2\|y_{(0)} - y^\star\|$, where the first inequality follows from nonexpansiveness of $[\cdot]_+$, and the third by the triangle inequality. Combining the last inequality with (15) and the fact that $\|v\|_\infty \leq \|v\|$ for any $v \in \mathbb{R}^m$, we obtain (14). ∎

B. Primal Suboptimality Bounds

We will next derive a global convergence rate for the distance of $V(\bar{z}_{(\nu)})$ from $V^\star$. 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 5: Let $\{y_{(\nu)}, w_{(\nu)}, z_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1. Then for any $\nu \in \mathbb{N}_+$

$$\begin{aligned} V\left(\bar{z}_{(\nu)}\right) - V^\star &\leq V\left(\bar{z}_{(\nu)}\right) - \Psi\left(y_{(\nu+1)}\right) \\ &\leq \frac{2L_\Psi}{(\nu+2)^2} \left(\|y^\star\|^2 + \left\|y_{(0)}\right\|^2\right) \qquad (16a) \\ V\left(\bar{z}_{(\nu)}\right) - V^\star &\geq -\frac{8L_\Psi}{(\nu+2)^2} \left\|y_{(0)} - y^\star\right\| \|y^\star\|. \qquad (16b) \end{aligned}$$

Proof: Since $\Psi(w_{(\nu)}) = \mathcal{L}(z_{(\nu)}, w_{(\nu)})$ and $\nabla\Psi(w_{(\nu)}) = g(z_{(\nu)})$, for any $y \geq 0$, the quantity $\Delta_\Psi(y; w_{(\nu)})$ can be expressed as $\Delta_\Psi(y; w_{(\nu)}) = \mathcal{L}(z_{(\nu)}, y) - \Psi(y)$. Now multiplying by θ_ν^{-1} and summing from 0 to ν,

$$\begin{aligned} \sum_{i=0}^{\nu} \theta_i^{-1} \Delta_\Psi\left(y; w_{(i)}\right) &= \sum_{i=0}^{\nu} \theta_i^{-1} \left(\mathcal{L}\left(z_{(i)}, y\right) - \Psi(y)\right) \\ &\geq \vartheta_\nu \left(\mathcal{L}\left(\bar{z}_{(\nu)}, y\right) - \Psi(y)\right) \\ &= \frac{1}{\theta_\nu^2} \left(\mathcal{L}\left(\bar{z}_{(\nu)}, y\right) - \Psi(y)\right) \qquad (17) \end{aligned}$$

where the inequality follows from convexity of $\mathcal{L}(\cdot, y)$ for $y \geq 0$ and the second equality follows from (9). Using (17) in Lemma 2 while dropping $(L_\Psi/2)\|y - v_{(\nu+1)}\|^2$ since it is nonnegative, yields

$$V\left(\bar{z}_{(\nu)}\right) + g\left(\bar{z}_{(\nu)}\right)' y - \Psi\left(y_{(\nu+1)}\right) \leq \theta_\nu^2 \frac{L_\Psi}{2} \left\|y - y_{(0)}\right\|^2. \tag{18}$$

By replacing $y = \bar{y}^\star \geq 0$ in (18), where

$$\bar{y}_i^\star = \begin{cases} y_i^\star, & \text{if } g_i\left(\bar{z}_{(\nu)}\right) \geq 0 \\ 0, & \text{if } g_i\left(\bar{z}_{(\nu)}\right) < 0 \end{cases}$$

and dropping the term $g(\bar{z}_{(\nu)})'\bar{y}^\star$ since it is nonnegative, we obtain

$$V\left(\bar{z}_{(\nu)}\right) - \Psi\left(y_{(\nu+1)}\right) \leq \theta_\nu^2 \frac{L_\Psi}{2} \left\|\bar{y}^\star - y_{(0)}\right\|^2. \tag{19}$$

Now

$$\begin{aligned} \left\|\bar{y}^\star - y_{(0)}\right\|^2 &= \|\bar{y}^\star\|^2 - 2y_{(0)}'\bar{y}^\star + \left\|y_{(0)}\right\|^2 \\ &\leq \|y^\star\|^2 + \left\|y_{(0)}\right\|^2 \qquad (20) \end{aligned}$$

since $\|\bar{y}^\star\| \leq \|y^\star\|$ and $2y_{(0)}'\bar{y}^\star \geq 0$. Therefore using $\theta_\nu \leq 2/(\nu+2)$ and (20) in (19) yields

$$V\left(\bar{z}_{(\nu)}\right) - \Psi\left(y_{(\nu+1)}\right) \leq \frac{2L_\Psi}{(\nu+2)^2} \left(\|y^\star\|^2 + \left\|y_{(0)}\right\|^2\right). \tag{21}$$

Since $\Psi(y_{(\nu+1)}) \leq \Psi^\star = V^\star$, (21) implies (16a). To prove (16b), we notice that $V(\bar{z}_{(\nu)}) = \mathcal{L}(\bar{z}_{(\nu)}, y^\star) - g(\bar{z}_{(\nu)})'y^\star \geq V^\star - g(\bar{z}_{(\nu)})'y^\star$, where the inequality follows from the saddle point inequality $V^\star = \mathcal{L}(z^\star, y^\star) \leq \mathcal{L}(z, y^\star)$ for all $z \in \mathcal{Z}$. Therefore $V(\bar{z}_{(\nu)}) - V^\star \geq -g(\bar{z}_{(\nu)})'y^\star$. Since $y^\star \geq 0$ and $[g(\bar{z}_{(\nu)})]_+ \geq g(\bar{z}_{(\nu)})$, it holds that $-g(\bar{z}_{(\nu)})'y^\star \geq -[g(\bar{z}_{(\nu)})]_+'y^\star \geq -\|[g(\bar{z}_{(\nu)})]_+\|\|y^\star\|$. Therefore $V(\bar{z}_{(\nu)}) - V^\star \geq -\|[g(\bar{z}_{(\nu)})]_+\|\|y^\star\|$. Using (14), we obtain (16b). ∎

Equation (16a) refines the results in [26, Cor. 2], where the latter assumes that the constraint set is bounded. This is not the case for GPAD, where the constraint set of the dual is the nonnegative orthant.

If Algorithm 1 starts from $y_{(0)} = 0$, then the cost $V(\bar{z}_{(\nu)})$ is always lower than $V^\star$, as it is shown below. In that case, one has to worry only about checking feasibility during the iterations.

Corollary 6: Let $\{y_{(\nu)}, w_{(\nu)}, z_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1 with $y_{(0)} = 0$. Then for any $\nu \in \mathbb{N}_+$

$$V\left(\bar{z}_{(\nu)}\right) \leq V^\star. \tag{22}$$

Proof: Simply put $y = 0$ in (18) and use $\Psi(y_{(\nu+1)}) \leq \Psi^\star = V^\star$. ∎

The next corollary gives a bound on the number of iterations needed by GPAD to reach an $(\varepsilon_V, \varepsilon_g)$-optimal solution. Its proof follows directly by Theorems 4 and 5 and Corollary 6.

Corollary 7: Let $\Delta_y^\star \triangleq \min_{y^\star \in \mathcal{Y}^\star} \|y_{(0)} - y^\star\|$. For any

$$\nu \geq \sqrt{2L_\Psi \Delta_y^\star} \max\left\{ \sqrt{\frac{\Delta_y^\star}{\varepsilon_V}}, \frac{2}{\sqrt{\varepsilon_g}} \right\} - 2 \qquad (23)$$

$y_{(\nu)}$ is a $(\varepsilon_V, \varepsilon_g)$-solution for (4). If $y_{(0)} = 0$, for any

$$\nu \geq \sqrt{\frac{8L_\Psi \Delta_y^\star}{\varepsilon_g}} - 2 \qquad (24)$$

$y_{(\nu)}$ is a $(0, \varepsilon_g)$-solution for (4).

According to Corollary 7, when started from $y_{(0)} = 0$, GPAD reaches $(0, \varepsilon)$-optimality in $O(\sqrt{L_\Psi \Delta_y^\star / \epsilon})$ iterations. The complexity estimate to achieve the same level of suboptimality for the dual cost, which is the standard result found in the literature (see, e.g., [12]) is of order $O(\sqrt{L_\Psi/\epsilon}\Delta_y^\star)$. In the case of Problem (1), the only quantity dependent on the initial state p is $\Delta_y^\star$. In fact, for MPC problems $\Delta_y^\star$ can become very large especially when the initial state is close to the boundary of the set of feasible states. When $y_{(0)} = 0$, the bound on the number of iterations becomes less sensitive to large values of $\Delta_y^\star$ (if $\Delta_y^\star > 1$), resulting to tighter iteration bounds.

Remark 2: Recently, a double smoothing technique for computing nearly optimal and primal feasible solutions for infinite-dimensional optimization problems was proposed in [28]. The method uses a regularization to obtain an approximate, differentiable dual function, in the spirit of [23]. Then it uses a second regularization in order to make the dual function strongly convex and applies the fast gradient algorithm of [13]. Translating the result of [28] to the present setting, the double smoothing technique is applied to problems of the form $V^\star = \min_{z \in \mathcal{Z}} \{V(z) | \tilde{A}z \in \mathcal{T}\}$, where $\mathcal{T}$ is closed and convex, and $\mathcal{Z}$ is closed, convex and bounded. It is based on forming the dual problem that results from relaxing the constraint $\tilde{A}z \in \mathcal{T}$. This means that V and $\mathcal{Z}$ must be sufficiently simple so that the dual function can be computed explicitly. In an optimal control setting such as MPC, the constraint $\tilde{A}z \in \mathcal{T}$ models the state equations ($\mathcal{T} = \{\tilde{b}\} \in \mathbb{R}^m$), while the stage cost, terminal cost and state-input constraints must be simple enough for the method to be implementable (see also [16]). Since we assume that the dual function is differentiable, the first smoothing is not needed. Taking this into account, the method of [28] requires $O((1/\sqrt{\varepsilon})\ln(1/\varepsilon))$ iterations to compute a primal vector $\tilde{z} \in \mathcal{Z}$ with $|V(\tilde{z}) - V^\star| \leq 2(1 + 2\sqrt{3})\varepsilon$ and $\|\tilde{A}\tilde{z} - b\| \leq (2\varepsilon/\Delta_y^\star)$, when started from $y_{(0)} = 0$. On the other hand, GPAD dualizes the input-state constraints and not the state equations, therefore it can deal with arbitrary polyhedral sets (for which the projection can be quite involved) as input-state constraints. According to Corollary 7 GPAD requires $O(1/\sqrt{\varepsilon})$ iterations to compute a primal vector $\tilde{z} \in \mathcal{Z}$ with $V(\tilde{z}) \leq V^\star$ and $\|[g(\tilde{z})]_+\|_\infty \leq (4\varepsilon/\Delta_y)$, when started from $y^{(0)} = 0$.

Although the two methods are not directly comparable, it is obvious that the $O(1/\sqrt{\varepsilon})$ rate of GPAD is better than the $O((1/\sqrt{\varepsilon})\ln(1/\varepsilon))$ rate of [28]. Furthermore, the technique of [28] requires knowledge of the dual bound $\Delta_y^\star$. As it will become clear in Section VI, the calculation of a tight dual bound is not trivial for parametric optimization problems, such as those arising in MPC. A loose dual bound can have negative effects on the practical convergence of [28].

C. Convergence Rate of Averaged Primal Sequence

The next theorem proves that if the cost function V is strongly convex over $\mathcal{Z}$, then the squared Euclidean distance of the averaged primal sequence from the unique optimizer, $\|\bar{z}_{(\nu)} - z^\star\|^2$, converges to zero with convergence rate $O(1/\nu^2)$, i.e., with the same convergence rate for primal optimality and primal feasibility.

