COMPUTATIONAL COMPLEXITY OF INEXACT GRADIENT AUGMENTED LAGRANGIAN METHODS: APPLICATION TO CONSTRAINED MPC

VALENTIN NEDELCU*, ION NECOARA* AND QUOC TRAN DINH†

Abstract. We study the computational complexity certification of inexact gradient augmented Lagrangian methods for solving convex optimization problems with complicated constraints. We solve the augmented Lagrangian dual problem that arises from the relaxation of complicating constraints with gradient and fast gradient methods based on inexact first order information. Moreover, since the exact solution of the augmented Lagrangian primal problem is hard to compute in practice, we solve this problem up to some given inner accuracy. We derive relations between the inner and the outer accuracy of the primal and dual problems and we give a full convergence rate analysis for both gradient and fast gradient algorithms. We provide estimates on the primal and dual suboptimality and on primal feasibility violation of the generated approximate primal and dual solutions. Our analysis relies on the Lipschitz property of the dual function and on inexact dual gradients. We also discuss implementation aspects of the proposed algorithms on constrained model predictive control problems for embedded linear systems.

Key words. Gradient and fast gradient methods, iteration-complexity certification, augmented Lagrangian, convex programming, embedded systems, constrained linear model predictive control.

1. Introduction.

Embedded control systems has been widely used in many applications and its usage in industrial plants has increased concurrently. The concept behind embedded control is to design a control scheme that can be implemented on autonomous electronic hardware, e.g a programmable logic controller [21], a microcontroller circuit board [22, 17] or field-programmable gate arrays [8]. One of the most successful advanced control schemes implemented in industry is model predictive control (MPC) and this is 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. In the recent decades there has been a growing focus on developing faster MPC schemes, improving the computational efficiency [15] and providing worst case computational complexity certificates for the applied solution methods [9, 16, 17], making these schemes feasible for implementation on hardware with limited computational power.

For fast embedded systems [7, 8, 17] 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 [2]. Further, these methods have complex iterations, involving inversion of matrices, which are usually difficult to implement on embedded systems, where the units demand simple computations. Therefore, first order methods are more suitable in these situations [17].

When the projection on the primal feasible set is hard to compute, e.g. for constrained MPC problems, an alternative to primal gradient methods is to use the

*V. Nedelcu and I. Necoara are with Automation and Systems Engineering Department, University Politehnica of Bucharest, 060042 Bucharest, Romania. Email:{valentin.nedelcu, ion.necoara}@acse.pub.ro

†Q. Tran Dinh is with Department of Electrical Engineering (ESAT-SCD) and Optimization in Engineering Center (OPTEC), KU Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium. Email: quoc.trandinh@esat.kuleuven.be. (Permanent address: Faculty of Mathematics, Mechanics and Informatics, VNU University of Science, Hanoi, Vietnam).

Lagrangian relaxation to handle the complicated constraints and then to apply dual gradient schemes. The computational complexity certification of gradient-based methods for solving the (augmented) Lagrangian dual of a primal convex problem is studied e.g. in [4, 9, 10, 11, 14, 16, 19]. In [4] the authors present a general framework for gradient methods with inexact oracle, i.e. only approximate information is available for the values of the function and of its gradient, and give convergence rate analysis. The authors also apply their approach to gradient augmented Lagrangian methods and provided estimates only for dual suboptimality. In [19] an augmented Lagrangian algorithm is analyzed using the theory of monotone operators. For this algorithm the author proves asymptotic convergence under general conditions and local linear convergence under second order optimality conditions. In [14] a dual fast gradient method is proposed for solving quadratic programs with linear inequality constraints and estimates on primal suboptimality and infeasibility of the primal solution are provided. In [9] the authors analyze the iteration complexity of an inexact augmented Lagrangian method where the approximate solutions of the inner problems are obtained by using a fast gradient scheme, while the dual variables are updated by using an inexact dual gradient method. The authors also provides upper bounds on the total number of iterations which have to be performed by the algorithm for obtaining a primal suboptimal solution. In [10] a dual method based on fast gradient schemes and smoothing techniques of the ordinary Lagrangian is presented. Using an averaging scheme the authors are able to recover a primal suboptimal solution and provide estimates on both dual and primal suboptimality and also on primal infeasibility.

Despite widespread use of the dual gradient methods for solving Lagrangian dual problems, there are some aspects of these methods that have not been fully studied. In particular, the previous work has several limitations. First, the focus is mainly on the convergence analysis of the dual variables. Second, only the dual gradient method is usually analyzed and using exact information. Third, there is no full convergence rate analysis (i.e. no estimates in terms of dual and primal suboptimality and primal feasibility violation) for both dual gradient and fast gradient schemes, while using inexact dual information. Therefore, in this paper we focus on solving convex optimization problems (possibly nonsmooth) approximately by using an augmented Lagrangian approach and inexact dual gradient and fast gradient methods. We show how approximate primal solutions can be generated based on averaging for general convex problems and we give a full convergence rate analysis for both algorithms that leads to error estimates on the amount of constraint violation and the cost of primal and dual solutions. Since we allow one to solve the inner problems approximately, our dual gradient schemes have to use inexact information.

Contribution. The contributions of this paper include the following:

1. We propose and analyze dual gradient algorithms producing approximate primal feasible and optimal solutions. Our analysis is based on the augmented Lagrangian framework which leads to the dual function having Lipschitz continuous gradient, even if the primal objective function is not strongly convex.
2. Since exact solutions of the inner problems are usually hard to compute, we solve these problems only up to a certain inner accuracy ε_{in}. We analyze several stopping criteria which can be used in order to find such a solution and point out their advantages.
3. For solving outer problem we propose two inexact dual gradient algorithms: - an inexact dual gradient algorithm, with complexity $\mathcal{O}(1/\varepsilon_{\text{out}})$ iterations, which allows us to find an ε_{out}-optimal solution of the original problem by

solving the inner problems with an accuracy ε_{in} of order $\mathcal{O}(\varepsilon_{\text{out}})$.
- an inexact dual fast gradient algorithm, with complexity $\mathcal{O}(\sqrt{1/\varepsilon_{\text{out}}})$ iterations, provided that the inner problems are solved with accuracy ε_{in} of order $\mathcal{O}(\varepsilon_{\text{out}}\sqrt{\varepsilon_{\text{out}}})$.

