Computational complexity certification for embedded MPC based on dual gradient method

Ion Necoara

Automatic Control and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania

Abstract

In this paper we analyze the computational complexity of a linear model predictive control (MPC) scheme for embedded systems based on the dual gradient algorithm. We recast our MPC problem as a linearly constrained convex problem. When it is difficult to project on the primal feasible set described by linear constraints, we use the Lagrangian relaxation to handle the complicated constraints and then, we apply the dual gradient algorithm for solving the corresponding dual. We give a unified convergence analysis for the dual gradient algorithm: we provide sublinear or linear estimates on the primal suboptimality and feasibility violation of the generated approximate primal solutions. Our analysis relies on the Lipschitz property of the dual function or an error bound property. Furthermore, the iteration complexity analysis is based on two types of approximate primal solutions: an average primal sequence or the last primal iterate sequence. We also discuss implementation aspects of the proposed algorithm on constrained linear MPC for embedded systems.

Keywords: Dual gradient algorithm; linear/sublinear convergence rate; computational complexity certification; model predictive control; embedded systems.

1. Introduction

Model predictive control (MPC) is one of the most successful advanced control technologies implemented in industry due to its ability to handle complex systems with hard input and state constraints. MPC requires the solution of an optimal control problem at every sampling instant at which new state information becomes available. Recently, embedded MPC has been widely used in many applications and its usage in industrial plants has increased concurrently. The concept behind embedded MPC is to design a control scheme that can be implemented on autonomous electronic hardware, e.g. programmable logic controller [1, 2], micro-controller circuit board [3, 4] or field-programmable gate arrays [5]. Such devices vary widely in both computational power and memory storage capabilities as well as cost. As a result, there has been a growing focus on developing faster MPC schemes, improving the computational efficiency [6, 7, 8] and providing worst-case computational complexity guarantees for the applied solution methods [9, 10, 11, 4, 12], making these schemes feasible for implementation on hardware with limited computational power.

For fast embedded systems [13, 5, 14, 3] the sampling times are very short, such that any iterative optimization algorithm must offer tight bounds on the total number of iterations which have to be performed in order to provide a desired optimal controller. Even if second order methods (e.g. interior point methods) can offer fast rates of convergence in practice, the worst case complexity bounds are high [15, 7]. Further, these methods have complex iterations, involving inversion of matrices, which are difficult to implement on embedded systems, where the units demand simple computations. Therefore, first order methods (e.g. gradient methods) are more suitable in these situations [16, 9, 11, 17, 12, 4].

In linear MPC for embedded systems, the corresponding optimal control problem can be recast as a convex quadratic optimization problem with linear constraints. If the constraints are simple, then we can use gradient type methods for solving the primal problem as in [4]. When the projection on the primal feasible set is hard to compute, e.g. MPC problem in condensed form, an alternative to primal gradient methods is to use the Lagrangian relaxation to handle the complicated constraints and then to apply dual gradient algorithms for solving the dual. The computational complexity certification of gradient-based methods for solving the (augmented) Lagrangian dual of a primal convex problem is studied e.g. in [18, 19, 16, 6, 9, 10, 17, 11, 20, 12]. In [9] the authors present a general framework for dual (fast) gradient methods with inexact oracle, i.e. only approximate information is available for the values of the dual function and its gradient, and provide estimates for primal suboptimality and infeasibility in an average point. Extension of this framework to augmented Lagrangian methods are given in [10]. In [19, 12, 21] dual fast gradient methods are proposed for solving convex programs with linear constraints and estimates on primal suboptimality and infeasibility are provided based on averaging. In [16] an embedded MPC scheme is derived based on combining Nesterov’s fast gradient method and the method of multipliers. In [6] a dual method based on fast gradient schemes and smoothing tech-
niques of the ordinary Lagrangian is presented and convergence rate in an averaging primal sequence is derived. However, in all these papers the authors provide an approximate primal solution based on averaging. There are very few attempts to derive iteration complexity of dual gradient based methods using as an approximate primal solution the last iterate of the algorithm\cite{11,22}. For example, in\cite{11} the authors show that, for quite general linearly constrained convex problems, the dual problem has an error bound property and then they prove linear convergence of dual gradient method, using the last primal iterate as an approximate primal solution. For dual fast gradient method, rate of convergence in the last iterate is provided in\cite{22}, but no results on dual gradient algorithm are given. Moreover, in practice it was observed that gradient type methods are converging faster in the last iterate than in an average sequence. These issues motivate our work here.

**Contributions.** In this paper we provide a unified analysis of the dual gradient algorithm producing approximate primal feasible and optimal solutions, used for solving linear embedded MPC problems. Our analysis is based on the Lipschitz property of the dual function or an error bound property. Further, the iteration complexity analysis is based on two types of approximate primal solutions: an average primal sequence or the last primal iterate sequence. We prove that the gradient algorithm for solving the dual problem has the following iteration complexity in terms of primal suboptimality and infeasibility: for strongly convex primal functions we show a sublinear convergence rate in both, an average primal sequence (convergence rate $O(1/k)$, with $k$ iteration counter) or the last primal iterate sequence (convergence rate $O(1/\sqrt{k})$); if additionally, the function has also Lipschitz gradient, then we show that the dual problem has an error bound property and that dual gradient method converges linearly in the last primal iterate sequence (convergence rate $O(\theta^k)$, with $\theta < 1$), a result which appears to be new in this area. To certify the complexity of the proposed method, we apply the algorithm on MPC problems for embedded systems.

**Paper outline.** In Section 2 we formulate a linear MPC problem, while in Section 3 we define our problem of interest and derive some important preliminary results. Then, in Section 4 we prove sublinear rate of convergence for dual gradient method, while in Section 5 we prove linear convergence for this algorithm provided that some error bound property holds for dual problem. Lastly, numerical issues are discussed in Section 6.