Theorem 8: Let $\{y_{(\nu)}, w_{(\nu)}, z_{(\nu)}, \theta_\nu\}$ be generated by Algorithm 1. If V is continuously differentiable and strongly convex on $\mathcal{Z}$ with convexity parameter μ_V, then for any $\nu \in \mathbb{N}_+$

$$\left\|\bar{z}_{(\nu)} - z^\star\right\|^2 \leq \frac{4L_\Psi \mu_V^{-1}}{(\nu+2)^2} \left(\|y^\star\|^2 + \left\|y_{(0)}\right\|^2 + 4\left\|y_{(0)} - y^\star\right\| \right).$$

Proof: By the assumptions of the statement, $z^\star$ and $y^\star$ satisfy (see e.g. [24])

$$\nabla_z \mathcal{L}(z^\star, y^\star)'(z - z^\star) \geq 0, \ \forall z \in \mathcal{Z} \qquad (25a)$$

$$y^\star \geq 0, \quad y^{\star\prime} g(z^\star) = 0. \qquad (25b)$$

The function $\mathcal{L}(\cdot, y^\star) : \mathbb{R}^n \to \mathbb{R}$ is strongly convex on $\mathcal{Z}$, with convexity parameter μ_V, as the nonnegative weighted sum of the strongly convex function V with the convex functions g_i, $i \in \mathbb{N}_{[1,m]}$ [13, Lem. 2.1.4]. Therefore, we have

$$\begin{aligned} \mathcal{L}\left(\bar{z}_{(\nu)}, y^\star\right) &\geq \mathcal{L}(z^\star, y^\star) + \nabla_z \mathcal{L}(z^\star, y^\star)' \left(\bar{z}_{(\nu)} - z^\star\right) \\ &\quad + \frac{1}{2}\mu_V \left\|\bar{z}_{(\nu)} - z^\star\right\|^2 \\ &\geq \mathcal{L}(z^\star, y^\star) + \frac{1}{2}\mu_V \left\|\bar{z}_{(\nu)} - z^\star\right\|^2 \end{aligned} \qquad (26)$$

where the first inequality follows from definition of strong convexity [13, Def. 2.1.2] and the second from the fact that $\bar{z}_{(\nu)} \in \mathcal{Z}$ and (25a). Therefore

$$\begin{aligned} \left\|\bar{z}_{(\nu)} - z^\star\right\|^2 &\leq \frac{2}{\mu_V}\left(\mathcal{L}\left(\bar{z}_{(\nu)}, y^\star\right) - \mathcal{L}(z^\star, y^\star)\right) \\ &\leq \frac{2}{\mu_V}\left(V\left(\bar{z}_{(\nu)}\right) - V^\star + y^{\star\prime} g\left(\bar{z}_{(\nu)}\right)\right) \end{aligned}$$

where the first inequality follows from (26) and the second from (25b). The proof of the claim completes by invoking Theorems 4 and 5. ∎

We will next show that V is strongly convex on $\mathcal{Z}$ for the MPC problem (1), therefore the assertion of Theorem 8 is valid for (1) (without affecting the validity of the result we assume that q, r, f are all 0). By performing the transformation $v_k = u_k + R^{-1}S'x_k$, the stage cost becomes $\ell(x_k, v_k) = (1/2)(x_k'(Q - SR^{-1}S')x_k + v_k' R v_k)$, where $Q - SR^{-1}S' \in \mathbb{S}_+^{n_x}$ is the Schur complement of $\begin{bmatrix} Q & S' \\ S & R \end{bmatrix} \in \mathbb{S}_+^{n_x+n_u}$. Thus, without loss of generality we can assume that $S = 0$. In that case, the cost function of (1) is

$V(z)=(1/2)(x'\tilde{Q}x+u'\tilde{R}u)$ and $\mathcal{Z}=\{(x,u)|\tilde{A}x+\tilde{B}u=\tilde{f}\}$, where $\tilde{Q}=\begin{bmatrix} I_N\otimes Q & 0 \\ 0 & Q_N \end{bmatrix}$, $\tilde{R}=I_N\otimes R$, and

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

Since $\tilde{A}$ is lower triangular with diagonal elements equal to -1, it follows that it is nonsingular. Let $\tilde{M}\triangleq \tilde{R}+\tilde{B}'(\tilde{A}^{-1})'\tilde{Q}\tilde{A}^{-1}\tilde{B}$. For any $z,\hat{z}\in\mathcal{Z}$

$$\begin{aligned}(\nabla V(z)-\nabla V(\hat{z}))'(z-\hat{z}) &= \begin{bmatrix} x-\hat{x} \\ u-\hat{u} \end{bmatrix}' \begin{bmatrix} \tilde{Q} & 0 \\ 0 & \tilde{R} \end{bmatrix} \begin{bmatrix} x-\hat{x} \\ u-\hat{u} \end{bmatrix} \\ &= (u-\hat{u})'\tilde{M}(u-\hat{u}) \\ &\geq \lambda_{\min}(R)\|u-\hat{u}\|^2 \\ &\geq \frac{\lambda_{\min}(R)}{1+\|\tilde{A}^{-1}\tilde{B}\|^2}\|z-\hat{z}\|^2\end{aligned}$$

where the second equality follows from $x=\tilde{A}^{-1}(\tilde{f}-\tilde{B}u)$, the first inequality from $\tilde{R}=I_N\otimes R$ and $R\in\mathbb{S}^{n_u}_{++}$ and the last inequality from $\|u-\hat{u}\|^2\geq(1/\|\tilde{A}^{-1}\tilde{B}\|^2)\|x-\hat{x}\|^2$.

D. Termination Criteria

Given accuracies $\varepsilon_V\geq 0$, $\varepsilon_g>0$, Theorems 4 and 5 provide a theoretical bound on the worst-case number of iterations needed by GPAD to compute an $(\varepsilon_V,\varepsilon_g)$-solution (cf. Corollary 7). In addition they can serve as practical termination criteria, since $\Psi(y_{(\nu+1)})$ appearing in (16a) can be computed during the course of the algorithm. Therefore at every iteration one can test $(\varepsilon_V,\varepsilon_g)$-optimality of the averaged iterate by examining if $\max_{i\in\mathbb{N}_{[1,m]}} g(\bar{z}_{(\nu)})\leq\varepsilon_g$, $V(\bar{z}_{(\nu)})-\Psi(y_{(\nu+1)})\leq\varepsilon_V$ are satisfied. Note that, in case $y_{(0)}=0$, by Corollary 6 one needs only to test if $\max_{i\in\mathbb{N}_{[1,m]}} g(\bar{z}_{(\nu)})\leq\varepsilon_g$ since $V(\bar{z}_{(\nu)})\leq V^\star$ always holds. This has the advantage of avoiding the computationally expensive calculation of the primal and dual cost. Note also that if g is affine [as in Problem (1)], then calculation of $g(\bar{z}_{(\nu)})$ requires $O(m)$ flops using the already calculated $g(z_{(\nu-1)})$, $g(z_{(\nu)})$.

One can also test the current iterate $z_{(\nu)}$ for $(\varepsilon_V,\varepsilon_g)$-optimality. We have observed experimentally that the cost accuracy and maximum constraint violation for the averaged iterate $\bar{z}_{(\nu)}$ converge smoothly (almost monotonically) with a behavior that is in accordance with the corresponding $O(1/\nu^2)$ bounds. On the other hand, the corresponding sequences for the running iterate $z_{(\nu)}$ are less predictable, exhibiting an oscillatory behavior. However, most often the latter reaches the required accuracy levels faster than the averaged iterate.

For QPs such as (1), the absolute accuracy criterion with respect to primal feasibility can be made scale-free; one can scale the linear constraints so as to have $c=\mathbf{1}$, $c_N=\mathbf{1}$ in problem (1), provided that the origin is contained in the interior of the constraint set. This type of scaling has been recently shown to improve an upper bound on the theoretical number of iterations for dual gradient methods [29].

However, for parametric optimization problems like (4), imposing a termination criterion based on absolute accuracy regarding the value function $V^\star(p)$ may be overly conservative in practice. For example, when MPC is applied to a regulation problem, if the current state p is far from the origin, the value function $V^\star(p)$ can be quite large, making an absolute accuracy termination criterion very hard to satisfy. On the other hand, when the current state is close to the origin, the value function can be arbitrarily close to zero, making a relative termination criterion hard to satisfy. Notice that if $y_{(0)}=0$, the latter issue concerns the optimality test for the current iterates only, since $V(\bar{z}_{(\nu)})\leq V^\star$ always. In practice it is more sensible to use a termination criterion that strikes a balance between absolute and relative accuracy

$$V\left(z_{(\nu)}\right)-\Psi\left(y_{(\nu+1)}\right)\leq\varepsilon_V\max\left\{\Psi\left(y_{(\nu+1)}\right),1\right\} \tag{27}$$

where now ε_V is the desired relative cost accuracy.

Since $\Psi(y_{(\nu+1)})\leq V^\star$, satisfaction of (27) implies $V(z_{(\nu)})-\Psi(y_{(\nu+1)})\leq\varepsilon_V\max\{V^\star,1\}$. The termination criterion (27) is particularly suitable for MPC problems where usually one has $V^\star\geq 0$. In that case, if $V^\star$ is larger than 1, then GPAD will provide a primal solution such that $(V(z_{(\nu)})-V^\star)/V^\star\leq\varepsilon_V$ is satisfied, which is more sensible than the absolute accuracy criterion. On the other hand, if $V^\star$ is nonnegative and close to zero, a relative accuracy stopping criterion becomes impractical. Equation (27) switches automatically between the two options *without requiring knowledge of* $V^\star$.

In many cases the calculation of the dual cost in (27) can be avoided. Specifically, if $w_{(\nu)}\geq 0$ then $V(z_{(\nu)})-V^\star\leq V(z_{(\nu)})-\Psi(w_{(\nu)})=-w'_{(\nu)}g(z_{(\nu)})$, where the inequality follows from dual feasibility of $w_{(\nu)}$ and the equality by (6) and Step 2 of GPAD. Therefore, if

$$-w'_{(\nu)}g\left(z_{(\nu)}\right)\leq\varepsilon_V\max\left\{1,\Psi\left(w_{(\nu)}\right)\right\} \tag{28}$$

then $V(z_{(\nu)})-V^\star\leq\varepsilon_V\max\{1,V^\star\}$. Since $\Psi(w_{(\nu)})=V(z_{(\nu)})+g(z_{(\nu)})'w_{(\nu)}$, it is easy to see that condition (28) is satisfied if and only if either

$$-w'_{(\nu)}g\left(z_{(\nu)}\right)\leq\varepsilon_V \tag{29}$$

or

$$-w'_{(\nu)}g\left(z_{(\nu)}\right)\leq\frac{\varepsilon_V}{1+\varepsilon_V}V\left(z_{(\nu)}\right) \tag{30}$$

is satisfied. Although in theory $w_{(\nu)}$ may not be nonnegative (cf. Step 1 of GPAD), in practice $w_{(\nu)}\geq 0$ holds after the first few iterations since $\theta_\nu(\theta_{\nu-1}^{-1}-1)$ converges rapidly to 1. The aforementioned observations lead to Algorithm 2 for the case $y_{(0)}=0$, which is very effective in practice according to our experience.

Due to Corollary 6, Algorithm 1 started with $y_{(0)}=0$ and the termination criterion given by Algorithm 2 will terminate after at most $\lceil\sqrt{8L_\Psi\Delta_y^\star/\varepsilon_g}\rceil-2$ steps.