4. For both methods we show how to generate approximate primal solutions and provide estimates on dual and primal suboptimality and primal infeasibility.
5. To certify the complexity of the proposed methods, we apply the algorithms on linear embedded MPC problems with state and input constraints.

Paper outline. The paper is organized as follows. In Section 1, motivated by embedded MPC, we introduce the augmented Lagrangian framework for solving constrained convex problems. In Section 2 we discuss different stopping criteria for finding a suboptimal solution of the inner problems and provide estimates on the complexity of finding such a solution. In Section 3 we propose an inexact dual gradient and fast gradient algorithm for solving the outer problem. For both algorithms we provide bounds on the dual and primal suboptimality and also on the primal infeasibility. In Section 4 we specialize our general results to constrained linear MPC problems and we obtain tight bounds on the number of inner and outer iterations. We also provide extensive numerical tests to prove the efficiency of the proposed algorithms.

Notation and terminology. We work in the space $\mathbb{R}^n$ composed by column vectors. For $x, y \in \mathbb{R}^n$, $\langle x, y\rangle := x^T y = \sum_{i=1}^n x_i y_i$ and $\|x\| := (\sum_{i=1}^n x_i^2)^{1/2}$ denote the standard Euclidean inner product and norm, respectively. We use the same notation $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ for spaces of different dimension. We denote by $\text{cone}\{a_i,\ i \in I\}$ the cone generated from vectors $\{a_i,\ i \in I\}$. We also denote by $R_{\text{p}} := \max_{z,y\in Z} \|z - y\|$ the diameter, $\text{int}(Z)$ the interior and $\text{bd}(Z)$ the boundary of a convex, compact set Z. By $\text{dist}(y, Z)$ we denote the Euclidean distance from a point y to the set Z and by $h_Z(y) := \sup_{z\in Z} y^T z$ the support function of the set Z. For any point $\tilde{z} \in Z$ we denote by $\mathcal{N}_Z(\tilde{z}) := \{s \mid \langle s, z - \tilde{z}\rangle \le 0\ \forall z \in Z\}$ the normal cone of Z at $\tilde{z}$. For a real number x, $\lfloor x \rfloor$ denotes the largest integer number which is less than or equal to x, while ":=" means "equal by definition".

1.1. A motivating example: Linear MPC problems with state-input constraints. We consider a discrete time linear system given by the dynamics:

$$x_{k+1} = A_x x_k + B_u u_k,$$

where $x_k \in \mathbb{R}^{n_x}$ represents the state and $u_k \in \mathbb{R}^{n_u}$ represents the input of the system. We also assume hard state and input constraints:

$$x_k \in X \subseteq \mathbb{R}^{n_x}, \quad u_k \in U \subseteq \mathbb{R}^{n_u}\ \ \forall k \ge 0.$$

Now, we can define the linear MPC problem over the prediction horizon of length N, for a given initial state x, as follows [20]:

$$f^*(x) := \left\{ \begin{array}{ll} \min\limits_{x_i,u_i} & \sum_{i=0}^{N-1} \ell(x_i, u_i) + \ell_{\text{f}}(x_N) \\ \text{s.t.} & x_{i+1} = A_x x_i + B_u u_i,\ x_0 = x, \\ & x_i \in X,\ u_i \in U\ \forall i,\ x_N \in X_{\text{f}}, \end{array} \right. \tag{1.1}$$

where both the stage cost ℓ and the terminal cost ℓ_{f} are convex functions (possibly nonsmooth). Note that in our formulation we do not require strongly convex costs. Further, the terminal set X_{f} is chosen so that stability of the closed-loop system is guaranteed. We assume the sets X, U and X_{f} to be compact, convex and simple (by simple we understand that the projection on these sets can be done easily, e.g. boxes).

Furthermore, we introduce the notation $z := \left[x_1^T \cdots x_N^T \; u_0^T \cdots u_{N-1}^T\right]^T$, $Z := \prod_{i=1}^{N-1} X \times X_{\mathrm{f}} \times \prod_{i=1}^{N} U$ and $f(z) := \sum_{i=0}^{N-1} \ell(x_i, u_i) + \ell_{\mathrm{f}}(x_N)$. We can also write compactly the linear dynamics $x_{i+1} = A_x x_i + B_u u_i$ for all $i = 0, \cdots, N-1$ and $x_0 = x$ as $Az = b(x)$ (see [20, 23] for details). Note that $b(x) \in \mathbb{R}^{Nn_x}$ depends linearly on x, i.e. $b(x) := \left[(A_x x)^T \; 0^T \cdots 0^T\right]^T$. In these settings, for linear MPC we need to solve, for a given initial state x, the primal convex optimization problem:

$$(\mathbf{P}(x)) \qquad \min_z \{f(z) \mid Az = b(x),\ z \in Z\},$$

where f is a convex function (possibly nonsmooth) and A is a matrix of appropriate dimension. Moreover, the set Z is simple as long as X, X_{f} and U are simple sets. In the following sections we discuss how we can efficiently solve optimization problem $(\mathbf{P}(x))$ approximately with dual gradient methods based on inexact first order information and we provide tight estimates for the total number of iterations which has to be performed in order to obtain a suboptimal solution in terms of primal suboptimality and infeasibility.

1.2. Augmented Lagrangian framework. Motivated by MPC problems, we are interested in solving convex optimization problems of the form:

$$(\mathbf{P}) \qquad f^* := \left\{ \begin{array}{ll} \min\limits_{z \in \mathbb{R}^n} & f(z) \\ \text{s.t.} & Az = b,\ z \in Z, \end{array} \right.$$

where f is convex function (possibly nonsmooth), $A \in \mathbb{R}^{m \times n}$ is a full row-rank matrix and Z is a simple set (i.e. the projection on this set is computationally cheap), compact and convex. We will denote problem $(\mathbf{P})$ as the primal problem and f as the primal objective function.

A common approach for solving problem $(\mathbf{P})$ consists of applying interior point methods, which usually perform much lower number of iterations in practice than those predicted by the theoretical worst case complexity analysis [2]. On the other hand, for first order methods the number of iterations predicted by the worst case complexity analysis is close to the actual number of iterations performed by the method [12]. This is crucial in the context of fast embedded systems. First order methods applied directly to problem $(\mathbf{P})$ imply projection on the feasible set $\{z \mid z \in Z,\ Az = b\}$. Note that even if Z is a simple set, the projection on the feasible set is hard due to the complicating constraints $Az = b$. An efficient alternative is to move the complicating constraints into the cost via Lagrange multipliers and solve the dual problem approximately by using first order methods and then recover a primal suboptimal solution for $(\mathbf{P})$. This is the approach that we follow in this paper: we derive inexact dual gradient methods that allow us to generate approximate primal solutions for which we provide estimates for the violation of the constraints and upper and lower bounds on the corresponding primal objective function value of $(\mathbf{P})$.