**Notation.** We work in the space $\mathbb{R}^n$ composed by column vectors. For $u,v \in \mathbb{R}^n$ we denote the standard Euclidean inner product $\langle u,v \rangle = u^T v$, the Euclidean norm $||u|| = \sqrt{\langle u,u \rangle}$, and the projection onto non-negative orthant $\mathbb{R}^+_n$ as $|u|_+$. Further, given a norm $|| \cdot ||$ on $\mathbb{R}^n$, we denote its dual norm as $||u||^*_n = \max_{x,y \in \mathbb{R}^n} \langle x,y \rangle$. From this definition it follows immediately the well-known Cauchy-Schwartz inequality: $\langle x,y \rangle \leq ||x||^*_n \cdot ||y||_n$, for all $x, y \in \mathbb{R}^n$. Recall that the dual norm of 1-norm (|| \cdot ||_1) is the infinity-norm (|| \cdot ||_\infty). Moreover, for a matrix G $\in \mathbb{R}^{m \times n}$ and two vector norms, $|| \cdot ||_p$ on $\mathbb{R}^n$ and $|| \cdot ||_q$ on $\mathbb{R}^m$, we define the induced matrix norm as follows: $|G|_{p,q} = \max_{||x||_q \leq 1} ||Gx||_p$. As a consequence we have the following inequality: $||G||_p \leq |G|_{1,\infty} ||x||_\infty$ for all $x \in \mathbb{R}^n$. We use the notation $|G|_{2,2} = ||G||$. We denote with $e \in \mathbb{R}^n$ the vector with all entries 1.

### 2. Motivation: MPC problems for linear systems

We consider discrete-time systems, defined by the following linear difference equations:

$$x(t+1) = Ax(t) + Bu(t),$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ represent the state and the input of the system at time $t$, respectively. We also impose local state and input constraints:

$$x(t) \in X, \quad u(t) \in U \quad \forall t \geq 0,$$

where $X \subseteq \mathbb{R}^n$ and $U \subseteq \mathbb{R}^m$ are polyhedral sets. For the system (1) we consider a quadratic convex stage cost:

$$J(x(t), u(t)) = \frac{1}{2} ||x(t)||^2_2 + \frac{1}{2} ||u(t)||^2_2,$$

where $||x||^2_2 = x^T Q x$. Typically, the pair $(A, B)$ is stabilizable, $Q \geq 0, R > 0$ and there exists a matrix $C \in \mathbb{R}^{m \times n}$, such that $Q = C^T C$ and the pair $(A, C)$ is detectable. Then, we denote with $K \in \mathbb{R}^{n \times n}$ the gain associated with the infinite horizon linear quadratic regulator (LQR) defined by the matrices $A, B, Q$ and $R$ and with $P$ the solution of the algebraic Riccati equation associated with the LQR problem. We also introduce a terminal cost $V^T(x) = \frac{1}{2} ||x||_P^2$, and a terminal polyhedral set $X^T$, which we consider to be invariant for the closed-loop system $x(t+1) = (A + BK)x(t)$ (see e.g., [17, 23] for a detailed discussion). For a prediction horizon of length $N$, the MPC problem for (1), with a given initial state $x \in X_N$, where $X_N$ denotes a region of attraction, can be formulated as [17]:

$$V^T(x) = \min_{x(0)=x(t)} \sum_{t=0}^{N-1} J(x(t), u(t)) + V^T(x(N))$$

s.t. \begin{align}
& x(t+1) = Ax(t) + Bu(t), \quad x(0) = x \\
& x(t) \in X, \quad u(t) \in U, \quad x(N) \in X^T \quad \forall t = 0, \ldots, N-1.
\end{align}

For the input trajectory of the system we use the notation:

$$u = [u(0)^T \ldots u(N-1)^T]^T \in \mathbb{R}^m.$$

By eliminating the states from the dynamics (1), the MPC problem (3) can be expressed in condensed form as a quadratic convex problem:

$$V^T(x) = \min_{u(x)} \sum_{t=0}^{N-1} J(x(t), u(t))$$

s.t. \begin{align}
& (u, x) \quad \Rightarrow \quad 1 - \frac{1}{2} u^T Q x + (W x)^T u \\
& G u + E x + g \leq 0, \quad (Q P(x))
\end{align}

where $Q$ is a positive definite matrix due to the assumption that $R$ is positive definite and the inequalities $G u + E x + g \leq 0$ are obtained by eliminating the input and states from the constraints $u(t) \in U, x(t) \in X$ and $x(N) \in X^T$. More precisely, we can express the state vector $x = [x(0)^T \ldots x(N)^T]^T$ over the entire prediction horizon as:

$$x = Ax + Bu.$$
where matrices $A$ and $B$ are:

$$A = \begin{bmatrix} I_n \\ A \\ A^2 \\ \vdots \\ A^{N-1} \\ A^{N} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ A & B & 0 & \ldots & 0 \\ AB & B & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ A^{N-2}B & A^{N-3}B & \ldots & B & 0 \\ A^{N-1}B & A^{N-2}B & \ldots & AB & B \end{bmatrix}$$

In this case we have a linearly constrained QP with dense matrices in the objective function:

$$Q = B^T D_Q B + D_R, \quad W = B^T D_Q A,$$

where $D_Q = \begin{bmatrix} I_N \otimes Q & 0 \\ 0 & P \end{bmatrix}$ and $D_R = I_N \otimes R$. With this formulation, $G \in \mathbb{R}^{n \times n}$ is a lower block triangular matrix. In MPC, at each time instant, given the initial state $x \in X_n$, we need to solve approximately the optimization problem (3) or equivalently optimization problem (QP(x)). Moreover, in the context of fast embedded systems we need to provide efficient numerical optimization algorithms for finding an approximate solution, with computational complexity certification and low memory. In the next sections we show that the dual gradient algorithm satisfies these requirements.