Algorithm 2: Termination criterion

if $\max_{i\in\mathbb{N}_{[1,m]}} g(\bar{z}_{(\nu)}) \leq \varepsilon_g$ **then**
  stop
else if $\max_{i\in\mathbb{N}_{[1,m]}} g(z_{(\nu)}) \leq \varepsilon_g$ **then**
  if $w_{(\nu)} \geq 0$ **then**
    if *Condition* (29) *is satisfied* **then** stop
    else if *Condition* (30) *is satisfied* **then** stop
  else if *Condition* (27) *is satisfied* **then**
    stop
  endif
endif

V. Efficient Calculation of the Gradient of the Dual Function

The main computational burden of GPAD lies in solving problem $\arg\min_{z\in\mathcal{Z}(p)} \mathcal{L}(z,y)$ (Step 2 in Algorithm 1). Notice that the dual cost (6) can be written as

$$\Psi(p,y) = \min_z \sum_{k=0}^{N-1} \bar{\ell}(x_k,u_k,y_k) + \bar{V}_f(x_N,y_N)$$
$$\text{s.t. } x_0 = p,\ x_{k+1} = Ax_k + Bu_k + f, \quad k\in\mathbb{N}_{N-1} \quad (31)$$

where $\bar{\ell}(x,u,y_k) \triangleq \ell(x,u)+y_k'g_k(x,u), \bar{V}_f(x,y_N) \triangleq V_f(x)+y_N'g_f(x)$. The next proposition describes how the optimal input-state sequences for (31) can be calculated efficiently, for any vector $y\in\mathbb{R}^m$. It is an extension of the standard Riccati recursion for finite-horizon LQ unconstrained optimal control problems, adjusted appropriately to account for the contribution of the dual vector in the cost function in a computationally efficient manner.

Proposition 9: Let $K_k, D_k, M_k, d_k, L_k, C_k, s_k, k\in\mathbb{N}_{N-1}$ be calculated by Algorithm 3. Given $y\in\mathbb{R}^m$, the unique optimal input sequence z^y for (29) is given by Algorithm 4.

Proof: See Appendix A. ∎

Counting only arithmetic operations of cubic order (matrix-matrix products, factorizations and forward-backward substitutions for linear systems with matrix-valued right hand-sides), the total complexity for Algorithm 3 is roughly $N(3n_x^3 + 6n_x^2n_u + 6n_u^2n_x + (1/3)n_u^3 + 2m_sn_xn_u + 2m_sn_u^2)$ flops, which increases only linearly with the prediction horizon. Having performed the factor step, calculating the gradient of (29) is very cheap. Specifically, computing z^y takes $N(4n_x^2 + 6n_xn_u + 2m_s(n_x+n_u)) + 2m_Nn_x$ flops, which again increases only linearly with the prediction horizon. In many cases, one can explore additional problem structure to further reduce the computational cost of Algorithm 4. For example, in the usual case of state-input box constraints the cost is roughly $N(4n_x^2 + 8n_xn_u + n_u^2) + 2m_Nn_x$. In the context of MPC for LTI systems, the factor step (Algorithm 3) can be performed *off-line*. Algorithms 3 and 4 can be trivially extended to MPC for LTV systems. In that case, the factor step needs to be performed only once at every sampling time.

Remark 3: For LTI systems, if the terminal weight matrix Q_N is the solution of the discrete-time algebraic Riccati equation (DARE) corresponding to the infinite-horizon unconstrained problem, then Algorithm 3 takes $6n_x^2n_u + 5n_u^2n_x + (1/3)n_u^3 + 2m_sn_u^2 + 2m_sn_xn_u$ flops, *independently* of the horizon length N, since all quantities appearing in Algorithm 3 become time-invariant. Furthermore memory requirements are very small and *independent* of the horizon length.

Algorithm 3: Factor Step

$P_N = Q_N$
for $k = N-1,\dots,0$ **do**
  $\bar{R}_k = R + B'P_{k+1}B, \bar{S}_k = S + B'P_{k+1}A$ (32a)
  $P_k = Q + A'P_{k+1}A - \bar{S}_k'\bar{R}_k^{-1}\bar{S}_k$ (32b)
end
for $k = 0,\dots,N-1$ **do**
  $K_k = -\bar{R}_k^{-1}\bar{S}_k,\ D_k = -\bar{R}_k^{-1}G'$ (33a)
  $M_k = -\bar{R}_k^{-1}B',\ d_k = -\bar{R}_k^{-1}(r + B'P_{k+1}f)$ (33b)
  $L_k = (A+BK_k)',\ C_k = (F+GK_k)'$ (34a)
  $s_k = K_k'r + L_kP_{k+1}f + q$ (34b)
end

Algorithm 4: Solve Step

$e_N = F_N'y_N + q_N$
for $k = N-1,\dots,1$ **do**
  $e_k = L_ke_{k+1} + C_ky_k + s_k$ (35)
end
$x_0 = p$
for $k = 0,\dots,N-1$ **do**
  $u_k = K_kx_k + D_ky_k + M_ke_{k+1} + d_k$ (36a)
  $x_{k+1} = Ax_k + Bu_k + f$ (36b)
end

Remark 4: Riccati-like recursions coupled with block elimination for factoring the KKT matrix arising in interior point methods for MPC problems have been proposed in the literature [30], [31]. The factor step needs exactly the same number of flops as Algorithm 3 (excluding the calculation of D_k, M_k and Step 3 which becomes null in the setting described in [31]). However, in [31], the solve step costs $N(8n_x^2 + 6n_xn_u + n_u^2)$, while Algorithm 4 takes $N(2n_x^2 + 4n_xn_u)$. Therefore, our approach requires $N(6n_x^2 + 2n_xn_u + n_u^2)$ flops less.

Remark 5: Note that if the condensed dual formulation of [21] was used, then Steps 2 and 3 of Algorithm 1 would require a matrix-vector product whose cost is of order $O(N^2)$. The cost of forming the matrices of the condensed dual is of order $O(N^3)$ since the Cholesky factorization of $Nn_u \times Nn_u$ matrix is required to form the Hessian of the dual. Therefore, the condensed approach is less suitable for LTV systems than the approach of Algorithms 3 and 4. Finally, in the case of open-loop unstable systems, the condition number of the Hessian of the primal may increase with the prediction horizon, possibly causing potential numerical instabilities, which also makes the use Algorithms 3 and 4 more suitable.

VI. Certification of Computational Complexity

The problem of certifying the computational complexity of GPAD algorithm when applied to solve $\mathbb{P}(p)$ amounts to determining a uniform iteration bound [cf. (4), equivalently (1)],

defined as follows. For simplicity, it is assumed that GPAD is started from $y_{(0)} = y_{(-1)} = 0$.

Definition 10: Given $\mathcal{P} \subset \mathrm{dom}\ \mathcal{S}$, where $\mathcal{S}(p) \triangleq \{z \in \mathcal{Z}(p) | g(z) \leq 0\}$, $\varepsilon_g > 0$, integer $\nu^\star(\varepsilon_g)$ is called an *ε_g-uniform iteration bound* (ε_g-UIB, for short) for GPAD on $\mathcal{P}$ if for every $p \in \mathcal{P}$ and every $\nu \geq \nu^\star(\varepsilon_g)$, $\bar{z}_{(\nu)}$ is a $(0, \varepsilon_g)$-optimal solution for $\mathbb{P}(p)$.

Determining a UIB *a priori* is important in embedded control applications to enforce hard real-time properties of the MPC controller. One can easily infer from Corollary 7 that in order to obtain a UIB for GPAD on $\mathcal{P}$, one must derive a bound on

$$\Delta_y^\star(\mathcal{P}) \triangleq \sup_{p \in \mathcal{P}} \|y^\star_{\min}(p)\| \tag{37}$$

where $y^\star_{\min} : \mathrm{dom}\ \mathcal{S} \to \mathbb{R}^m$ is the single-valued mapping giving for each $p \in \mathrm{dom}\ \mathcal{S}$ the minimum Euclidean-norm dual optimal solution, i.e.,

$$y^\star_{\min}(p) \triangleq \arg\min \left\{ \frac{1}{2}\|y\|^2 | y \in \mathcal{Y}^\star(p) \right\}. \tag{38}$$

Definition 11: For a $\mathcal{P} \subset \mathrm{dom}\ \mathcal{S}$, we say that $\Delta_y(\mathcal{P}) < \infty$ is a *uniform dual bound (UDB) for $\mathbb{P}(p)$ on $\mathcal{P}$* if $\Delta_y^\star(\mathcal{P}) \leq \Delta_y(\mathcal{P})$.

The calculation of a UDB immediately leads to the ε_g-UIB [cf. (24)]

$$\nu \geq \left\lceil \sqrt{\frac{8 L_\Psi \Delta_y(\mathcal{P})}{\varepsilon_g}} \right\rceil - 2. \tag{39}$$

The next two subsections show how to compute a UDB for $\mathbb{P}(p)$. The first approach provides easily computable UDBs by calculating a bound for $\max_{y^\star(p) \in \mathcal{Y}^\star(p)} \|y^\star(p)\|$ on compact subsets of $\mathrm{int}(\mathrm{dom}\ \mathcal{S})$. The second approach provides a tight UDB, valid on the entire set of parameters for which $\mathbb{P}(p)$ is feasible, i.e., $\mathrm{dom}\ \mathcal{S}$ and it requires the solution of a *Linear Program with Linear Complementarity Constraints* (LPCC), for which dedicated solvers providing globally optimal solutions exist.

A. UDB Based on Slater's Condition

It turns out that finding an upper bound for $\max_{y^\star(p) \in \mathcal{Y}^\star(p)} \|y^\star(p)\|$ on a compact subset of $\mathrm{int}(\mathrm{dom}\ \mathcal{S})$ (assuming that $\mathrm{dom}\ \mathcal{S}$ has a nonempty interior) is a much easier problem than calculating $\Delta_y^\star(\mathcal{P})$. The need to confine the search for such a bound only on a compact set $\mathcal{P} \subseteq \mathrm{int}(\mathrm{dom}\ \mathcal{S})$ is dictated by Lemma 13 presented below. First, the following definition is needed.

Definition 12: 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 $g(\bar{z}) < 0$. We call any such vector $\bar{z}$ a *Slater vector* for $\mathbb{P}(\bar{p})$.

The first uniform dual bound is valid on a polytope $\mathcal{P}_S \subseteq \mathrm{int}(\mathrm{dom}\ \mathcal{S})$ given as the convex hull of a set of parameter vectors in $\mathrm{int}(\mathrm{dom}\ \mathcal{S})$. It requires the solution of an optimization problem for every point in the set. It is largely based on the following lemma which is an extension of [22, Ex. 5.3.1], [32, Lem. 1] for parametric optimization problems.

Lemma 13: Consider $\mathbb{P}(p)$. If $\mathrm{int}(\mathrm{dom}\ S) \neq \emptyset$, then $\mathbb{P}(\bar{p})$ satisfies the Slater condition if and only if $\bar{p} \in \mathrm{int}(\mathrm{dom}\ S)$, in which case $\mathcal{Y}^\star(\bar{p})$ is bounded and

$$\|y^\star(\bar{p})\| \leq \gamma^{-1}\left(V(\bar{z}) - V^\star(\bar{p})\right) \tag{40}$$

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 vector for $\mathbb{P}(\bar{p})$.

Proof: See Appendix B. ∎

Remark 6: For $\bar{p} \in \mathrm{int}(\mathrm{dom}\ \mathcal{S})$, the tightest possible bound of the form (41) can be obtained by solving $\min_{z,\gamma}\{\gamma^{-1}V(z) | z \in \mathcal{Z}(\bar{p}), g(z) + \gamma \leq 0, \gamma \geq 0\}$. Although this problem may not be convex, it is equivalent to $\Gamma(\alpha) = 0$, where

$$\Gamma(\alpha) = \min_{z \in \mathcal{Z}(\bar{p}), \gamma \geq 0} \left\{ V(z) - \alpha\gamma | g(z) + \gamma \leq 0 \right\}. \tag{41}$$

Notice that Γ is a univariate function for which one can compute a root by employing a generalized Newton method with respect to α, where at each iteration we solve (41), which is a convex QP, cf. [33]. The algorithm is very efficient and it converges only in a few iterations.