First let us define the dual function:

$$d(\lambda) := \min_{z \in Z} \mathcal{L}(z, \lambda), \tag{1.2}$$

where $\mathcal{L}(z, \lambda) := f(z) + \langle \lambda, Az - b \rangle$ represents the partial Lagrangian with respect to the constraints $Az = b$ and λ the associated Lagrange multipliers. Now, we can write the corresponding dual problem as follows:

$$(\mathbf{D}) \qquad \max_{\lambda \in \mathbb{R}^m} d(\lambda).$$

We assume that Slater's constraint qualification holds, so that problems (**P**) and (**D**) have the same optimal value. We also denote by z^* an optimal solution of (**P**) and by λ^* the corresponding multiplier (i.e. an optimal solution of (**D**)).

In general, the dual function d is not differentiable [1] and therefore any subgradient method for solving (**D**) has a slow convergence rate [11]. We will see in the sequel how we can avoid this drawback by means of augmented Lagrangian framework. We define the augmented Lagrangian function [6]:

$$\mathcal{L}_\rho(z,\lambda) := f(z) + \langle \lambda, Az - b\rangle + \frac{\rho}{2}\|Az-b\|^2, \tag{1.3}$$

where $\rho > 0$ represents a penalty parameter. The augmented dual problem, called also the *outer* problem, is defined as:

$$\max_{\lambda\in\mathbb{R}^m} d_\rho(\lambda), \tag{$\mathbf{D}_\rho$}$$

where $d_\rho(\lambda) := \min\limits_{z\in Z} \mathcal{L}_\rho(z,\lambda)$ and we denote by $z^*(\lambda)$ an optimal solution of the *inner* problem $\min_{z\in Z}\mathcal{L}_\rho(z,\lambda)$ for a given λ. It is well-known [1, 9] that the optimal value and the set of optimal solutions of the dual problems (**D**) and ($\mathbf{D}_\rho$) coincide. Furthermore, the function d_ρ is concave and differentiable and its gradient is [13]:

$$\nabla d_\rho(\lambda) := Az^*(\lambda) - b.$$

Moreover, the gradient mapping $\nabla d_\rho(\cdot)$ is Lipschitz continuous with a Lipschitz constant [1] given by:

$$L_{\mathrm{d}} := \rho^{-1}.$$

In conclusion, we want to solve within an accuracy $\varepsilon_{\mathrm{out}}$ the equivalent smooth outer problem ($\mathbf{D}_\rho$) by using first order methods with inexact gradients (e.g. dual gradient or fast gradient algorithms) and then recover an approximate primal solution. In other words, the goal of this paper is to generate a primal-dual pair $(\hat{z},\hat{\lambda})$, with $\hat{z}\in Z$, for which we can ensure bounds on dual suboptimality, primal infeasibility and primal suboptimality of order $\varepsilon_{\mathrm{out}}$, i.e.:

$$f^* - d_\rho(\hat{\lambda}) \leq \mathcal{O}\left(\varepsilon_{\mathrm{out}}\right),\ \|A\hat{z}-b\| \leq \mathcal{O}\left(\varepsilon_{\mathrm{out}}\right) \text{ and } |f(\hat{z}) - f^*| \leq \mathcal{O}\left(\varepsilon_{\mathrm{out}}\right). \tag{1.4}$$

2. Complexity estimates of solving the inner problems. As we have seen in the previous section, in order to compute the gradient ∇d_ρ we have to find, for a given λ, an optimal solution of the *inner* convex problem:

$$z^*(\lambda) \in \arg\min_{z\in Z} \mathcal{L}_\rho(z,\lambda). \tag{2.1}$$

From the optimality conditions [18], we know that a point $z^*(\lambda)$ is an optimal solution of (2.1) if and only if:

$$\langle \nabla\mathcal{L}_\rho\left(z^*(\lambda),\lambda\right), z - z^*(\lambda)\rangle \geq 0 \quad \forall z\in Z. \tag{2.2}$$

An equivalent way to characterize an optimal solution $z^*(\lambda)$ of (2.1) can be given in terms of the following inclusion:

$$0 \in \nabla\mathcal{L}_\rho\left(z^*(\lambda),\lambda\right) + \mathcal{N}_Z(z^*(\lambda)). \tag{2.3}$$

Since an exact minimizer of the inner problem (2.1) is usually hard to compute, we are interested in finding an approximate solution of this problem instead of its optimal one. Therefore, we have to consider an inner accuracy ε_{in} which measures the suboptimality of such an approximate solution for (2.1):

$$\bar{z}(\lambda) \approx \arg\min_{z \in Z} \left\{ f(z) + \langle \lambda, Az - b \rangle + \frac{\rho}{2} \|Az - b\|^2 \right\}.$$

Since there exist several ways to characterize an ε_{in}-optimal solution [4, 9, 19], we will further discuss different stopping criteria which can be used in order to find such a solution. A well-known stopping criterion, which measures the distance to optimal value of (2.1), is given by:

$$\bar{z}(\lambda) \in Z, \quad \mathcal{L}_\rho\left(\bar{z}(\lambda), \lambda\right) - \mathcal{L}_\rho\left(z^*(\lambda), \lambda\right) \leq \varepsilon_{\text{in}}^2. \tag{2.4}$$

The main advantage of using (2.4) as a stopping criterion for finding $\bar{z}(\lambda)$ consists of the fact that in the literature [12] there exist explicit bounds on the number of iterations which has to be performed by some well-known first or second order methods to ensure the ε_{in}-optimality.

Another stopping criterion, which measures the distance of $\bar{z}(\lambda)$ to the set of optimal solution $Z^*(\lambda)$ of (2.1) is given by:

$$\bar{z}(\lambda) \in Z, \ \text{dist}\left(\bar{z}(\lambda), Z^*(\lambda)\right) \leq \mathcal{O}\left(\varepsilon_{\text{in}}\right). \tag{2.5}$$

It is known that this distance can be bounded by an easily computable quantity when the objective function satisfies the so-called gradient error bound [1]. Thus, we can use this bound to define stopping rules in iterative algorithms for solving the optimization problem. Note that gradient error bound assumption is a generalization of the more restrictive notion of strong convexity.