3. Problem formulation

In this paper we study the following linearly constrained convex optimization problem:

$$F^* = \min_{u \in \mathbb{R}^n} \{ F(u) : \text{Gu + g} \leq 0 \},$$

where $F : \mathbb{R}^n \to \mathbb{R}$ is a convex function, $G \in \mathbb{R}^{n \times n}$, and $g \in \mathbb{R}^n$. We note that problem (QP(x)) is a particular case of problem (5) with $F$ being a convex quadratic function. Note that our framework allows us to also consider equality constraints, but for simplicity we omit them here. The following assumption holds throughout the paper:

**Assumption 3.1.** The primal objective function $F$ is a $\sigma_F$-strongly convex function (with $\sigma_F > 0$). Moreover, the polyhedral feasible set is nonempty.

Since we assume strong convexity on $F$, primal problem (5) has a unique optimal solution $u^*$. We can notice that since $G u + g \leq 0$ (also called complicating constraints) is a general polyhedron, the projection on this set is hard to compute. Therefore, solving primal problem (5) approximately with primal projected gradient type methods is numerically difficult and thus we usually use dual gradient methods for finding an approximate solution for (5) (see e.g. [22, 9, 11, 17, 16, 12]). By moving the complicating constraints into the cost via Lagrange multipliers we define the dual function:

$$d(\lambda) = \min_{u \in \mathbb{R}^n} \mathcal{L}(u, \lambda),$$

where $\mathcal{L}(u, \lambda) = F(u) + \langle \lambda, G u + g \rangle$ denotes the Lagrangian w.r.t. the complicating constraints $G u + g \leq 0$. We also denote by $u(\lambda)$ the optimal solution of the inner problem:

$$u(\lambda) = \arg \min_{u \in \mathbb{R}^n} \mathcal{L}(u, \lambda).$$

If we cannot solve the inner optimization problem (7) exactly, but with some inner accuracy, then our framework allows us also to use inexact gradients and approximate values of the dual function $d$. For simplicity, we do not tackle this subject in this paper but for a full discussion on this topic see [9, 10, 17]. Since we have only linear constraints, strong duality holds [24] and thus the outer problem satisfies:

$$F^* = \max_{d(\lambda)} \mathcal{R}(\lambda) = \min_{\lambda \in \Lambda^*} ||\lambda^* - \lambda||.$$  

3.1. Preliminary results

In this section we derive some important relations between the optimal solution of the inner problem $u(\lambda)$ and the dual function $d(\lambda)$ that will be used in the subsequent derivations. Since $F$ is strongly convex (Assumption 3.1), it can be proved that the gradient of the dual function $d(\lambda)$ is $L_d$-Lipschitz continuous and given by (see [9] for more general settings):

$$\nabla d(\lambda) = G u(\lambda) + g, \quad L_d = \frac{||G||^2}{\sigma_F}.$$  

From Assumption 3.1, which implies Lipschitz gradient for the dual function, we can derive bounds on $d$ in terms of a linear and a quadratic model (the so called descent lemma) [9]:

$$\forall \lambda, \mu \in \mathbb{R}^n, \quad 0 \geq d(\mu) - [d(\lambda) + \langle \nabla d(\lambda) , \mu - \lambda \rangle] \geq -\frac{L_d}{2} ||\mu - \lambda||^2.$$  

**Lemma 3.2.** Under Assumption 3.1, the following inequality holds:

$$\frac{\sigma_F}{2} ||\lambda(\lambda) - u^*||^2 \leq F^* - d(\lambda) \quad \forall \lambda \in \mathbb{R}^n.$$  

**Proof:** First, let us recall the following relations:

$$d(\lambda) = F(u(\lambda)) + \langle \lambda, G u(\lambda) + g \rangle \quad \text{and} \quad \nabla d(\lambda) = G u(\lambda) + g.$$  

Moreover, for any $\lambda \geq 0$ we have $\langle \lambda, \nabla d(\lambda^*) \rangle \leq 0$. Since $F$ is $\sigma_F$-strongly convex, it follows that $\mathcal{L}(u, \lambda)$ is also $\sigma_F$-strongly convex in the variable $u$, which gives the following inequalities:

$$\frac{\sigma_F}{2} ||\lambda(\lambda) - u^*||^2 \leq \mathcal{L}(u^*, \lambda) - \mathcal{L}(u(\lambda), \lambda) = F(u^*) + \langle \lambda, \nabla d(\lambda^*) \rangle - d(\lambda) \leq F^* - d(\lambda),$$  

valid for all $\lambda \in \mathbb{R}^n$, where we used that $\mathcal{L}(u(\lambda), \lambda) = d(\lambda)$.
We now express the primal feasibility violation in terms of \(\|u(\lambda) - u^*\|\) for any \(\lambda \in \mathbb{R}_+^p\). Indeed, let us note that (see also [22] for a similar derivation):

\[
\|G u(\lambda) + g - g^* - g\|_\infty = \|G(u(\lambda) - u^*)\|_\infty \\
\leq \|G\|_{2,\infty}\|u(\lambda) - u^*\|. \tag{11}\]

Using the properties of the \(\infty\)-norm we get:

\[
G u(\lambda) + g \leq \|G\|_{2,\infty}\|u(\lambda) - u^*\| \epsilon + (G u^* + g). \tag{12}
\]

We now express the primal suboptimality in terms of \(\|u(\lambda) - u^*\|\), a result which appears to be new:

**Lemma 3.3.** Under Assumption 3.1, the following inequality holds for all \(\lambda \in \mathbb{R}_+^p\) and \(\lambda^* \in \Lambda^*\):

\[
|F(u(\lambda)) - F^*| \leq (\|\lambda - \lambda^*\| + \|\lambda^*\|)\|G\|_{2,\infty}\|u(\lambda) - u^*\|. \tag{13}
\]

**Proof:** Firstly, using the complementarity condition \(\langle \lambda^*, Gu^* + g \rangle = 0\) valid for any \(\lambda^* \in \Lambda^*\) we get:

\[
\langle \lambda^*, Gu^* + g \rangle + F(u^*) = d(\lambda^*) \\
= \min_{u \in \mathbb{R}^p} F(u) + \langle \lambda^*, Gu + g \rangle \\
\leq F(u(\lambda)) + \langle \lambda^*, Gu(\lambda) + g \rangle,
\]

which leads to the following relation:

\[
F(u(\lambda)) - F^* \geq \langle \lambda^*, Gu^* + g - Gu(\lambda) - g \rangle.
\]

Using the Cauchy-Schwartz inequality we derive:

\[
F(u(\lambda)) - F^* \geq -\|\lambda\|\|G u^* + g - Gu(\lambda) - g\|_\infty \\
(\text{valid for all } \lambda \in \mathbb{R}_+^p, \text{where in the first inequality we used concavity of dual function } d, \text{in the second inequality the relation } \nabla d(\lambda) = Gu(\lambda) + g \text{ and Cauchy-Schwartz inequality and in the third inequality relation (11). In conclusion, using the triangle inequality for norms, we obtain the following inequality:})
\]

\[
F(u(\lambda)) - F^* \leq (\|\lambda - \lambda^*\| + \|\lambda^*\|)\|G\|_{2,\infty}\|u(\lambda) - u^*\|. \tag{15}
\]

valid for all \(\lambda \in \mathbb{R}_+^p\) and \(\lambda^* \in \Lambda^*\). Now, combining (14) and (15) we obtain a bound on primal suboptimality (13).

Note that, based on our derivations from above, we are able to characterize primal suboptimality (13) without assuming any Lipschitz property on \(F\) as opposed to the results in [22] where the authors had to require Lipschitz continuity of \(F\) for providing estimates on primal suboptimality. E.g., for linear MPC the objective function is quadratic and thus it is not Lipschitz continuous, so that our theory covers this important case.

### 3.2. Dual gradient algorithm

We now propose the following dual gradient algorithm for finding an approximate solution for primal problem (5):

**Algorithm (DG)(\(\lambda^0\))**

Given \(\lambda^0 \in \mathbb{R}_+^p\), for \(k \geq 0\) compute:

1. \(u^k = \arg \min_{u \in \mathbb{R}^p} L(u, \lambda^k)\)
2. \(\lambda^{k+1} = \left[\lambda^k + \frac{1}{\Lambda} \nabla d(\lambda^k)\right]^+\),

where we recall that \(\nabla d(\lambda^k) = Gu^k + g\). Let us mention some important property of the gradient method, which holds for all \(k \geq 0\) and \(\lambda \in \mathbb{R}_+^p\), that will be useful in the following sections (see [9, eq. (18)], [15]):

\[
\|\lambda^{k+1} - \lambda\|^2 \leq \|\lambda^k - \lambda\|^2 + \frac{2}{\Lambda} \left(d(\lambda^{k+1}) - d(\lambda^k) - \langle \nabla d(\lambda^k), \lambda - \lambda^k \rangle\right). \tag{16}
\]

Now, if we take \(\lambda = \lambda^*\) in (16) and use concavity of \(d\), we get that

\[
\|\lambda^{k+1} - \lambda^*\|^2 \leq \|\lambda^k - \lambda^*\|^2 + \frac{2}{\Lambda} \left(d(\lambda^{k+1}) - d(\lambda^k)\right) \leq \|\lambda^k - \lambda^*\|^2.
\]

Thus, we obtain:

\[
\|\lambda^k - \lambda^*\| \leq \|\lambda^0 - \lambda^*\| \quad \forall k \geq 0, \quad \lambda^* \in \Lambda^*. \tag{17}
\]

Moreover, if we take \(\lambda = \lambda^k\) in (16), then we get that the dual gradient algorithm is an ascent method (see also [11]):

\[
d(\lambda^{k+1}) \geq d(\lambda^k) + \frac{Ld}{2}\|\lambda^{k+1} - \lambda^k\| \quad \forall k \geq 0. \tag{18}
\]

Finally, if we take \(\lambda = 0\) in (16), using that \(d(\lambda^{k+1}) \leq F^*\) and that \(F(u^k) = d(\lambda^k) - \langle \nabla d(\lambda^k), \lambda^k \rangle\), then we get:

\[
\|\lambda^{k+1}\|^2 \leq \|\lambda^k\|^2 + \frac{2}{\Lambda} (F^* - F(u^k)) \quad \forall k \geq 0. \tag{19}
\]

In the next sections we analyze the iteration complexity of Algorithm (DG) in terms of primal suboptimality and feasibility violation of the generated approximate primal solutions. Furthermore, our iteration complexity analysis is based on two types of approximate primal solutions: the last primal iterate sequence \((u^k)_{k \geq 0}\) or an average primal sequence of the form \(\bar{u}^k = \frac{1}{k+1} \sum_{j=0}^k u^j\).
4. Sublinear convergence of Algorithm (DG)

In this section we prove sublinear convergence of Algorithm (DG) based on the Lipschitz gradient property of the dual function. Based on the descent lemma we can derive the well-known sublinear convergence rate of Algorithm (DG) in terms of dual suboptimality.