Proposition 14: Let $P_S = \{p^s\}_{s \in \mathbb{N}_{[1,S]}}$, with $p^s \in \mathrm{int}(\mathrm{dom}\ \mathcal{S})$, and

$$\gamma = \min\left\{-g_i(z^s) | s \in \mathbb{N}_{[1,S]}, i \in \mathbb{N}_{[1,m]}\right\} \tag{42a}$$
$$\beta = \max\left\{V(z^s) | s \in \mathbb{N}_{[1,S]}\right\} \tag{42b}$$

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 $\mathcal{P}_S \triangleq \mathrm{conv}\ P_S$.

Proof: See Appendix B. ∎

Lemma 13 can also be exploited to calculate an easily computable UDB for $\mathbb{P}(p)$ on $\mathcal{P}_\gamma \triangleq \mathrm{dom}\ \mathcal{S}_\gamma$, where

$$\mathcal{S}_\gamma(p) \triangleq \{z \in \mathcal{Z}(p) | g(z) + \gamma \leq 0\} \tag{43}$$

and $\gamma > 0$, as shown in the following proposition.

Proposition 15: Given $\gamma > 0$, if $V^\star \geq 0$ then $\Delta_y(\mathcal{P}_\gamma) = \gamma^{-1}\beta_\gamma$ is a UDB for $\mathbb{P}(p)$ on $\mathcal{P}_\gamma$, where

$$\beta_\gamma \triangleq \max_{p,z} \left\{ V(z) | z \in \mathcal{S}_\gamma(p) \right\}. \tag{44}$$

Proof: According to (43), for any $p \in \mathrm{dom}\ \mathcal{S}_\gamma$ there exists a Slater vector z such that $g_i(z) + \gamma \leq 0$, $i \in \mathbb{N}_{[1,m]}$. Invoking Lemma 13, this implies $\|y^\star(p)\| \leq \gamma^{-1}(V(z) - V^\star(p)) \leq \gamma^{-1}V(z)$, where the second inequality follows from nonnegativity of $V^\star$. Taking the maximum in both sides of the inequality among all $p \in \mathrm{dom}\ \mathcal{S}_\gamma$, we arrive at (44). ∎

Problem (44) entails the maximization of a convex quadratic function over a polyhedral set. Calculating a globally optimal solution for such a problem is NP-hard [34]. However, there exist efficient global optimization techniques [35], such as spatial branch-and-bound methods [36] or branch-and-bound combined with linear [37] and semidefinite [38] programming relaxations, to name a few. Notice that Proposition 15 remains valid if β_γ is merely an upper bound of the cost function of $\mathbb{P}(p)$ along the polyhedral set $\mathrm{gph}\ \mathcal{S}_\gamma$. The following corollary shows how one can compute such an upper bound, and therefore a UDB for $\mathbb{P}(p)$ on $\mathrm{dom}\ \mathcal{S}_\gamma$, by solving a single convex QP.

Corollary 16: Given $\gamma > 0$, if $V^\star \geq 0$ then $\hat{\Delta}_y(\mathcal{P}_\gamma) = \gamma^{-1}\hat{\beta}_\gamma$ is a UDB for $\mathbb{P}(p)$ on $\mathcal{P}_\gamma$, where $\hat{\beta}_\gamma$ is the optimal value of the following concave quadratic maximization problem

$$\hat{\beta}_\gamma = \max_{p,z} \left\{ V(z) - \frac{1}{2} r(z - z_\ell)'(z - z_u) | z \in \mathcal{S}_\gamma(p) \right\} \quad (45)$$

$r \geq \nabla^2 V(z)$ and z_ℓ, z_u are such that rge $\mathcal{S}_\gamma \subseteq \{z \in \mathbb{R}^n | z_\ell \leq z \leq z_u\}$.

Proof: Clearly, the function $V(z) - (1/2)rz'z$ is concave due to the choice of r, while $(p, z) \in \text{gph } \mathcal{S}_\gamma$ implies $(1/2)r(z - z_\ell)'(z - z_u) \leq 0$, therefore $V(z) - (1/2)r(z - z_\ell)'(z - z_u)$ overapproximates $V(z)$ on gph $\mathcal{S}_\gamma$ (see, e.g. [39, p. 278]). This implies that (45) is a concave quadratic maximization problem and $\hat{\beta}_\gamma \geq \beta_\gamma$. Invoking Proposition 15 we conclude that $\hat{\Delta}_y(\text{dom } \mathcal{S}_\gamma)$ is a UDB for $\mathbb{P}(p)$ on $\mathcal{P}_\gamma$. ■

B. UDB Based on Minimum-Norm Dual Optimal Solution

Although UDBs based on Proposition 14 and Corollary 16 are easy to compute even for large-scale systems, usually they are not tight, leading to conservative UIBs. Furthermore, these UDBs are valid only on a subset of the interior of dom $\mathcal{S}$. However, tight UIBs valid on the entire dom $\mathcal{S}$ can be crucial in embedded MPC applications for systems with fast sampling rates, where stability and invariance guarantees are required. On the other hand, the tightest UDB is clearly $\Delta_y^\star(\text{dom } \mathcal{S})$ [cf. (37)]. Notice that since for $p \in \text{dom } \mathcal{S}$, $\mathcal{Y}^\star(p)$ is a nonempty polyhedral set, there always exists a unique $y^\star_{\min}(p)$ that attains the minimum in (38). In principle one can determine $\mathcal{Y}^\star(p)$ explicitly (by solving the convex parametric quadratic optimization problem $\mathbb{D}(p)$ using the algorithm of [40]) and then calculate $y^\star_{\min}$ for each critical region. However, this procedure is valid for small-scale problems only.

In this subsection we will show how one can efficiently compute the UDB

$$\Delta_y^1(\text{dom } \mathcal{S}) \triangleq \sup_{p \in \text{dom } \mathcal{S}} \|y^\star_{\min}(p)\|_1 \quad (46)$$

without solving the parametric problem $\mathbb{D}(p)$ explicitly. The quantity $\Delta_y^1(\text{dom } \mathcal{S})$ is a tight approximation of $\Delta_y^\star(\text{dom } \mathcal{S})$ as the next proposition shows.

Proposition 17: Consider $\Delta_y^\star(\text{dom } \mathcal{S}), \Delta_y^1(\text{dom } \mathcal{S})$ given by (37), (47) respectively. Then

$$\Delta_y^\star(\text{dom } \mathcal{S}) \leq \Delta_y^1(\text{dom } \mathcal{S}) \leq \sqrt{m}\Delta_y^\star(\text{dom } \mathcal{S}).$$

Proof: Follows directly from $\|y^\star_{\min}(p)\| \leq \|y^\star_{\min}(p)\|_1 \leq \sqrt{m}\|y^\star_{\min}(p)\|$, $p \in \text{dom } \mathcal{S}$. ■

We next present a new characterization of the minimum Euclidean-norm dual optimal solution of $\mathbb{P}(p)$. To keep notation simple, the results are presented for the condensed dual formulation

$$\min \left\{ \frac{1}{2} y'Hy + (Dp + d)'y | y \geq 0 \right\} \quad (47)$$

where H, D, d can be computed from (1) [21] (see also Remark 5). First, we need the following lemma.

Lemma 18: Consider a $p \in \text{dom } \mathcal{S}$, a $y^\star \in \mathcal{Y}^\star(p)$ and let $N(p) \triangleq \{i \in \mathbb{N}_{[1,m]} | H_{i.}y^\star + D_{i.}p + d_i > 0\}$. The solution set of (47) is given by

$$\mathcal{Y}^\star(p) = \left\{ y \in \mathbb{R}^m_+ | Hy = Hy^\star,\ y_i = 0,\ i \in N(p) \right\}. \quad (48)$$

Proof: See [40, Prop. 1]. ■

Theorem 19: Consider $p \in \text{dom } \mathcal{S}$. Then $y \in \mathbb{R}^m$ is the minimum-norm solution of (47) if and only if there exists a $\lambda \in \mathbb{R}^m$ such that

$$0 \leq y \perp Hy + Dp + d \geq 0 \quad (49a)$$

$$y_i(y_i + H_{i.}\lambda) = 0,\ y_i + H_{i.}\lambda \geq 0,\ i \notin N(p). \quad (49b)$$

Proof: See Appendix B. ■

Based on Theorem 19 we can compute the UDB $\Delta_y^1(\text{dom } \mathcal{S})$ [cf. (47)] by solving the optimization problem

$$\max_{p,y} \quad \sum_{i=1}^m y_i \quad \text{s.t.} \quad (50a),\ (50b). \quad (50)$$

Assuming that bounds on y, z, λ are available, Problem (50) can be modeled as a Mixed-Integer Linear Program (MILP) by employing the so-called "big-M" technique (see e.g. [41, Prop. 2]) and using a small tolerance (e.g., the machine precision) beyond which any number is considered to be positive to model the clause $[\delta_i = 1] \iff [z_i = 0]$ (see, e.g., [42, Eq. (4e)]). In practice one can iterate on estimating large-enough bounds on y, z, λ. The resulting MILP would contain $2m$ binary variables and the choice of tolerance ϵ in conjunction with large bounds on the variables can cause numerical problems to MILP solvers.

Next, another UDB valid on dom $\mathcal{S}$ will be presented, whose calculation requires the solution of an LPCC, for which there exist specialized global optimization algorithms, e.g., [43] and [44], that do not rely on the big-M formulation and thus, do not require bounds on variables.

Notice that every index set $N' \supseteq N(p)$ defines a (possibly empty) face of the polyhedral set $\mathcal{Y}^\star(p)$ [cf. (48)], i.e., $\mathcal{Y}^\star_{N'}(p) = \{y \in \mathbb{R}^m_+ | Hy = Hy^\star, y_i = 0, i \in N'\}$. The next lemma sets the stage for the establishment of a tight UDB on dom $\mathcal{S}$ which is easier to compute than $\Delta_y^1(\text{dom } \mathcal{S})$. The key idea is that instead of computing the minimum-norm element of $\mathcal{Y}^\star(p)$, one could more easily compute the minimum-norm optimal solution along some face of $\mathcal{Y}^\star(p)$, which at the same time lies in the relative interior of the face. Notice that since $\mathcal{Y}^\star(p)$ is pointed (it does not contain a line) it has at least one extreme point, which automatically satisfies the above requirement.

Lemma 20: Consider any $p \in \text{dom } \mathcal{S}$ and let $\hat{y} \in \mathcal{Y}^\star(p)$ and $\hat{N} = \{i \in \mathbb{N}_{[1,m]} | \hat{y}_i = 0\}$. Then $\hat{y} = \arg\min\{(1/2)\|y\|^2 | y \in \mathcal{Y}^\star_{\hat{N}}(p)\}$ if and only if there exists a $\lambda \in \mathbb{R}^m$ such that $\hat{y}_i(\hat{y}_i + H_{i.}\lambda) = 0$, $i \in \mathbb{N}_{[1,m]}$.

Proof: See Appendix B. ■

Based on Lemma 20, the following theorem characterizes all such dual optimal solutions via complementarity conditions. Furthermore, it provide an alternative characterization for the minimum-norm element of $\mathcal{Y}^\star(p)$.