As a direct consequence of the optimality condition (2.2), one can use the following stopping criterion:

$$\bar{z}(\lambda) \in Z, \quad \langle \nabla \mathcal{L}_\rho(\bar{z}(\lambda), \lambda), z - \bar{z}(\lambda) \rangle \geq -\mathcal{O}(\varepsilon_{\text{in}}) \quad \forall z \in Z. \tag{2.6}$$

Note that (2.6) can be formulated using the support function as:

$$h_Z\left(-\nabla \mathcal{L}_\rho\left(\bar{z}(\lambda), \lambda\right)\right) + \langle \nabla \mathcal{L}_\rho(\bar{z}\left(\lambda\right), \lambda), \bar{z}(\lambda) \rangle \leq \mathcal{O}\left(\varepsilon_{\text{in}}\right).$$

When the set Z has a specific structure (e.g. a ball defined by some norm), tight upper bounds on the support function can be computed explicitly and thus the stopping criterion can be efficiently verified.

Based on optimality conditions (2.3), the following stopping criterion can also be used in order to characterize an ε_{in}-optimal solution $\bar{z}(\lambda)$ of the inner problem (2.1):

$$\bar{z}(\lambda) \in Z, \ \text{dist}\big(0, \nabla \mathcal{L}_\rho(\bar{z}(\lambda), \lambda) + \mathcal{N}_Z(\bar{z}(\lambda))\big) \leq \mathcal{O}(\varepsilon_{\text{in}}). \tag{2.7}$$

The main advantage of using this criterion is given by the fact that the distance in (2.7) can be computed efficiently for sets Z having a certain structure. Note that (2.7) can be verified by solving the following projection problem:

$$s^* \in \arg\min_{s \in \mathcal{N}_Z(\bar{z}(\lambda))} \|\nabla \mathcal{L}_\rho\left(\bar{z}(\lambda), \lambda\right) + s\|^2. \tag{2.8}$$

To see how (2.8) can be solved efficiently we are first interested in finding an explicit characterization of the normal cone $\mathcal{N}_Z(\bar{z}(\lambda))$, when the set Z has a certain structure.

LEMMA 2.1. *Assume that the set Z is a general polyhedral set, i.e. $Z := \{z \in \mathbb{R}^n \mid Cz \leq c\}$, with $C \in \mathbb{R}^{p\times n}$ and $c \in \mathbb{R}^p$. Then, problem* (2.8) *can be recast as the following quadratic optimization problem:*

$$\min_{\mu\geq 0} \|\nabla\mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) + \tilde{C}^T\mu\|^2, \tag{2.9}$$

where matrix $\tilde{C}$ contains the rows of C corresponding to the active constraints in $C\bar{z}(\lambda) \leq c$. In particular, if Z is a box in $\mathbb{R}^n$, then problem (2.9) *becomes separable and it can be solved explicitly in $\mathcal{O}(\tilde{p})$ operations, where $\tilde{p}$ represents the number of active constraints in $C\bar{z}(\lambda) \leq c$.*

Proof. Let us recall that if $\bar{z}(\lambda) \in \mathrm{int}(Z)$, then we have $\mathcal{N}_Z\left(\bar{z}(\lambda)\right) = \{0\}$ and therefore the distance $\mathrm{dist}\left(0, \nabla\mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) + \mathcal{N}_Z\left(\bar{z}(\lambda)\right)\right)$ will be equal to $\|\nabla\mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right)\|$. In the case of $\bar{z}(\lambda) \in \mathrm{bd}(Z)$ there exists an index set $\mathrm{I}\left(\bar{z}(\lambda)\right) \subseteq \{1,\cdots,p\}$ such that $C_i\bar{z}(\lambda) = c_i$ for all $i \in \mathrm{I}(\bar{z}(\lambda))$, where C_i and c_i represent the i-th row and i-th element of C and c, respectively. Using now Theorem 6.46 in [18] we have $\mathcal{N}_Z\left(\bar{z}(\lambda)\right) = \mathrm{cone}\left\{C_i^T, i \in \mathrm{I}\left(\bar{z}(\lambda)\right)\right\}$. Introducing the notation $\tilde{C}$ for the matrix whose rows are C_i for all $i \in \mathrm{I}\left(\bar{z}(\lambda)\right)$, we can write (2.8) as (2.9). Note that, in problem (2.9), the dimension of the variable μ is $\tilde{p} = |\mathrm{I}\left(\bar{z}(\lambda)\right)|$ (i.e. the number of active constraints) which usually is smaller in comparison with the dimension n of problem (2.8).

Now, if we assume that Z is a box in $\mathbb{R}^n$, then problem (2.9) can be written in the following equivalent form:

$$\min_{\mu\geq 0} \frac{1}{2}\mu^T\tilde{C}\tilde{C}^T\mu + \nabla\mathcal{L}_\rho^T\left(\bar{z}(\lambda),\lambda\right)\tilde{C}^T\mu.$$

Since for box constraints we have $\tilde{C}\tilde{C}^T = \mathrm{I}_{\tilde{p}}$, i.e. the identity matrix, the previous optimization problem can be decomposed into $\tilde{p}$ scalar projection problems onto the nonnegative orthant and thus in order to compute the optimal solution of (2.9) we only have to perform $\mathcal{O}(\tilde{p})$ comparisons. ∎

The next lemma establishes some relations between stopping criteria (2.4)-(2.7).

LEMMA 2.2. *The conditions* (2.4), (2.5), (2.6) *and* (2.7) *satisfy the following:*

(i) *Let $\nabla\mathcal{L}_\rho$ be Lipschitz continuous with a Lipschitz constant L_p. Then*

$$(2.4) \;\Rightarrow\; (2.6),\; (2.5) \;\Rightarrow\; (2.6),\; (2.7) \;\Rightarrow\; (2.6).$$

(ii) *If, in addition, $\mathcal{L}_\rho$ is strongly convex with a convexity parameter $\sigma_p > 0$, then*

$$(2.7) \;\Rightarrow\; (2.4) \;\Rightarrow\; (2.5).$$

Proof. (i) (2.4) ⇒ (2.6): In Section 3 of [4] the authors show that if (2.4) holds, then (2.6) also holds with $\mathcal{O}(\varepsilon_{\mathrm{in}}) = \varepsilon_{\mathrm{in}}^2 + \sqrt{L_\mathrm{p}}R_\mathrm{p}\varepsilon_{\mathrm{in}}$.