Theorem 4.1. [15] Let Assumption 3.1 hold and sequence \( \{\lambda_k\}_{k \geq 0} \) be generated by Algorithm (DG). Then, a sublinear estimate on dual suboptimality for dual problem (8) is:

\[
F^* - d(\hat{x}) \leq \frac{L_d R_d^2}{2k},
\]

where \( R_d = \mathcal{R}(\lambda^0) = \min_{\lambda \in \Lambda} ||\lambda^0 - \lambda^*||. \)

4.1. Sublinear convergence in the last primal iterate

In this section we derive sublinear estimates for primal feasibility and primal suboptimality for the last primal iterate sequence \( \{u^k\}_{k \geq 0} \) generated by Algorithm (DG). Let us notice that from the definition of Algorithm (DG) we have \( u^k = u(\lambda^k) \). Firstly, combining (10) and (20) we obtain the following important relation characterizing the distance from the last iterate \( u^k \) to the unique optimal solution \( u^* \) of our original problem (5):

\[
||u^k - u^*|| \leq \frac{L_d R_d^2}{\sqrt{k} \sigma F}.
\]

Secondly, combining the previous relation (21) and (12) we obtain a sublinear estimate for feasibility violation of the last iterate \( u^k \) for Algorithm (DG):

\[
G u^k + \mathcal{g} \leq ||G||_{2,\infty} ||u^k - u^*|| e + (G u^* + \mathcal{g}) \leq ||G||_{2,\infty} \frac{L_d R_d^2}{\sqrt{k} \sigma F} e + (G u^* + \mathcal{g}) \]

\[
\leq ||G||_{2,\infty} \frac{L_d R_d^2}{\sqrt{k} \sigma F} e \leq \sqrt{\frac{||G||^2}{\sigma F} \frac{L_d R_d^2}{k}} \ e = \frac{L_d R_d}{\sqrt{k}} \ e,
\]

where we used that \( ||G||_{2,\infty} \leq ||G|| \) and that \( L_d = ||G||^2/\sigma F \). Finally, we derive a sublinear estimate for primal suboptimality of the last iterate \( u^k \). Combining (21), (13) and (17) we obtain:

\[
|F(u^k) - F^*| \leq \left( ||u^k - u^*||_1 + ||u^*||_1 \right) ||G||_{2,\infty} \ rac{L_d R_d^2}{\sqrt{k} \sigma F} \leq \left( \frac{1}{\sqrt{\mathcal{R}}} ||u^0 - u^*||_1 + ||u^*||_1 \right) ||G||_{2,\infty} \ rac{L_d R_d^2}{\sqrt{k} \sigma F} \leq (2 \frac{1}{\sqrt{\mathcal{R}}} ||u^0 - u^*||_1 + ||u^*||_1) \frac{L_d R_d^2}{\sqrt{k} \sigma F} \leq (2 \frac{1}{\sqrt{\mathcal{R}}} R_d + ||u^*||_1) \frac{L_d R_d}{\sqrt{k}}.
\]

4.2. Sublinear convergence in an average primal sequence

In this section we derive sublinear estimates for primal infeasibility and primal suboptimality for an average primal sequence \( \{\hat{u}^k\}_{k \geq 0} \) generated by Algorithm (DG). Let us define the following sequence:

\[
\hat{u}^k = \frac{1}{k+1} \sum_{j=0}^{k} u^j.
\]

First, given the definition of \( \lambda^{j+1} \) in Algorithm (DG) we get:

\[
\lambda^j + \frac{1}{L_d} \nabla d(\lambda) \leq \lambda^{j+1} \quad \forall j \geq 0.
\]

Taking into account that \( \nabla d(\lambda) = G u^* + \mathcal{g} \), that \( \lambda^0 \geq 0 \) and adding up the above inequality for \( j = 0 \) to \( j = k \), we obtain:

\[
\frac{1}{L_d} \sum_{j=0}^{k} G u^j + \mathcal{g} \leq \lambda^{k+1} - \lambda^0.
\]

Dividing both sides by \( k+1 \) and using the expression for \( \hat{u}^k \) we get:

\[
\hat{G} u^k + \mathcal{g} \leq \frac{L_d}{k+1} (\lambda^{k+1} - \lambda^0).
\]

It remains to bound \( \lambda^{k+1} - \lambda^0 \). But, combining the inequality \( ||\cdot||_{\infty} \leq ||\cdot|| \) and (17) we get:

\[
\lambda^{k+1} - \lambda^0 = \lambda^{k+1} + \lambda^* - \lambda^0 \leq ||\lambda^{k+1} - \lambda^*||_{\infty} + ||\lambda^* - \lambda^0||_{\infty} e \leq (||\lambda^{k+1} - \lambda^*|| + ||\lambda^* - \lambda^0||) e \\
\leq 2 ||\lambda^* - \lambda^0|| e \leq 2 R_d e.
\]

Using this bound in (25) we get the following estimate on feasibility violation:

\[
\hat{G} u^k + \mathcal{g} \leq \frac{2L_dR_d}{k+1} e.
\]

In order to prove estimates for primal suboptimality we first use:

\[
F^* = \min_{\lambda \in \Lambda} F(u) + \langle \lambda^*, G u + \mathcal{g} \rangle \leq F(\hat{u}^k) + \langle \lambda^*, \hat{G} u^k + \mathcal{g} \rangle \leq F(\hat{u}^k) + \frac{2L_dR_d}{k+1} \langle \lambda^*, e \rangle \leq F(\hat{u}^k) + \frac{2L_dR_d}{k+1} ||e||_\infty ||\lambda^*||_1.
\]
Using a similar reasoning as in (23) we get:

$$F(\hat{u}^t) - F^* \geq \frac{2L_d R_d}{k + 1} \sqrt[3]{p R_d + ||\lambda^0||_1}. \tag{27}$$