Theorem 21: For $p \in \operatorname{dom} \mathcal{S}$, let

$$\mathcal{Y}^\star_{\min}(p) \triangleq \left\{ y \left| \begin{array}{l} 0 \le y \perp Hy + Dp + d \ge 0, \exists \lambda \in \mathbb{R}^m \\ \text{s.t. } y_i(y_i + H_i.\lambda) = 0,\ i \in \mathbb{N}_{[1,m]} \end{array} \right. \right\}. \quad (51)$$

Then $\mathcal{Y}^\star_{\min}(p) \subseteq \mathcal{Y}^\star(p)$ is nonempty, bounded, and

$$y^\star_{\min}(p) = \arg\min \left\{ \frac{1}{2}\|y\|^2 | y \in \mathcal{Y}^\star_{\min}(p) \right\}. \quad (52)$$

Proof: See Appendix B. ∎

Next, define the following UDB:

$$\tilde{\Delta}_y(\operatorname{dom} \mathcal{S}) \triangleq \sup \{ \|y\|_1 | (p, y) \in \operatorname{gph} \mathcal{Y}^\star_{\min} \}. \quad (53)$$

Taking into account (51), UDB $\tilde{\Delta}_y(\operatorname{dom} \mathcal{S})$ can be expressed as the optimal cost of the LPCC

$$\begin{aligned} \tilde{\Delta}_y(\operatorname{dom} \mathcal{S}) = \max_{p,y,\lambda} & \sum_{i=1}^m y_i \\ & \text{s.t. } z = Hy + Dp + d,\ w = y + H\lambda, \\ & 0 \le y \perp z \ge 0,\ y_i w_i = 0,\ i \in \mathbb{N}_{[1,m]} \quad (54). \end{aligned}$$

As the next proposition shows, $\tilde{\Delta}_y(\operatorname{dom} \mathcal{S})$ is no smaller than $\Delta^1_y(\operatorname{dom} \mathcal{S})$. However, the next proposition provides a practical criterion (involving only the calculation of $y^\star_{\min}(\tilde{p})$ at some point $\tilde{p}$) which ascertains their equality without calculating $\Delta^1_y(\operatorname{dom} \mathcal{S})$. Specifically, after an optimal solution $(\tilde{p}, \tilde{y})$ of (55) has been computed, one can solve the QP defined in (39) for $p = \tilde{p}$. If the resulting dual vector $y_{\min}(\tilde{p})$ is equal to $\tilde{y}$ then $\tilde{\Delta}_y(\operatorname{dom} \mathcal{S})$ is equal to $\Delta^1_y(\operatorname{dom} \mathcal{S})$.

Proposition 22: One has $\tilde{\Delta}_y(\operatorname{dom} \mathcal{S}) \ge \Delta^1_y(\operatorname{dom} \mathcal{S})$. If $(\tilde{p}, y^\star_{\min}(\tilde{p})) \in \arg\max\{\|y\|_1 | (p, y) \in \operatorname{gph} \mathcal{Y}^\star_{\min}\}$ then $\tilde{\Delta}_y(\operatorname{dom} \mathcal{S}) = \Delta^1_y(\operatorname{dom} \mathcal{S})$.

Proof: Notice that $\Delta^1_y(\operatorname{dom} \mathcal{S}) = \sup\{\|y\|_1 | p \in \operatorname{dom} \mathcal{S}, y = y^\star_{\min}(p)\}$ and $\{(p, y) | p \in \operatorname{dom} \mathcal{S}, y = y^\star_{\min}(p)\} \subseteq \operatorname{gph} \mathcal{Y}^\star_{\min}$ due to Proposition 21. Therefore, every feasible solution of (47) is also a feasible solution of (53) proving $\tilde{\Delta}_y(\operatorname{dom} \mathcal{S}) \ge \Delta^1_y(\operatorname{dom} \mathcal{S})$. Now, if the premise of the proposition holds, one has $\|y^\star_{\min}(\tilde{p})\|_1 \le \Delta^1_y(\operatorname{dom} \mathcal{S}) \le \tilde{\Delta}_y(\operatorname{dom} \mathcal{S}) = \|y^\star_{\min}(\tilde{p})\|_1$. ∎

VII. Simulations

A. Ball and Plate Example

The example considers a plate tilted around two axes to control the position of a ball [16]. Assuming a sampling time equal to 0.01 s, the motion of the ball along each axis on the plate can be described by a discrete-time LTI model, whose states are the ball position and velocity along the axis, the input is the tilt angle, and the state-update matrices are $A = \begin{bmatrix} 1 & 0.01 \\ 0 & 1 \end{bmatrix}$, $B = \begin{bmatrix} -0.0004 \\ -0.0701 \end{bmatrix}$. The state vector is constrained in $[-0.2, 0.01] \times [-0.1, 0.1]$, while the input must belong to $[-0.0524, 0.0524]$. The stage cost is the same as in [16], i.e., $\ell(x, u) = (1/2)(x'Qx + u'Ru)$ with $Q = \begin{bmatrix} 100 & 0 \\ 0 & 10 \end{bmatrix}$, $R = 1$.

1) Comparison With [17]: In the first experiment we compare GPAD against the fast dual gradient method proposed in [17]. We remark that the purpose of the experiment is mainly to highlight the fact that although both of the algorithms are based on Nesterov's accelerated gradient method, their behavior can be completely different in practice.

TABLE I
Comparison With [17] on the Ball and Plate Example for Achieving Dual Accuracy $\varepsilon_\Psi = 10^{-2}$

	Iterations		UIB		
N	GPAD	[17]	Eq.(54)	Cor. 16	Prop. 14
5	2	91	12	277	54743
7	4	122	28	395	72341
9	6	153	52	517	90309
11	7	183	84	639	108378
13	8	214	125	761	126568
15	10	215	176	885	144908

The comparison concerns both the performance of the algorithms in terms of observed iterations and the tightness of the proposed UDBs. For this reason, we replicate the simulation setting of [17]. The terminal weight matrix $Q_N = Q$ and F_N, c_N represent the bound constraints on the state vector. The required accuracy on the dual cost is $\varepsilon_\Psi = 10^{-2}$ and horizon length varies from $N = 5$ to $N = 15$ with step 2. The set of initial states is obtained by scaling the maximum admissible set for $N = 15$, in accordance to Figure 1 of [17]. We compute a worst-case number of iterations for both algorithms, by forming a uniform grid on the set of initial states and computing for each point of the grid a dual solution whose dual cost is no larger than ε_Ψ from the optimal cost. Since there is no termination criterion presented in [17], for each parameter point the exact optimal cost is computed using Gurobi [45] and used by the algorithms for their termination. The results of the simulations are presented in Table I. In the third column, the worst-case number of iterations for [17] is shown, which is aligned with the results appearing in that paper. As it can be seen from the second column of Table I, the worst-case number of iterations for GPAD is much smaller. We remark that the cost per iteration of both algorithms is of the same order. In the last three columns of Table I, the UIBs computed according to (53), Corollary 16, and Proposition 14 are reported. The UIBs coming from Proposition 14 are roughly one order of magnitude smaller than the ones reported in Fig. 2 of [17]. However, UIBs coming from (53) and Corollary 16 are much tighter.

2) Certification Analysis: The purpose of the next experiment is to test the tightness of the UDBs proposed in Section VI. Two separate scenarios are analyzed. In the first scenario, the terminal cost is $V_f(x) = (1/2)x'P_f x$, where P_f is the solution of the corresponding DARE and there are no terminal constraints in the MPC formulation; in the second scenario, V_f is the same while F_N, g_N represent the maximal positively invariant (MPI) set of the system in closed-loop with the LQR feedback law. We remark here, that both scenarios cannot be handled by the algorithm in [17], since P_f is not diagonal, and the polyhedral terminal constraint set has a complicated structure. To make the comparison fair, the obtained UIBs are compared on $\mathcal{P}_\gamma = \{p | H_{\mathcal{S}} p \le (1 - \gamma)\mathbf{1}\}$, $\gamma \ge 0$, where $\operatorname{dom} \mathcal{S} = \{p | H_{\mathcal{S}} p \le \mathbf{1}\}$ is calculated via projection. However, we remark that the calculation of UDBs of (53) and Corollary 16 does not require the projection of a polytope, in general. The certification analysis is carried out for GPAD when applied to Problem (1) with prediction horizon ranging from $N = 5$ to $N = 15$ with step 2 and

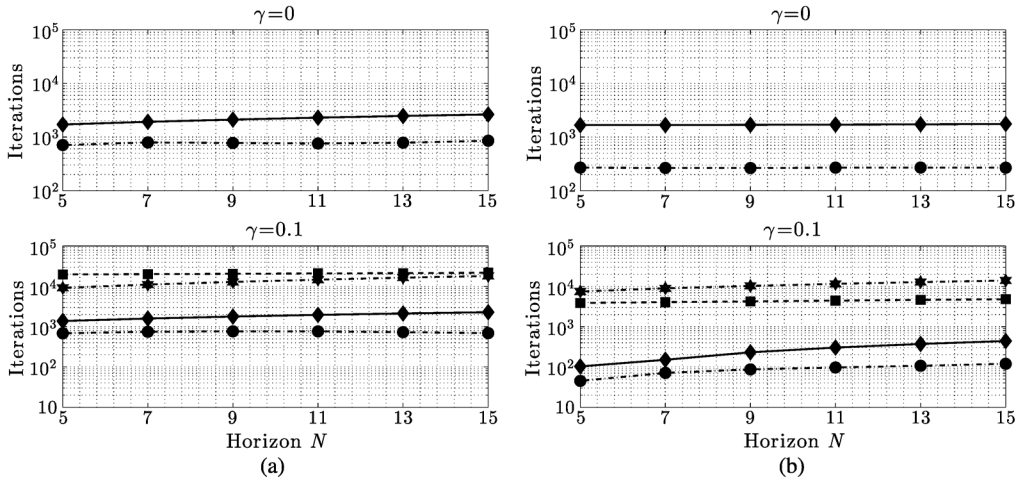

Fig. 1. Complexity certification of MPC without terminal constraints for the ball on the plate example: 10^{-2}-UIB based on (53) (solid-diamond), based on Proposition 14 (dash-square), based on Corollary 16 (dash-dot-star), and observed maximum number of iterations based on sampling (dash-dot-circle). (a) Without terminal constraints; (b) with terminal constraints.

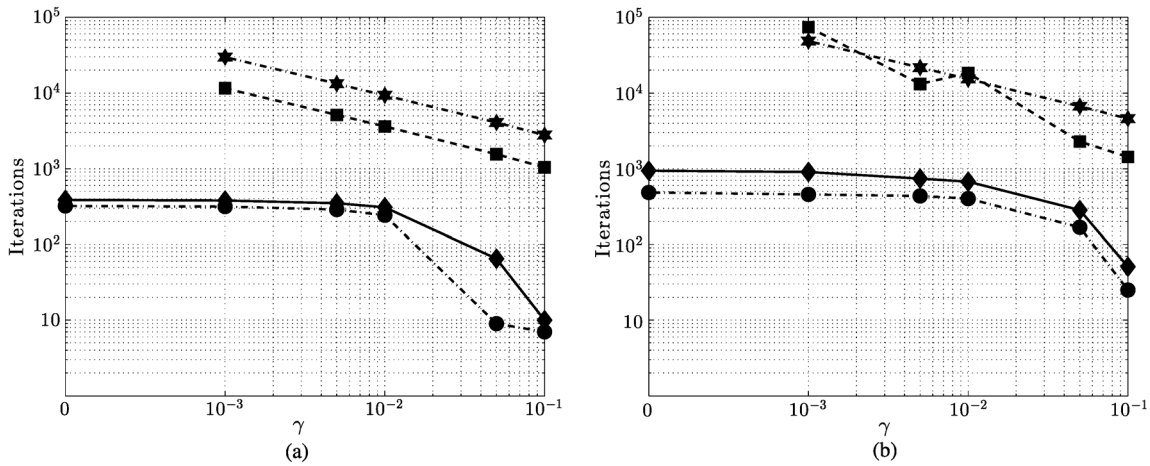

Fig. 2. Complexity certification of MPC with terminal constraints for the masses example: 10^{-2}-UIB based on (53) (solid-diamond), based on Proposition 14 (dash-square), based on Corollary 16 (dash-dot-star), and observed maximum number of iterations based on sampling (dash-dot-circle) for various values of γ. (a) Masses $M = 2$, horizon $N = 10$; (b) masses $M = 3$, horizon $N = 10$.

parameter $\gamma \in \{0, 0.10\}$. Notice that UDBs given by Corollary 16 and Proposition 14 are finite only in the case of γ being positive. The certification analysis is conducted for each N and γ, by applying the following steps:

1) constraints of (1) are normalized so as to have $c = \mathbf{1}$, $c_N = \mathbf{1}$;
2) the Lipschitz constant is calculated as in [21];
3) compute UDB (53) for $\mathbb{P}(p)$ on $\mathcal{P}_\gamma$. Problem (54) giving UDB (53) is solved using the mixed-integer solver of CPLEX [46], by modeling the complementarity constraints with indicator variables, thus removing the need to select proper big-M values;
4) compute UDB of Corollary 16, for $\mathbb{P}(p)$ on $\mathcal{P}_\gamma$;
5) compute UDB of Proposition 14, for $\mathbb{P}(p)$ on $\mathcal{P}_\gamma$, with the collection of points being the vertices of $\mathcal{P}_\gamma$;
6) for each UDB compute the corresponding ε_g-UIB according to (39), where $\varepsilon_g = 10^{-2}$;
7) compute a worst-case number of iterations based on sampling, by forming a uniform grid on $\mathcal{P}_\gamma$ and computing for each point of the grid a $(0, 10^{-2})$-solution, using Algorithm 2.