(2.5) ⇒ (2.6): We can write:

$$\begin{aligned}
&\langle\nabla\mathcal{L}_\rho(\,\bar{z}(\lambda),\lambda\,), z - \bar{z}(\lambda)\rangle\\
&= \langle\nabla\mathcal{L}_\rho(\,\bar{z}(\lambda),\lambda\,) - \nabla\mathcal{L}_\rho(z^*(\lambda),\lambda\,), z - \bar{z}(\lambda)\rangle + \langle\nabla\mathcal{L}_\rho(z^*(\lambda),\lambda\,), z - z^*(\lambda) + z^*(\lambda) - \bar{z}(\lambda)\rangle\\
&\geq -\left(L_\mathrm{p}R_\mathrm{p} + \|\nabla\mathcal{L}_\rho\left(z^*(\lambda),\lambda\right)\|\right)\varepsilon_{\mathrm{in}}.
\end{aligned}$$

Since Z is compact and $\nabla\mathcal{L}_\rho(\cdot,\lambda)$ is continuous, then $\nabla\mathcal{L}_\rho(\cdot,\lambda)$ is bounded. Hence, our statement follows from the last inequality.

(2.7) $\Rightarrow$ (2.6): The condition (2.7) can be written as $\langle \nabla \mathcal{L}_\rho(\,\bar{z}(\lambda),\lambda\,), z-\bar{z}(\lambda)\rangle \geq -\varepsilon_{\rm in}\,\langle e, z-\bar{z}(\lambda)\rangle$ for all e such that $\|e\|\leq 1$ and $z\in Z$. If Z is bounded, then the last inequality implies $\langle \nabla \mathcal{L}_\rho(\,\bar{z}(\lambda),\lambda\,), z-\bar{z}(\lambda)\rangle \geq -R_{\rm p}\varepsilon_{\rm in}$.
(ii) (2.7) $\Rightarrow$ (2.4): Since $\mathcal{L}_\rho(z,\lambda)$ is strongly convex and $z^*(\lambda)$ is its minimizer over Z, we have:

$$0 \geq \mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) - \mathcal{L}_\rho\left(z^*(\lambda),\lambda\right) \geq \frac{\sigma_{\rm p}}{2}\|\bar{z}(\lambda)-z^*(\lambda)\|^2.$$

From the convexity of $\mathcal{L}_\rho(z,\lambda)$ we can write:

$$\begin{aligned}
&\mathcal{L}_\rho(\bar{z}(\lambda),\lambda) - \mathcal{L}_\rho\left(z^*(\lambda),\lambda\right) \leq \langle \nabla \mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right), \bar{z}(\lambda)-z^*(\lambda)\rangle\\
&\leq \langle \nabla \mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) + s^*, \bar{z}(\lambda)-z^*(\lambda)\rangle \leq \|\nabla \mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) + s^*\|\|\bar{z}(\lambda)-z^*(\lambda)\|\\
&\leq \varepsilon_{\rm in}\left[\frac{2}{\sigma_{\rm p}}\left(\mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) - \mathcal{L}_\rho\left(z^*(\lambda),\lambda\right)\right)\right]^{1/2},
\end{aligned}$$

which implies (2.4).
(2.4) $\Rightarrow$ (2.5): Taking into account that $\mathcal{L}_\rho(\cdot,\lambda)$ is strongly convex we have $\frac{\sigma_{\rm p}}{2}\|z^*(\lambda)-\bar{z}(\lambda)\| \leq \mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) - \mathcal{L}_\rho\left(z^*(\lambda),\lambda\right) \leq \varepsilon_{\rm in}^2$. This leads to $\|\bar{z}(\lambda)-z^*(\lambda)\| \leq (2/\sigma_{\rm p})^{1/2}\varepsilon_{\rm in}$. The lemma is proved. □

The next theorem provides estimates on the number of iterations that are required by fast gradient schemes to obtain an $\varepsilon_{\rm in}$ approximate solution for inner problem (2.1).

THEOREM 2.3. *[12] Assume that function $\mathcal{L}_\rho(\cdot,\lambda)$ has Lipschitz continuous gradient w.r.t. variable z, with a Lipschitz constant L_p and a fast gradient scheme [12] is applied for finding an $\varepsilon_{\rm in}$ approximate solution $\bar{z}(\lambda)$ of* (2.1) *such that stopping criterion* (2.4) *holds, i.e. $\mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) - \mathcal{L}_\rho\left(z^*(\lambda),\lambda\right) \leq \varepsilon_{\rm in}^2$. Then, the complexity of finding $\bar{z}(\lambda)$ is $\mathcal{O}\left(\sqrt{\frac{L_p}{\varepsilon_{\rm in}^2}}\right)$ iterations. If, in addition $\mathcal{L}_\rho(\cdot,\cdot)$ is strongly convex with a convexity parameter $\sigma_p>0$, then $\bar{z}(\lambda)$ can be computed in at most $\mathcal{O}\left(\sqrt{\frac{L_p}{\sigma_p}}\ln\left(\frac{\sigma_p}{\varepsilon_{\rm in}^2}\right)\right)$ iterations by using a fast gradient scheme.*

Note that if the function f is nonsmooth, we have a complexity $\mathcal{O}\left(\frac{1}{(\varepsilon_{\rm in}^2)^2}\right)$ iterations with a projected subgradient method or an improved $\mathcal{O}\left(\frac{1}{\varepsilon_{\rm in}^2}\right)$ by using smoothing techniques [13], provided that f has a certain structure.

3. Complexity estimates of solving the outer problem using approximate dual gradients. In this section we solve the augmented Lagrangian dual problem ($\mathbf{D}_\rho$) approximately by using dual gradient and fast gradient methods with inexact information and derive computational complexity certificates for these methods. Since we solve the inner problem inexactly, we have to use inexact gradients and approximate values of the augmented dual function d_ρ defined in terms of $\bar{z}(\lambda)$, i.e. we introduce the following pair:

$$\bar{d}_\rho(\lambda) := \mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right) \quad \text{and} \quad \nabla \bar{d}_\rho(\lambda) := A\bar{z}(\lambda) - b.$$

The next theorem, which is similar to the results in [4], provides bounds on the dual function when the inner problem (2.1) is solved approximately. For completeness we give the proof.