On the other hand, from (19) we have:

$$||\lambda^{j+1}||^2 \leq ||\lambda^j||^2 + \frac{2}{L_d} (F^* - F(\hat{u}^j)) \quad \forall j \geq 0.$$ Adding up these inequalities for $j = 0$ to $j = k$ we obtain:

$$||\lambda^{k+1}||^2 + \frac{2(k + 1)}{L_d} \sum_{j=0}^{k} \left( \frac{1}{k + 1} (F(\hat{u}^j) - F^*) \right) \leq ||\lambda^0||^2.$$ Using the definition of $\hat{u}^t$ and the convexity of $F$ we get:

$$F(\hat{u}^t) - F^* \leq \frac{L_d ||\lambda^0||^2}{2(k + 1)}. \tag{28}$$

Combining (27) and (29) we also derive bounds on primal suboptimality:

$$-\frac{2L_d R_d}{k + 1} \sqrt[3]{p R_d + ||\lambda^0||_1} \leq F(\hat{u}^{k+1}) - F^* \leq \frac{L_d ||\lambda^0||^2}{2(k + 1)}. \tag{29}$$

Note that the iteration complexity estimates of order $O\left(\frac{1}{\sqrt{k}}\right)$ corresponding to the last iterate sequence $u^k$ (Section 4.1) are inferior to those estimates of order $O\left(\frac{1}{k}\right)$ corresponding to an average primal sequence $\hat{u}^k$ given in this section. However, in practical applications we have observed that the dual gradient algorithm converges faster in the last primal iterate $u^k$ than in the primal average sequence $\hat{u}^k$ (see e.g. [9, 22] and also Section 6 below for more details). However, this does not mean that our analysis is weak, since we can also construct problems which show the behavior predicted by our theory (see also [22]).

**Remark 4.2.** It is important to know that our approach allows us to analyze in the same framework also the convergence rate of dual fast gradient method in an average sequence (see [9]) or in the last primal sequence (see [22]). Due to space limitation we do not tackle this subject in the present paper.

5. Linear convergence of Algorithm (DG)

In this section we prove linear convergence of Algorithm (DG) under the following additional assumption on the primal objective function $F$:

**Assumption 5.1.** The primal objective function $F$ has additionally $L_F$-Lipschitz continuous gradient, with $L_F > 0$. Moreover, there exists $\bar{g}$ such that $G \bar{g} + g < 0$.

Therefore, in this section we assume that the function $F$ is $\sigma_F$-strongly convex (according to Assumption 3.1) and has $L_F$-Lipschitz continuous gradient (according to Assumption 5.1). We proved in [11] that the dual problem satisfies an error bound type property. For completeness, in the next section we briefly review this important result.

5.1. Error bound property of the dual problem

For a smooth convex problem in the form:

$$\min_{y \in D} \psi(y),$$

where $\psi(\cdot)$ is convex function, with Lipschitz continuous gradient, and $D$ is a polyhedral set, we are interested in finding optimal points for this problem, i.e. points $y$ satisfying $y = \arg \min_{y \in D} \psi(y)$) (we use the notation $\{y\}$ for the Euclidean projection of a vector on the set $D$). Typically, in order to show linear convergence for gradient based methods used for solving the above problem, we need to require some nondegeneracy assumption on the problem (e.g. strong convexity) which does not hold for many practical applications (e.g. the dual of an MPC problem). A new line of analysis, that circumvents these difficulties, was developed using the notion of error bound, which estimates the distance to the solution set from an $y \in D$ by the norm of the gradient mapping $\nabla \psi(y) = [y - \nabla \psi(y)]_D - y$. For smooth objective functions of the form $\psi(y) = \psi(G^T y)$, with $\psi(\cdot)$ strongly convex function and $G$ a general matrix, we have proved in [11] a global error bound property, provided that the set $D$ is a general polyhedron.

For our primal objective function $F$ in (5), we denote its conjugate by $\tilde{F}(x) = \max_{u \in \mathbb{R}^n} \langle x, u \rangle - F(u)$. According to Proposition 12.60 in [24], under the Assumption 5.1 the conjugate function $\tilde{F}(x)$ is strongly convex w.r.t. Euclidean norm, with constant $\frac{1}{\sigma_F}$. Note that in these settings our dual function can be written:

$$d(\lambda) = -\tilde{F}(-G^T \lambda) - g^T \lambda. \tag{30}$$

We further denote the gradient mapping by [15]:

$$\nabla^* d(\lambda) = [\lambda + \nabla d(\lambda)]_\Lambda, \lambda \forall \lambda \in \mathbb{R}^n_+. \tag{31}$$

Moreover, from Assumption 5.1 we have that $\Lambda^*$ is nonempty and bounded (see [25, Theorem 2.3.2]). Since the set of optimal Lagrange multipliers is bounded, for any $\lambda \in \mathbb{R}^n_+$ we can define the following finite quantity:

$$T(\lambda) = \max_{\lambda \in \Lambda^*} \|\lambda^* - \lambda\|_W. \tag{32}$$

The following theorem establishes a global error bound like property for our dual problem (8):

**Theorem 5.2.** [11] Let Assumptions 3.1 and 5.1 hold. Then, there exists constant $k$ depending on $T(\lambda)$, such that the following error bound property holds for dual problem (8):

$$||\lambda - \lambda^*|| \leq k (T(\lambda)) ||\nabla^* d(\lambda)|| \forall \lambda \in \mathbb{R}^n_+,$$

where $\lambda^* = [\lambda]_{\Lambda}^*$ (Euclidean projection of $\lambda$ on $\Lambda^*$) and $k(T(\lambda))$ is a quadratic expression in $T(\lambda)$ whose coefficients depend on the data of problem (5).