The results of the analysis of the two scenarios are summarized in Fig. 1. One can observe that the UIBs obtained by the UDB (53) are much tighter than the ones of Proposition 14 and Corollary 16, as expected. In fact, in all cases the assumptions of Proposition 22 are satisfied, showing that UDB (53) is tight in the sense of Proposition 17. Furthermore, contrary to the UDBs of Proposition 14 and Corollary 16, the UDBs of (53) are also valid on the boundary of $\mathrm{dom}\,\mathcal{S}$, and one can observe that for the particular example, the UDB does not increase too much as one moves from the interior to the boundary. However, extensive simulations have shown that for many MPC problems this is not the case, with the corresponding UDB increasing rapidly as the boundary of $\mathrm{dom}\,\mathcal{S}$ is approached.

The worst-case iteration counts of Algorithm 2 are within 15–59% of the tight UIBs obtained by (53), confirming that the theoretical bound of (39) is quite tight. On the downside, UDB (53) is harder to compute than the UDBs based on Proposition 14 and Corollary 16. Interestingly, one can observe that the UIBs obtained for the scenario where no terminal constraints are present are tighter for $\gamma = 0$. On the other hand, the number of iterations, both in theory and in practice, are smaller for the scenario in which terminal constraints are present, although the

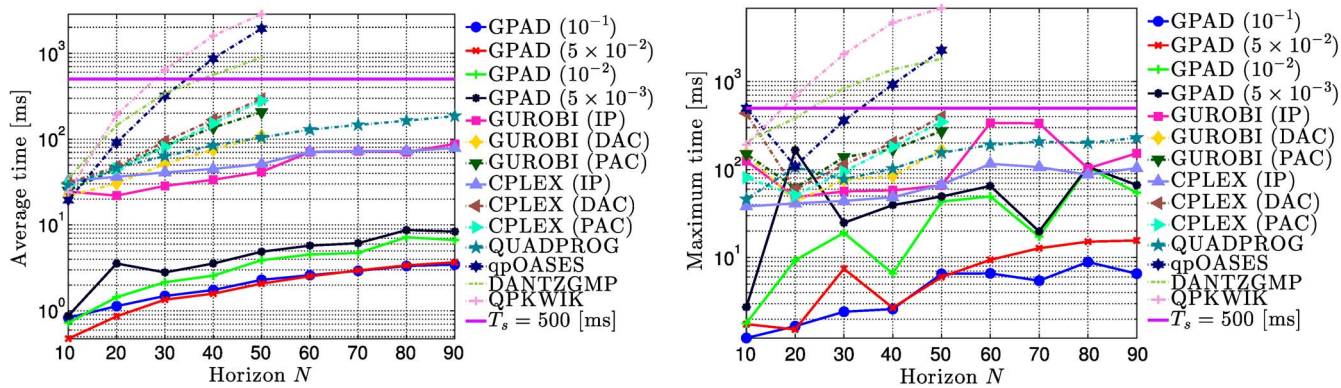

Fig. 3. Runtime results for the masses example ($M = 5$).

problem is more "complex" due to the presence of the terminal set.

B. Masses Example

The setup for this example is similar to [8], [9]. It consists of a sequence of M objects of equal mass m serially connected to each other, by spring-dampers of equal spring constant k and damping coefficient c, and to walls on either side. Between two consecutive masses there is and actuator exerting tensions. The discrete-time LTI model is obtained after discretization with sampling period of 0.5 s, and it consists of $n_x = 2M$ states and $n_u = M - 1$ inputs. Each state of the system is constrained to lie in $[-4, 4]$, while each input in $[-1, 1]$. The stage cost is $\ell(x, u) = (1/2)(x'Qx + u'Ru)$ with $Q = I_{n_x}$, $R = I_{n_u}$.

1) Complexity Certification: The purpose of the first set of simulations is to test the tightness of the UDBs proposed in Section VI to problems of larger dimension than the previous example. It is assumed that $m = 1$, $k = 2$, $c = 1$. Furthermore the terminal cost is $V_f(x) = (1/2)x'P_f x$, where P_f is the solution of the corresponding DARE, and F_N, g_N represent the MPI set of the system in closed-loop with the LQR feedback law. The UIBs obtained by computing the UDBs of (53), Corollary 16, and Proposition 14 are compared on the masses example for $M = \{2, 3\}$, $N = 10$, $\gamma = \{0, 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}, 10^{-1}\}$. The certification analysis follows exactly the same Steps 1–7 of Section VII-A. The results are summarized in Fig. 2. The worst-case iteration counts for Algorithm 2 are within 14–83% and 49–60% of the UIBs obtained by (53), for $M = 2$ and $M = 3$, respectively. Therefore for the specific example the theoretical bound of (24) is quite tight. In both cases, Proposition 22 is valid confirming that the UDBs of (53) are tight. The parameter value $\tilde{p}$ appearing in Proposition 22 lies always on the boundary of $\mathrm{dom}\, \mathcal{S}$, a phenomenon which appears in most of our simulations. For $\gamma = 0$, the tight UDBs obtained by (53) are 588 ($M = 2$) and 545 ($M = 3$). In fact, UDBs can take significantly large values especially for $\gamma = 0$, as we have observed in numerous simulations. In such cases, the advantage of choosing $y_{(0)} = 0$ is evident since the UIB of (39) is of order $O(\sqrt{\Delta_y})$, while the iteration bounds for primal and dual optimality are of order $O(\Delta_y)$.

2) Performance Comparison: The purpose of the next experiment is to compare the practical performance of GPAD against existing solvers: the interior-point, primal active set, and dual active set solvers of GUROBI 5.0 [45] and CPLEX v12.4 [46], the interior-point solver QUADPROG of MATLAB R2012a Optimization Toolbox, QPOASES [5], the QP solver of the MPC Toolbox [47] DANTZGMP, based on the active set method described in [4], and the new QP solver of the MPC Toolbox implementing the QPKWIK algorithm [6].

As the emphasis of the comparison is on worst-case execution time that is a fundamental parameter in real-time embedded optimization, all algorithms are cold-started. The negative effect of this choice is more significant for active-set solvers, especially for QPOASES which takes most advantage of warm-starting.

The algorithms are tested on the MPC problem (1) for the masses example, with the number of masses M ranging from 5 to 25 with step 5 and horizon length N ranging from 10 to 90 with step 10. Here, $m = k = 1$, $c = 0$ and the terminal conditions are the same as in the previous subsection.

For GPAD, the inequality constraints of the MPC problem were scaled so as to have $c = \mathbf{1}$, $c_N = \mathbf{1}$, the initial iterate is always $y_{(0)} = 0$, and the termination criterion of Algorithm 2 is checked at every iteration. The resulting MPC problem was solved for relative accuracies $\varepsilon_V = \varepsilon_g \in \{10^{-1}, 5 \times 10^{-2}, 10^{-2}, 5 \times 10^{-3}\}$. GPAD was programmed in C (compiler GCC 4.2) using BLAS for matrix-vector and inner products, and called through MATLAB via a mex interface. The MPC problem matrices were passed in sparse form (keeping the states as decision variables) to the interior-point solvers, and in condensed form (having eliminated the states) to the active set solvers. For each of the solvers their default termination criteria were used. For every different combination of M, N, 100 different initial states were chosen at random by solving the LP $\min_{p,z}\{r'p | z \in \mathcal{Z}(p), g(z) + \gamma \leq 0\}$, where the elements of $r \in \mathbb{R}^{2M}$ are drawn from the standard normal distribution and γ is uniformly distributed between 0 and 0.6.

Simulations were performed on a MacBook Pro (2.66 GHz Intel Core 2 Duo, 4 GB RAM). All algorithms were called in MATLAB through their mex interfaces and runtimes were measured using the `tic-toc` command of MATLAB. The results of the simulation experiments are summarized in Figs. 3–5. The active-set solvers QPOASES, DANTZGMP, QPKWIK were run only for $M \in \{5, 10\}$ and $N \in \{10, \ldots, 50\}$ since their average runtime well exceeds the sampling time of $T_s = 500$ ms. The same is true for the active set solvers of GUROBI and CPLEX when $N > 50$.

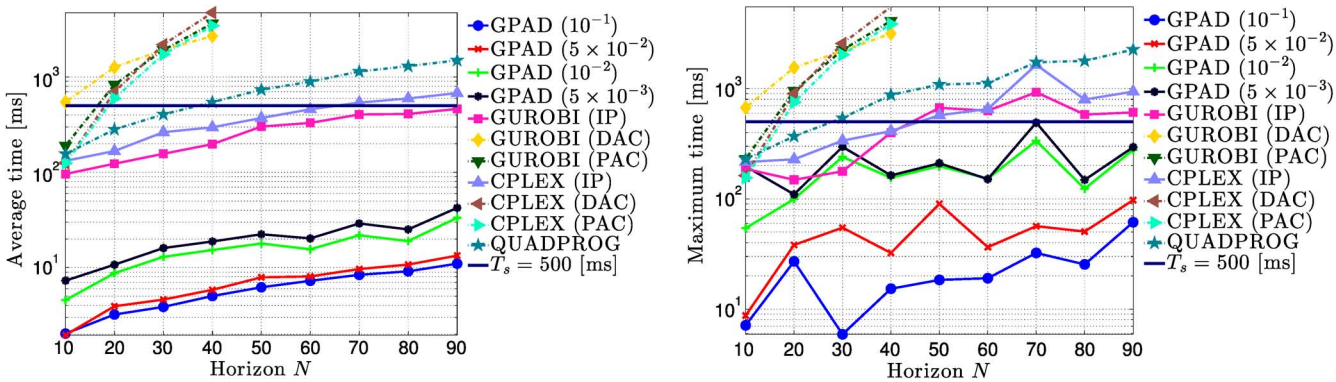

Fig. 4. Runtime results for the masses example ($M = 15$).