THEOREM 3.1. *If $\bar{z}(\lambda)$ is computed such that the stopping criterion* (2.6) *is satisfied, i.e. $\bar{z}(\lambda)\in Z$ and $\min_{z\in Z}\langle \nabla \mathcal{L}_\rho(\bar{z}(\lambda),\lambda), z-\bar{z}(\lambda)\rangle \geq -\left(1+\sqrt{L_{\rm p}}R_{\rm p}\right)\varepsilon_{\rm in}$,*

then the following inequalities hold:

$$\begin{aligned}(3.1)\quad &\bar{d}_\rho(\lambda)+\left\langle\nabla\bar{d}_\rho(\lambda),\mu-\lambda\right\rangle-\frac{L_{\rm d}}{2}\|\mu-\lambda\|^2-\left(1+\sqrt{L_{\rm p}}R_p\right)\varepsilon_{\rm in}\le d_\rho(\mu)\\ &\le \bar{d}_\rho(\lambda)+\left\langle\nabla\bar{d}_\rho(\lambda),\mu-\lambda\right\rangle\quad\forall\lambda,\mu\in\mathbb{R}^m.\end{aligned}$$

Proof. For simplicity we introduce the notation $C_Z := 1+\sqrt{L_{\rm p}}R_{\rm p}$. The right-hand side inequality follows directly from the definitions of d_ρ and $\bar{d}_\rho$. We only have to prove the left-hand side inequality. Following the derivations from Section 3.3 in [4] we have:

$$\begin{aligned}d_\rho(\mu)&\ge\min_{z\in Z}\left\{f(\bar{z}(\lambda))+\langle\nabla f(\bar{z}(\lambda)),z-\bar{z}(\lambda)\rangle+\langle\mu,Az-b\rangle+\frac{\rho}{2}\|Az-b\|^2\right\}\\ &\ge\min_{z\in z}\left\{f(\bar{z}(\lambda))-\left\langle A^T\lambda+\rho A^T(A\bar{z}(\lambda)-b),z-\bar{z}(\lambda)\right\rangle+\langle\mu,Az-b\rangle+\frac{\rho}{2}\|Az-b\|^2\right\}\\ &\quad+\min_{z\in Z}\left\langle\nabla\mathcal{L}_\rho\left(\bar{z}(\lambda),\lambda\right),z-\bar{z}(\lambda)\right\rangle,\end{aligned}$$

where we use the convexity of f and the properties of minimum in the first and the second inequality, respectively. Using now the assumptions of the theorem and the definition of $\bar{d}_\rho$ we obtain:

$$\begin{aligned}&d_\rho(\mu)\\ &\ge\min_{z\in Z}\left\{f(\bar{z}(\lambda))-\left\langle A^T\lambda+\rho A^T(A\bar{z}(\lambda)-b),z-\bar{z}(\lambda)\right\rangle+\langle\mu,Az-b\rangle+\frac{\rho}{2}\|Az-b\|^2\right\}-C_Z\varepsilon_{\rm in}\\ &=\min_{z\in Z}\left\{\langle A(z-\bar{z}(\lambda)),\mu-\lambda\rangle+\frac{\rho}{2}\|A(z-\bar{z}(\lambda))\|^2\right\}+\bar{d}_\rho(\lambda)+\left\langle\nabla\bar{d}_\rho(\lambda),\mu-\lambda\right\rangle-C_Z\varepsilon_{\rm in}\\ &\ge\min_{z\in\mathbb{R}^n}\left\{\langle A(z-\bar{z}(\lambda)),\mu-\lambda\rangle+\frac{\rho}{2}\|A(z-\bar{z}(\lambda))\|^2\right\}+\bar{d}_\rho(\lambda)+\left\langle\nabla\bar{d}_\rho(\lambda),\mu-\lambda\right\rangle-C_Z\varepsilon_{\rm in}.\end{aligned}$$

By taking into account that $\min_{\xi\in\mathbb{R}^n}\frac{1}{2}\|\xi\|^2+g^T\xi=-\frac{1}{2}\|g\|^2$ and using the definition of C_Z, we obtain from the last expression the right-hand side inequality. ∎

Note that the first inequality helps us construct a quadratic model which bounds from below the function d_ρ, when the exact values of the dual function and its gradients are unknown. The second inequality can be viewed as an approximation of the concavity condition on d_ρ. The two models, the linear and the quadratic one, use only approximate function values and approximate gradients evaluated at certain points. A more general framework for inexact gradient methods can be found in [4].

It can be easily proved that under the assumptions of Theorem 3.1, the following relation holds between the true and approximate gradient:

$$\|\nabla\bar{d}_\rho(\lambda)-\nabla d_\rho(\lambda)\|\le\sqrt{2L_{\rm d}\left(1+\sqrt{L_{\rm p}}R_{\rm p}\right)\varepsilon_{\rm in}}\quad\forall\lambda\in\mathbb{R}^m.$$

3.1. Inexact dual gradient method. In this section we provide the convergence rate analysis of an inexact dual gradient ascent method. Let $\{\alpha_j\}_{j\ge0}$ be a sequence of positive numbers. We denote by $S_k := \sum_{j=0}^k\alpha_j$. In this section we consider an inexact gradient method for updating the dual variables:

(**IDGM**) $$\boxed{\lambda_{k+1}:=\lambda_k+\alpha_k\nabla\bar{d}_\rho(\lambda_k),}$$

where $\alpha_k \in [\underline{L}^{-1}, L_{\rm d}^{-1}] \subset (0, +\infty)$ is a given step size and $\underline{L} \geq L_{\rm d}$. We recall that the inexact gradient is given by:

$$\nabla \bar{d}_\rho(\lambda_k) := A\bar{z}_k - b,$$

where $\bar{z}_k := \bar{z}(\lambda_k)$. The following theorem provides estimates on dual suboptimality for the scheme (**IDGM**).

THEOREM 3.2. *Suppose that the conditions of Theorem 3.1 are satisfied. Let* $\{\lambda_k\}_{k\geq 0}$ *be a sequence generated by* (**IDGM**) *and* $\{\hat{\lambda}_k\}_{k\geq 0}$ *be an average sequence derived from* $\{\lambda_k\}_{k\geq 0}$ *as* $\hat{\lambda}_k := S_k^{-1}\sum_{j=0}^k \alpha_j\lambda_{j+1}$. *Then, for any* $k \geq 1$, *the following estimate for the dual suboptimality holds:*

$$f^* - d_\rho(\hat{\lambda}_k) \leq \frac{\underline{L}R_d^2}{2(k+1)} + \left(1 + \sqrt{L_{\rm p}}R_p\right)\varepsilon_{\rm in}, \tag{3.2}$$

where we define $R_d := \|\lambda_0 - \lambda^*\|$.