The main difficulty here is the tight estimation of the constant $k$ in the general case, which will be tackled in a future work. Based on Theorem 5.2, on the fact that dual gradient is an ascent method (see (18)) and that $||\lambda^k - \lambda^*|| \leq ||\lambda^0 - \lambda^*||$ for all $k \geq 1$ (see (17)), in [11] we have proved the linear rate of convergence of Algorithm (DG) in terms of dual suboptimality.
Theorem 5.3. [11] Let Assumptions 3.1 and 5.1 hold and sequence \((\lambda^k)_{k \geq 0}\) be generated by Algorithm (DG). Then, a linear estimate on dual suboptimality for dual problem (8) is:

\[
F^* - d(x^k) \leq \frac{1}{2} \left( \frac{4(1 + \bar{k})}{1 + 4(1 + \bar{k})} \right)^{k-2} \mathcal{R}^2_d,
\]

where we defined \(\bar{k} = \kappa(T_d)\) and \(T_d = \mathcal{T}(\lambda^0) = \max_{x \in X} \|\lambda^0 - \lambda^*\|\).

5.2. Linear convergence in the last primal iterate

In this section we derive linear estimates for primal feasibility and primal suboptimality for the last iterate sequence \((u^k)_{k \geq 0}\) generated by our Algorithm (DG). Recall that we have \(u^k = u(x^0)\). For simplicity of the exposition let us denote:

\[
c_1 = \frac{1}{2} \mathcal{R}^2_d \quad \text{and} \quad \theta = \frac{4(1 + \bar{k})}{1 + 4(1 + \bar{k})}.
\]

Clearly, \(\theta < 1\). From Theorem (5.3) we obtain:

\[
F^* - d(x^k) \leq c_1 \theta^{k-2}.
\]

Combining (10) and (35) we obtain the following relation:

\[
\|u^k - u^*\| \leq \sqrt{\frac{2c_1 \theta^{k-2}}{\sigma_F}}.
\]

Then, combining the previous relation (36) and (12) we obtain a linear estimate for feasibility violation of the last iterate \(u^k\):

\[
\mathcal{R}_d = \max_{\lambda} \mathcal{R}_d(x) = \max_{\lambda} \min_{x} \|\lambda^*\| \quad \text{and} \quad \mathcal{T}_d = \max_{\lambda} \|\lambda^*\|.
\]

6. Numerical implementation in embedded MPC

In this section we investigate the cost of computing one iteration of Algorithm (DG) for linear MPC problem (QP(x)). First of all, we observe that in this case, the inner problem (step 1 in Algorithm (DG)) reduces to solving an unconstrained QP of dimension \(N_{n\lambda}\), but having a dense Hessian \(Q\). In terms of the corresponding system of linear equations, \(Q \in \mathbb{R}^{N_{n\lambda} \times N_{n\lambda}}\) is therefore an SPD dense matrix (see (4)), hence the inner problem can be solved using an unstructured Cholesky factorization in \(O(N_{n\lambda}^3\lambda^2)\) operations. For updating the Lagrange multipliers (step 2 in Algorithm (DG)) we need to perform a matrix vector multiplication, involving the lower block triangular matrix \(G \in \mathbb{R}^{p \times N_{n\lambda}}\). Thus, step 2 in Algorithm (DG) is much cheaper than step 1. Recall that in the sparse MPC formulation, i.e. the future states are also kept as decision variables and the system dynamics are incorporated into the problem by enforcing equality constraints, the cost of computing one step in Algorithm (DG) is of order \(O(N(\lambda^1 + \lambda^2))\) operations (see e.g. [10]). The cubic growth of computational requirements with respect to the horizon length, in contrast to the linear growth exhibited by the sparse formulation, suggests that the condensed approach could be favored by applications with small number of inputs and that require short horizons.

Regarding the memory requirements, which are an important aspect for embedded implementations, we observe that most memory would be used for storing matrices \(Q\) and \(G\) (see (4)). Thus, in the condensed approach, the cost of storing these matrices is approximately \(O(N_{n\lambda}^2\lambda^2)\) elements.

Finally, we note that for \(\lambda^0 = 0\) our estimates on the convergence rate depend on \(\mathcal{R}_d = \min_{\lambda} \|\lambda^*\|\) and \(\mathcal{T}_d = \max_{\lambda} \|\lambda^*\|\). In MPC, for each initial state \(x \in X\) we have a corresponding optimal set of Lagrange multipliers \(\Lambda(x)\) and consequently an \(\mathcal{R}_d(x)\) and \(\mathcal{T}_d(x)\). In order to get tight estimates we need to obtain a good bound on:

\[
\mathcal{R}_d = \max_{\lambda} \mathcal{R}_d(x) = \max_{\lambda} \min_{x \in X} \|\lambda^*\| \quad \text{and} \quad \mathcal{T}_d = \max_{\lambda} \mathcal{T}_d(x) = \max_{\lambda} \min_{x \in X} \|\lambda^*\|.
\]