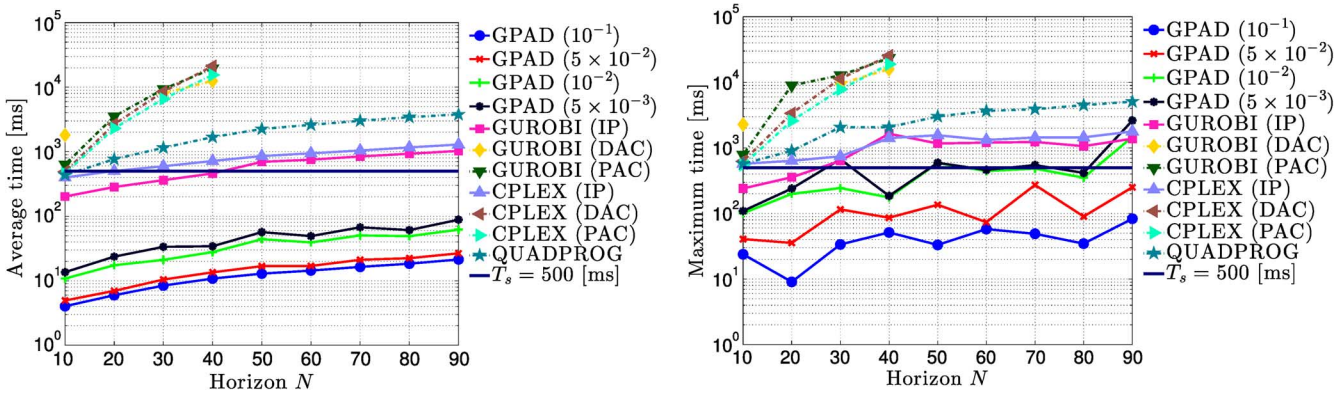

Fig. 5. Runtime results for the masses example ($M = 25$).

As the running time of active set solvers grows rapidly with problem size, we do not report the corresponding results for large problems. Clearly, as observed earlier, their performance can be significantly improved by warm-starting, an information that might not always be available during MPC operations, such as after a set-point change or the occurrence of a large disturbance.

One can observe that GPAD outperforms all the solvers in terms of average runtime by more than an order of magnitude. For relative accuracies $\varepsilon_V = \varepsilon_g \in \{10^{-1}, 5 \times 10^{-2}\}$ its worst-case runtime is well below the sampling time of 500 ms, even for long horizons and large number of masses. For $M = \{20, 25\}$, the worst-case runtime of GPAD with relative accuracy 10^{-2} or 5×10^{-3} exceeds the sampling time T_s for some values of the horizon length. Among the rest of the solvers, the interior-point algorithm of GUROBI seems to be the most efficient. For up to 10 masses, its worst-case runtime is less than T_s, while for larger number of masses it exceeds it for horizon length above certain threshold. Overall, the conclusion drawn is that, if only medium accuracy solutions are sought, GPAD is the algorithm of choice for the specific example, not to mention that it is even the simplest to implement.

We remark that the offline time needed to build problem matrices for the active-set problems is not included in the figures. However this grows very fast with the prediction horizon and number of masses. Due to the choice of the terminal cost and Remark 3, the time needed by the factor step (cf. Algorithm 3) is independent of the horizon length. Specifically, the execution time of Algorithm 3 is 47, 110, 169, 288, 811 μs for $M = 5$, 10, 15, 20, 25, respectively. In all cases it is almost negligible compared to the average time per iteration of GPAD.

Regarding the efficacy of the termination criterion (cf. Algorithm 2), in 35.9% of the cases the algorithm was stopped due to criterion (28), in 52.5% due to criterion (29) and in 11.43% due to criterion (27). Criterion (28) was successful 25.7% of the times it was tested, while the corresponding percentage for (29) and (27) is 51% and 65.27% respectively. Finally, criterion (28) was tested in 6% of the total number of iterations, while the corresponding percentage for (29) and (27) is 4.48% and 0.76% respectively. The conclusion drawn from these numbers is that Algorithm 2 reduces the number of iterations needed for the termination of Algorithm 1 in practice, since the criterion $g(\bar{z}_{(\nu)}) \leq \varepsilon_g$ dictated by theory is hardly ever satisfied before (28), (29) or (27). Furthermore, the most computationally expensive criterion (27) which requires the calculation of the dual function is rarely checked and when it does, almost 2/3 of the times is successful.

VIII. Conclusions and Future Work

This paper has proposed an accelerated dual gradient projection algorithm that is particularly tailored to embedded applications of linear MPC designs. Although GPAD is a dual algorithm, its convergence properties derived in this paper regard primal optimality and feasibility. Specifically, GPAD can compute an $(0, \varepsilon)$-optimal solution for the primal problem in $O(1/\sqrt{\varepsilon})$ iterations. Although GPAD has a computation

performance that is comparable to other existing QP solvers, its main purpose is to be extremely simple to code (as it only involves products, sums, and comparisons), and to have rather tight upper-bounds (computable offline) on the maximum number of iterations. These characteristics ease the certification of the control code and make it quite suitable for implementation in embedded systems, compared to other (possibly faster, but more complicated) solvers.

One drawback of GPAD is its sensitivity to scaling, a fact that is well known for first-order methods [22]. Preconditioning can dramatically improve the convergence rate of GPAD. According to our experience, one simple choice of preconditioning that works well in practice is to compute a diagonal approximation $\tilde{H} \in \mathbb{S}^m_{++}$ of the Hessian of the dual cost in (47) and perform a change of coordinates in the dual space with scaling matrix equal to $\tilde{H}^{-1/2}$ [22, Sec. 2.3.1]. An alternative approach was suggested recently in [29] and [48].

Ongoing and future work includes closed-loop stability analysis of MPC under $(\varepsilon_V, \varepsilon_g)$-optimality, and derivation of convergence and bounds for the algorithm in the presence of errors due to fixed-point computations [49], [50].

APPENDIX A

Proof of Proposition 9: For each $k \in \mathbb{N}_{N-1}$ consider the *Hamiltonian function* $H_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_x} \to \mathbb{R}$ [22, Sec. 1.9], $H_k(x,u,\lambda) = \bar{\ell}(x,u,y_k) + \lambda'(Ax+Bu+f)$. Since H_k is jointly convex in (x,u) and strictly convex in u, the necessary and sufficient condition for z to be the unique optimal input-state sequence for (5) is

$$\nabla_{u_k} H_k(x_k, u_k, \lambda_{k+1}) = 0,\ k \in \mathbb{N}_{N-1} \tag{56}$$

where the *costate* vectors $\lambda_k \in \mathbb{R}^{n_x}$ satisfy the adjoint equations

$$\lambda_k = \nabla_{x_k} H_k(x_k, u_k, \lambda_{k+1}),\ k \in \mathbb{N}_{[1,N-1]} \tag{57a}$$
$$\lambda_N = \nabla_{x_N} \bar{V}_f(x_N, y_N). \tag{57b}$$

Next, we claim that

$$\lambda_{k+1} = P_{k+1}x_{k+1} + e_{k+1},\ k \in \mathbb{N}_{N-1} \tag{58}$$

with $e_N = F'_N y_N + q_N$ and e_k, $k \in \mathbb{N}_{[1,N-1]}$ given by (36). For $k = N-1$, the claim is valid since by (57b), $\lambda_N = Q_N x_N + q_N + F'_N y_N$. Assume that (58) holds for some $k \in \mathbb{N}_{N-1}$. Solving (56) with respect to u_k, one gets

$$u_k = -R^{-1}\left(Sx_k + r + G'y_k + B'\lambda_{k+1}\right). \tag{59}$$

Substituting (58) in (59), we obtain $u_k = -R^{-1}(Sx_k + r + G'y_k + B'(P_{k+1}x_{k+1} + e_{k+1}))$. Substituting the state equation $x_{k+1} = Ax_k + Bu_k + f$, and solving for u_k, we arrive at (37a), with K_k, M_k, D_k, d_k given by (34).

Next, multiplying the state equation by $A'P_{k+1}$ and using (58) we arrive at

$$A'\lambda_{k+1} = A'P_{k+1}Ax_k + A'P_{k+1}Bu_k + A'e_{k+1} + A'P_{k+1}f$$

Adding $Qx_k + S'u_k + q + F'y_k$ in both sides, using (57a), (37a) and rearranging terms we obtain $\lambda_k = P_k x_k + e_k$, where e_k is given by (36) in Algorithm 4 with L_k, C_k, s_k given by (35) confirming the induction argument and completing the proof. ■

APPENDIX B

Proof of Lemma 13: Let $\mathcal{C} = \{(p,z) | z \in \mathcal{Z}(p), g(z) \leq 0\} (= \operatorname{gph} \mathcal{S})$. We have $\operatorname{int}(\operatorname{dom} \mathcal{S}) = \operatorname{rint}(\operatorname{dom} \mathcal{S}) = \operatorname{rint}(\mathsf{P}_p \mathcal{C}) = \mathsf{P}_p(\operatorname{rint} \mathcal{C})$, where the first equality follows from $\operatorname{int}(\operatorname{dom} \mathcal{S})$ being nonempty, the second by the definition of the domain of a set-valued mapping and the third by [51, Prop. 2.44]. Using the well-known representation for the relative interior of polyhedral sets this means that $\bar{p} \in \operatorname{int}(\operatorname{dom} \mathcal{S})$ if and only if there exists a $\bar{z}$ such that $\bar{z} \in \mathcal{Z}(\bar{p})$ and $g(z) < 0$, i.e., the Slater condition holds for $\mathbb{P}(\bar{p})$, which in turn is equivalent to the boundedness of $\mathcal{Y}^\star(\bar{p})$ and implies (41), [22, Ex. 5.3.1], [32, Lem. 1]. ■

Proof of Proposition 14: Let $\gamma_s = \min_{i \in \mathbb{N}_{[1,m]}} \{-g_i(z^s)\}$. Any $\hat{p} \in \mathcal{P}_S$ can be expressed as $\hat{p} = \sum_{s=1}^S \alpha_s p^s$, for some $\alpha_s \geq 0$ such that $\sum_{s=1}^S \alpha_s = 1$. Let $\hat{z} = \sum_{s=1}^S \alpha_s z^s$. Then since $\operatorname{gph} \mathcal{Z}$ is an affine subspace, $(\hat{p}, \hat{z}) \in \operatorname{gph} \mathcal{Z}$, i.e., $\hat{z} \in \mathcal{Z}(\hat{p})$. Furthermore, for any $i \in \mathbb{N}_{[1,m]}$, $g_i(\hat{z}) + \gamma \leq \sum_{s=1}^S \alpha_s (g_i(z^s) + \gamma_s) \leq 0$, the first inequality being valid due to convexity of each component of g and the choice of γ [cf. (42a)]. Therefore $\hat{z}$ is a Slater vector for $\mathbb{P}(\hat{p})$. Thus,

$$\begin{aligned}\|y^\star(\hat{p})\| &\leq \gamma^{-1}\left(V(\hat{z}) - V^\star(\hat{p})\right) \\ &\leq \gamma^{-1}\left(\sum_{s=1}^S \alpha_s V(z^s) - V^\star(\hat{p})\right) \leq \gamma^{-1}\beta\end{aligned}$$

where the first inequality is due to Lemma 13, the second follows by convexity of V, and the last by the nonnegativity of $V^\star$ and (42b). ■

Proof of Theorem 19: Since $y^\star_{\min}(p) \in \mathcal{Y}^\star(p)$, it must satisfy the KKT conditions for (48), which are given by (49a). Using Lemma 18, $y^\star_{\min}(p)$ is the solution of the following strictly convex quadratic program:

$$\min_y \quad \frac{1}{2}\|y\|^2 \tag{60a}$$
$$\text{s.t.} \quad Hy = Hy^\star \tag{60b}$$
$$y_i = 0,\ i \in N(p),\ y_i \geq 0,\ i \notin N(p) \tag{60c}$$

where $y^\star$ is any optimal solution. According to the KKT conditions for (60), $y^\star_{\min}(p)$ must be primal feasible [cf. (60b), (60c)] and there must exist a $\lambda \in \mathbb{R}^m$ such that (49b) holds. Notice that one can use $y^\star = y^\star_{\min}(p) \in \mathcal{Y}^\star(p)$ in (60b) making it redundant. This concludes the proof. ■