Proof. Let $r_j := \|\lambda_j - \lambda^*\|$. By using (**IDGM**) and the estimates (3.1) we have:

$$\begin{aligned}
r_{j+1}^2 &= r_j^2 + 2\left\langle \lambda_{j+1} - \lambda_j, \lambda_{j+1} - \lambda^*\right\rangle - \|\lambda_{j+1} - \lambda_j\|^2\\
&\overset{(\mathbf{IDGM})}{=} r_j^2 - 2\alpha_j\left\langle \nabla\bar{d}_\rho(\lambda_j), \lambda^* - \lambda_j\right\rangle - (1 - \alpha_j L_{\rm d})\|\lambda_{j+1} - \lambda_j\|^2\\
&\quad + 2\alpha_j\left[\left\langle \nabla\bar{d}_\rho(\lambda_j), \lambda_{j+1} - \lambda_j\right\rangle - \frac{L_{\rm d}}{2}\|\lambda_{j+1} - \lambda_j\|^2\right]\\
&\overset{(3.1)}{\leq} r_j^2 + 2\alpha_j\left[\bar{d}_\rho(\lambda_j) - d_\rho(\lambda^*)\right] + 2\alpha_j\left[d_\rho(\lambda_{j+1}) - \bar{d}_\rho(\lambda_j)\right]\\
&\quad + 2\alpha_j C_Z\varepsilon_{\rm in} - (1 - \alpha_j L_{\rm d})\|\lambda_{j+1} - \lambda_j\|^2\\
&\leq r_j^2 - 2\alpha_j\left[d_\rho(\lambda^*) - d_\rho(\lambda_{j+1})\right] + 2\alpha_j C_Z\varepsilon_{\rm in}. \qquad (3.3)
\end{aligned}$$

Here the last inequality follows from $\alpha_j \in [\underline{L}^{-1}, L_{\rm d}^{-1}]$. Summing up the last inequality from $j = 0$ to k and taking into account that $d_\rho(\lambda^*) \equiv f^*$, we obtain:

$$\sum_{j=0}^k 2\alpha_j[f^* - d_\rho(\lambda_{j+1})] \leq R^2 + 2S_k C_Z\varepsilon_{\rm in}.$$

Now, by the concavity of d_ρ and the definition of $\hat{\lambda}_k$, this inequality implies:

$$S_k\left[f^* - d_\rho(\hat{\lambda}_k)\right] \leq \frac{R_{\rm d}^2}{2} + S_k C_Z\varepsilon_{\rm in}.$$

Note that $f^* - d_\rho(\hat{\lambda}_k) \geq 0$ and $S_k \geq \underline{L}^{-1}(k+1)$. The last inequality together with the definition of C_Z imply (3.2). ☐

Next, we show how we can compute an approximate solution of the primal problem (**P**). For this approximate solution we estimate the feasibility violation and the bound on the suboptimality for (**P**). Let us consider the following average sequence:

$$\hat{z}_k := S_k^{-1}\sum_{j=0}^k \alpha_j\bar{z}_j. \tag{3.4}$$

Since $\bar{z}_j \in Z$ for all $j \geq 0$ and Z is convex, then $\hat{z}_k \in Z$. From the iteration of Algorithm (**IDGM**) and (3.4), by induction, we have:

$$\lambda_{k+1} = \lambda_0 + S_k(A\hat{z}_k - b). \tag{3.5}$$

The following two theorems provide bounds on the primal infeasibility and the primal suboptimality, respectively.

THEOREM 3.3. *Under assumptions of Theorem 3.2, the sequence $\hat{z}_k$ generated by* (3.4) *satisfies the following upper bound on the infeasibility for primal problem* (**P**)*:*

$$\|A\hat{z}_k - b\| \le \nu(k, \varepsilon_{\rm in}), \tag{3.6}$$

where $\nu(k,\varepsilon_{\rm in}) := \frac{2\underline{L}R_d}{k+1} + \sqrt{\frac{2\underline{L}\left(1+\sqrt{L_{\rm p}}R_p\right)\varepsilon_{\rm in}}{k+1}}$.

Proof. From (3.3) we have:

$$\|\lambda_{j+1}-\lambda^*\|^2 \le \|\lambda_j-\lambda^*\|^2 - 2\alpha_j[d_\rho(\lambda^*)-d_\rho(\lambda_{j+1})] + 2\alpha_j C_Z\varepsilon_{\rm in} \le \|\lambda_j-\lambda^*\|^2 + 2\alpha_j C_Z\varepsilon_{\rm in}.$$

Here the last inequality follows from the fact that $d_\rho(\lambda^*) - d_\rho(\lambda_{j+1}) \ge 0$ and $\alpha_j > 0$. By induction, it follows from the above inequality that:

$$\|\lambda_{k+1} - \lambda^*\|^2 \le \|\lambda_0 - \lambda^*\|^2 + 2S_k C_Z\varepsilon_{\rm in}. \tag{3.7}$$

Now, from (3.5) we have:

$$\begin{aligned}\|\lambda_{k+1} - \lambda^*\|^2 &= \|\lambda_0 - \lambda^* + S_k(A\hat{z}_k - b)\|^2 \ge \left[\|\lambda_0 - \lambda^*\| - S_k\|A\hat{z} - b\|\right]^2\\ &= \|\lambda_0 - \lambda^*\|^2 - 2S_k\|\lambda_0 - \lambda^*\|\|A\hat{z}_k - b\| + S_k^2\|A\hat{z}_k - b\|^2.\end{aligned}$$

Substituting this inequality into (3.7) we obtain:

$$S_k^2\|A\hat{z}_k-b\|^2 - 2S_k\|\lambda_0-\lambda^*\|\|A\hat{z}_k-b\| \le 2S_k C_Z\varepsilon_{\rm in}.$$

The last inequality implies:

$$\|A\hat{z}_k - b\| \le \frac{R_{\rm d} + [R_{\rm d}^2 + 2S_k C_Z\varepsilon_{\rm in}]^{1/2}}{S_k} \le \frac{2R_{\rm d}}{S_k} + \sqrt{\frac{2C_Z\varepsilon_{\rm in}}{S_k}}$$