Given the expression of MPC problem (QP(x)), we can see immediately that its dual problem has the form:

\[
\min_{\lambda} \frac{1}{2} \lambda^T H \lambda + (Dx + d)^T \Lambda,
\]

for appropriate matrices \(H\) and \(D\). Based on [21, Theorem 19] we can compute \(\mathcal{R}_d\) by solving the non-convex problem:

\[
\mathcal{R}_d = \max_{\lambda} \|\lambda^*\|
\]

s.t. \(0 \leq \lambda^* \perp H \lambda^* + Dx + d \geq 0\)

\[
\lambda^*_i (H_i \lambda^* + d_i) > 0, \quad \lambda^*_i + H_i \lambda^* \geq 0 \quad \forall i \in N(x),
\]

where \(N(x) = \{i : H_i \lambda^* + D_i x + d_i > 0\}\) for some \(\lambda^* \in \Lambda(x)\) and \(H_i, D_i\) denote the \(i\)th row of \(H\) and \(D\). Moreover, using the expression of MPC problem (QP(x)) we obtain the following
non-convex problem for $T_d^*$:

$$T_d^* = \max_{x, \lambda} \|\lambda\|$$

s.t. $
\begin{bmatrix}
Q \lambda' + Wx + G^T \lambda' = 0 \\
Gu + g \leq 0, \quad \lambda' \geq 0, \quad (Gu + g)^T \lambda' = 0.
\end{bmatrix}
$$

Note that the complementarity conditions in the above two problems induces nonlinearity, thus making them difficult to solve. This can be handled by introducing auxiliary binary variables and can be modeled as mixed-integer quadratic programs by employing the so-called "big-M" technique (see e.g. [12]).

6.1. Numerical simulations

We consider a simplified model for the self balancing Lego mindstorm NXT extracted from [26, 3]. The model is linear time invariant and stabilizable. The continuous linear model has $x \in \mathbb{R}^4$ and $u \in \mathbb{R}$. The states for this system are the horizontal position and speed ($h, \dot{h}$), and the angle to the vertical and the angular velocity of the robot’s body ($\theta, \dot{\theta}$). The input for the system represents the pulse-width modulated voltage applied to both wheel motors in percentages. We discretize the system via the zero-order hold method for a sample time of $T = 8$ms to obtain the system matrices:

$$A = \begin{bmatrix}
1 & 0.0054 & -2 \times 10^{-4} & 10^{-4} \\
0 & 0.4717 & -0.0465 & 0.0211 \\
0 & 0.03 & 1.0049 & 0.0068 \\
0 & 6.0742 & 1.0721 & 0.7633
\end{bmatrix}, \quad B = \begin{bmatrix}
0.0002 \\
0.0448 \\
-0.0025 \\
-0.5147
\end{bmatrix}.$$

For this system we consider the duty-cycle percentage constraints for the inputs, i.e. $-12 \leq u(t) \leq 12$ and additional constraints for the position: $-0.5 \leq h \leq 0.5$ and for the body angle in degrees, i.e. $-15 \leq \theta \leq 15$. For the cost matrices we consider: $Q = \text{diag}(1 1 6 \times 10^{-1})$ and $R = 2$. We consider two MPC formulations: smooth MPC given in (QP(x)) and nonsmooth MPC, where we add additionally a penalty term $[u(t) - u(t - 1)]$ in order to get smoother controller. Note that in the first formulation we obtain a QP as in (QP(x)) for which both Assumptions 3.1 and 5.1 hold, while in the second formulation we have an additional penalty term of the form $\beta||Du||_1$, where $\beta = 1$ and $D$ is bidiagonal matrix, for which only Assumption 3.1 holds. Initial state is $x = [0 0 0.5 - 0.35]^T$ and we add gentle disturbances to the system at each 20 simulation steps. In the figure we plot the MPC trajectories of the state angle and input for $N = 10$ obtained using Algorithm (DG) in the last iterate with accuracy $\epsilon = 10^{-2}$. We observe a smoother behavior for MPC with penalty term $[u(t) - u(t - 1)]$.

From our knowledge all the existing results from literature of linear MPC based on dual first order methods state sublinear convergence. On the other hand, the condensed formulation of the smooth MPC (problem (QP(x))) leads to a convex quadratic problem for which both Assumptions 3.1 and 5.1 hold. Thus, according to our theory, the Algorithm (DG) converges on this smooth MPC formulation linearly. Moreover, from the first part of the table below we observe that Algorithm (DG) for smooth MPC problems, with $N$ ranging from 10 to 100 and initial state as above, has a better behavior in the last iterate ($K^{DG}$ iterations for obtaining accuracy $\epsilon = 10^{-2}$ for primal suboptimality and infeasibility) than in the average sequence ($K^{av}$) and on certain problems even performs better than dual fast gradient algorithm in an average sequence ($K^{av}$) (proposed in [9]). In the second part of the table we consider nonsmooth MPC under the same conditions as above, but for a small penalty $\beta = 0.1$. All algorithms (i.e. (DG) and fast gradient in [9]) performs worse in the nonsmooth case than in the smooth MPC, and still Algorithm (DG) has a better behavior in the last iterate sequence than fast gradient method for horizon lengths $N < 80$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
<th>60</th>
<th>70</th>
<th>80</th>
<th>90</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K^{DG}$</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>13</td>
<td>25</td>
<td>65</td>
<td>197</td>
<td>628</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K^{av}$</td>
<td>53</td>
<td>58</td>
<td>66</td>
<td>94</td>
<td>182</td>
<td>470</td>
<td>1.4×10^4</td>
<td>4.4×10^4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K^{av}$</td>
<td>15</td>
<td>16</td>
<td>19</td>
<td>27</td>
<td>45</td>
<td>80</td>
<td>146</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K^{DG}$</td>
<td>10</td>
<td>12</td>
<td>14</td>
<td>17</td>
<td>30</td>
<td>81</td>
<td>219</td>
<td>681</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K^{av}$</td>
<td>66</td>
<td>72</td>
<td>81</td>
<td>132</td>
<td>226</td>
<td>584</td>
<td>1.7×10^4</td>
<td>5.4×10^4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$K^{av}$</td>
<td>21</td>
<td>23</td>
<td>27</td>
<td>34</td>
<td>37</td>
<td>62</td>
<td>101</td>
<td>178</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

References