Proof of Lemma 20: Since

$$\min\left\{\frac{1}{2}\|y\|^2 | y \in \mathcal{Y}^\star_{\hat{N}}(p)\right\} \tag{61}$$

is a strictly convex QP and $\mathcal{Y}^\star_{\hat{N}}(p)$ is nonempty [indeed $\hat{y} \in \mathcal{Y}^\star_{\hat{N}}(p)$], it has a unique optimal solution $y \in \mathbb{R}^m$ which must satisfy the KKT conditions, i.e., feasibility and

$$(y_i + H_{i.}\lambda)y_i = 0,\ y_i + H_{i.}\lambda \geq 0,\ y_i \geq 0,\ i \notin \hat{N}. \tag{62}$$

By definition $\hat{y}$ is feasible and $\hat{y}_i > 0$, $i \notin \hat{N}$. Therefore, $\hat{y}$ is optimal for (61) if and only if $\hat{y}_i + H_{i.}\lambda = 0$, for every $i \notin \hat{N}$, which can be equivalently written as $\hat{y}_i(\hat{y}_i + H_{i.}\lambda) = 0$, $i \in \mathbb{N}_{[1,m]}$, since $\hat{y}_i = 0$, $i \in \hat{N}$. ■

Proof of Theorem 21: By Lemma 20, $\mathcal{Y}^\star_{\min}(p)$ is the set of all $y \in \mathcal{Y}^\star(p)$ that are minimum-norm optimal solutions along some face of $\mathcal{Y}^\star(p)$. Since $\mathcal{Y}^\star(p)$ is a polyhedral set, it has a finite number of faces. This means that the set $\mathcal{Y}^\star_{\min}(p)$ is finite and therefore bounded. We next show that $y^\star_{\min}(p) \in \mathcal{Y}^\star_{\min}(p)$ proving that $\mathcal{Y}^\star_{\min}(p)$ is nonempty. According to Theorem 19 there exists a $\lambda \in \mathbb{R}^m$ such that $y^\star_{\min,i}(p)(y^\star_{\min,i}(p) + H_i.\lambda) = 0$, $y^\star_{\min,i}(p) + H_i.\lambda \geq 0$, $i \notin N(p)$, while for $i \in N(p)$ one has $y^\star_{\min,i}(p) = 0$, therefore $y^\star_{\min,i}(p)(y^\star_{\min,i}(p) + H_i.\lambda) = 0$ for $i \in N(p)$ as well. This also implies that $\min\{(1/2)\|y\|^2 | y \in \mathcal{Y}^\star_{\min}(p)\} \leq (1/2)\|y^\star_{\min}(p)\|^2 = \min\{(1/2)\|y\|^2 | y \in \mathcal{Y}^\star(p)\}$. On the other hand, since $\mathcal{Y}^\star_{\min}(p) \subseteq \mathcal{Y}^\star(p)$, it follows that $\min\{(1/2)\|y\|^2 | y \in \mathcal{Y}^\star(p)\} \leq \min\{(1/2)\|y\|^2 | y \in \mathcal{Y}^\star_{\min}(p)\}$, proving (53). ■

ACKNOWLEDGMENT

The authors express their gratitude to P. Sopasakis for implementing GPAD in C. The authors also thank the Associate Editor and the anonymous reviewers for their valuable comments and suggestions for improving the original manuscript.

REFERENCES

[1] D. Mayne and J. Rawlings, *Model Predictive Control: Theory and Design*. Madison, WI, USA: Nob Hill, 2009.

[2] J. Maciejowski, *Predictive Control With Constraints*. Harlow, U.K.: Prentice-Hall, 2002.

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

[4] N. Ricker, "Use of quadratic programming for constrained internal model control," *Ind. Eng. Chem. Process Des. Dev.*, vol. 24, no. 4, pp. 925–936, 1985.

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

[6] C. Schmid and L. Biegler, "Quadratic programming methods for reduced Hessian SQP," *Comput. Chem. Eng.*, vol. 18, no. 9, pp. 817–832, 1994.

[7] J. Mattingley and S. Boyd, "CVXGEN: A code generator for embedded convex optimization," *Optim. Eng.*, vol. 13, no. 1, pp. 1–27, Mar. 20102.

[8] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.

[9] A. Domahidi, A. Zgraggen, M. Zeilinger, M. Morari, and C. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control," in *Proc. 51st IEEE Conf. Decision and Control*, 2012, pp. 668–674.

[10] 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.

[11] 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.

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

[13] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Dordrecht, The Netherlands: Kluwer Academic, 2004.

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

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

[16] S. Richter, C. Jones, and M. Morari, "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method," *IEEE Trans. Autom. Control*, vol. 57, no. 6, pp. 1391–1403, Jun. 2012.

[17] 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. Decision and Control and Eur. Control Conf.*, Orlando, FL, USA, 2011, pp. 5223–5229.

[18] M. Kögel and R. Findeisen, "A fast gradient method for embedded linear predictive control," in *Proc. 18th IFAC World Congr.*, 2011, vol. 18, no. 1, pp. 1362–1367.

[19] M. Kögel and R. Findeisen, "Fast predictive control of linear systems combining Nesterov's gradient method and the method of multipliers," in *Proc. 50th IEEE Conf. Decision and Control and Eur. Control Conf.*, Orlando, FL, USA, 2011, pp. 501–506.

[20] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for linear model predictive control," in *Proc. 51st IEEE Conf. Decision and Control*, Maui, HI, USA, Dec. 2012, pp. 662–667.

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

[22] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

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

[24] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA, USA: Athena Scientific, 2003.

[25] D. Bertsekas, *Convex Optimization Theory*. Belmont, MA, USA: Athena Scientific, 2009.

[26] P. Tseng, On accelerated proximal gradient methods for convex-concave optimization Dept. Math., Univ. Washington, Seattle, WA, USA, 2008, Tech. Rep.

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

[28] O. Devolder, F. Glineur, and Y. Nesterov, "Double smoothing technique for large-scale linearly constrained convex optimization," *SIAM J. Optim.*, vol. 22, no. 2, pp. 702–727, 2012.

[29] P. Giselsson, "Execution time certification for gradient-based optimization in model predictive control," in *Proc. 51st IEEE Conf. on Decision and Control*, Maui, HI, USA, Dec. 2012, pp. 3165–3170.

[30] C. Rao, S. Wright, and J. Rawlings, "Application of interior-point methods to model predictive control," *J. Optim. Theory Appl.*, vol. 99, no. 3, pp. 723–757, 1998.

[31] L. Vandenberghe, S. Boyd, and M. Nouralishahi, Robust linear programming and optimal control Dept. Elect. Eng., Univ. California, Los Angeles, CA, USA, 2002, Tech. Rep..

[32] A. Nedic and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1757–1780, 2009.

[33] A. Zhang and S. Hayashi, "Celis–Dennis–Tapia based approach to quadratic fractional programming problems with two quadratic constraints," *Numer. Alg., Control Optim.*, vol. 1, no. 1, pp. 83–98, 2011.

[34] P. Pardalos and S. Vavasis, "Quadratic programming with one negative eigenvalue is NP-hard," *J. Global Optim.*, vol. 1, no. 1, pp. 15–22, 1991.

[35] P. Pardalos, "Global optimization algorithms for linearly constrained indefinite quadratic problems," *Comput. Math. Appl.*, vol. 21, no. 6–7, pp. 87–97, 1991.

[36] I. Androulakis, C. Maranas, and C. Floudas, "αbb: A global optimization method for general constrained nonconvex problems," *J. Global Optim.*, vol. 7, no. 4, pp. 337–363, 1995.

[37] H. Sherali and C. Tuncbilek, "A reformulation-convexification approach for solving nonconvex quadratic programming problems," *J. Global Optim.*, vol. 7, no. 1, pp. 1–31, 1995.

[38] S. Burer and D. Vandenbussche, "A finite branch-and-bound algorithm for nonconvex quadratic programming via semidefinite relaxations," *Math. Program.*, vol. 113, no. 2, pp. 259–282, 2008.

[39] H. Tuy, *Convex Analysis and Global Optimization*. Dordrecht, The Netherlands: Kluwer Academic, 1998.

[40] 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.

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

[42] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[43] J. Júdice, "Algorithms for linear programming with linear complementarity constraints," *TOP*, pp. 1–22, 2011.
[44] J. Hu, J. Mitchell, J. Pang, K. Bennett, and G. Kunapuli, "On the global solution of linear programs with linear complementarity constraints," *SIAM J. Optim.*, vol. 19, no. 1, pp. 445–471, 2008.
[45] *"Gurobi Optimizer Reference Manual"* Gurobi Optimization, Inc., 2012 [Online]. Available: http://www.gurobi.com
[46] *"IBM ILOG CPLEX Optimization Studio 12.4—User Manual,"* IBM, Inc., 2012.
[47] A. Bemporad, M. Morari, and N. Ricker, *Model Predictive Control Toolbox for Matlab—User's Guide*. : The Mathworks, Inc., 2004 [Online]. Available: http://www.mathworks.com/access/helpdesk/help/toolbox/mpc/
[48] P. Giselsson, "Optimal preconditioning and iteration complexity bounds for gradient-based optimization in model predictive control," in *Proc. 2013 Amer. Control Conf.*, 2013.
[49] M. Rubagotti, P. Patrinos, and A. Bemporad, "Stabilizing embedded MPC with computational complexity guarantees," in *Proc. 12th Eur. Control Conf. (ECC)*, Zurich, Switzerland, Jul. 2013.
[50] P. Patrinos, A. Guiggiani, and A. Bemporad, "Fixed-point dual gradient projection for embedded model predictive control," in *Eur. Contr. Conf.*, Zurich, Switzerland, Jul. 2013, pp. 3602–3607.
[51] R. Rockafellar and R. J. Wets, *Variational Analysis*. New York, NY, USA: Springer Verlag, 2009.

Panagiotis Patrinos received the M.Eng. degree in chemical engineering, the M.Sc. degree in applied mathematics, and the Ph.D. degree in control and optimization, all from the National Technical University of Athens, Athens, Greece, in 2003, 2005, and 2010, respectively.

He is currently an Assistant Professor at the IMT Institute for Advanced Studies, Lucca, Italy. Previously, he was a Postdoctoral Fellow at IMT Lucca and at the University of Trento. His current research interests are focused on devising efficient algorithms for distributed and nonsmooth optimization with applications in embedded model predictive control (MPC) and machine learning. He is also interested in stochastic, risk-averse and distributed MPC with applications in the energy and power systems domain.

Alberto Bemporad (S'95–M'99–SM'06–F'10) received the master's degree in electrical engineering in 1993 and the Ph.D. degree in control engineering in 1997 from the University of Florence, Florence, Italy.

He spent the academic year 1996/97 at the Center for Robotics and Automation, Department of Systems Science and Mathematics, Washington University, St. Louis, MO, USA, as a Visiting Researcher. In 1997–1999 he held a postdoctoral position at the Automatic Control Laboratory, ETH Zurich, Switzerland, where he collaborated as a Senior Researcher in 2000–2002. In 1999–2009 he was with the Department of Information Engineering, University of Siena, Italy, becoming an Associate Professor in 2005. In 2010–2011 he was with the Department of Mechanical and Structural Engineering, University of Trento, Italy. In 2011 he joined as a Full Professor at the IMT Institute for Advanced Studies Lucca, Italy, where he became the Director in 2012. He cofounded the spinoff company ODYS S.r.l. He has published more than 250 papers in the areas of model predictive control, hybrid systems, automotive control, multiparametric optimization, computational geometry, robotics, and finance. He is author or coauthor of various Matlab toolboxes for model predictive control design, including the Model Predictive Control Toolbox (The Mathworks, Inc.).

Dr. Bemporad was an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL during 2001–2004 and Chair of the Technical Committee on Hybrid Systems of the IEEE Control Systems Society in 2002–2010. He has been an IEEE Fellow since 2010.