Note that $S_k \ge \underline{L}^{-1}(k+1)$. This inequality implies (3.6). □

THEOREM 3.4. *Under the assumptions of Theorem 3.3, the primal suboptimality can be characterized by the following lower and upper bounds:*

$$-\left[\|\lambda^*\| + \frac{\rho}{2}\nu(k,\varepsilon_{\rm in})\right]\nu(k,\varepsilon_{\rm in}) \le f(\hat{z}_k) - f^* \le \frac{\underline{L}\|\lambda_0\|^2}{2(k+1)} + \left(1 + \sqrt{L_{\rm p}}R_{\rm p}\right)\varepsilon_{\rm in}.$$

Proof. Let us first prove the left-hand side inequality. Since $f^* = d_\rho(\lambda^*)$, by using the definition of $\mathcal{L}_\rho(\hat{z}_k, \lambda^*)$ and the Cauchy-Schwartz inequality we get:

$$\begin{aligned}f^* = d_\rho(\lambda^*) &\le \mathcal{L}_\rho(\hat{z}_k, \lambda^*) = f(\hat{z}_k) + \langle\lambda^*, A\hat{z}_k - b\rangle + \frac{\rho}{2}\|A\hat{z}_k - b\|^2\\ &\le f(\hat{z}_k) + \|\lambda^*\|\|A\hat{z}_k - b\| + \frac{\rho}{2}\|A\hat{z}_k - b\|^2 \le f(\hat{z}_k) + \|\lambda^*\|\nu(k,\varepsilon_{\rm in}) + \frac{\rho}{2}\nu(k,\varepsilon_{\rm in})^2.\end{aligned}$$

Here, the last inequality follows from Theorem 3.3. In order to prove the right-hand side inequality we first use the convexity of $\mathcal{L}_\rho$ and the assumptions of Theorem 3.1:

$$\mathcal{L}_\rho(\bar{z}_j, \lambda_j) \le \mathcal{L}_\rho\left(z^*(\lambda_j), \lambda_j\right) - \langle\nabla\mathcal{L}_\rho(\bar{z}_j, \lambda_j), z^*(\lambda_j) - \bar{z}_j\rangle \le d_\rho(\lambda_j) + C_Z\varepsilon_{\rm in}.$$

Previous inequality together with the definition of $\mathcal{L}_\rho$ and $d_\rho(\lambda_j) \leq f^*$ lead to:

$$f(\bar{z}_j) + \langle \lambda_j, A\bar{z}_j - b\rangle + \frac{\rho}{2}\|A\bar{z}_j - b\|^2 - f^* \leq C_Z\varepsilon_{\text{in}}.$$

Using the iteration of Algorithm (**IDGM**) and $\alpha_j \leq \rho = L_{\text{d}}^{-1}$ we obtain:

$$\begin{aligned} f(\bar{z}_j) - f^* &\leq C_Z\varepsilon_{\text{in}} - \left\langle \lambda_j, \alpha_j^{-1}(\lambda_{j+1} - \lambda_j)\right\rangle - \frac{\rho\alpha_j^{-2}}{2}\|\lambda_{j+1} - \lambda_j\|^2 \\ &\leq \frac{1}{2\alpha_j}(\|\lambda_j\|^2 - \|\lambda_{j+1}\|^2) + C_Z\varepsilon_{\text{in}}. \end{aligned}$$

Multiplying this inequality with α_j and then summing up these inequalities from $j = 0$ to k we get:

$$\sum_{j=0}^{k}\alpha_j(f(\bar{z}_j) - f^*) \leq \frac{1}{2}\left(\|\lambda_0\|^2 - \|\lambda_{k+1}\|^2\right) + S_kC_Z\varepsilon_{\text{in}} \leq \frac{1}{2}\|\lambda_0\|^2 + S_kC_Z\varepsilon_{\text{in}}.$$

Now, using the definition of $\hat{z}_k$ and the convexity of f we can deduce that:

$$f(\hat{z}_k) - f^* \leq \frac{\|\lambda_0\|^2}{2S_k} + C_Z\varepsilon_{\text{in}}.$$

Finally, by taking into account that $S_k \geq \underline{L}(k+1)$, we get from the last estimate the right-hand side inequality. □

Let us fix the outer accuracy ε_{out}. We want to find the number of outer iterations k_{out} and a relation between ε_{out} and ε_{in} such that after this number of iterations of Algorithm (**IDGM**) the estimates $(\hat{z}_{k_{\text{out}}}, \hat{\lambda}_{k_{\text{out}}})$ satisfy (1.4). For this purpose we can choose the following values for k_{out} and ε_{in}:

$$k_{\text{out}} := \left\lfloor \frac{\underline{L}R_{\text{d}}^2}{\varepsilon_{\text{out}}} \right\rfloor \text{ and } \varepsilon_{\text{in}} := \frac{1}{2\left(1 + \sqrt{L_{\text{p}}}R_{\text{p}}\right)}\varepsilon_{\text{out}}. \tag{3.8}$$

Thus, we conclude from Theorems 3.2, 3.3 and 3.4 that for these choices of k_{out} and ε_{in} the following estimates hold:

$$\begin{aligned} &f^* - d_\rho(\hat{\lambda}_{k_{\text{out}}}) \leq \varepsilon_{\text{out}},\ \hat{z}_{k_{\text{out}}} \in Z,\ \|A\hat{z}_{k_{\text{out}}} - b\| \leq \frac{3}{R_{\text{d}}}\varepsilon_{\text{out}} \text{ and} \\ &-\left(\frac{3\|\lambda^*\|}{R_{\text{d}}} + \frac{9\rho}{2R_{\text{d}}^2}\varepsilon_{\text{out}}\right)\varepsilon_{\text{out}} \leq f(\hat{z}_{k_{\text{out}}}) - f^* \leq \left(\frac{1}{2} + \frac{\|\lambda_0\|^2}{2R_{\text{d}}^2}\right)\varepsilon_{\text{out}}. \end{aligned}$$

Finally, we are ready to summarize the above convergence rate analysis in the following algorithm.

Algorithm (*Inexact dual gradient method* (**IDGM**)).
Initialization: Choose parameters $\rho > 0$ and $0 < L_{\text{d}} \leq \underline{L}$. Choose an initial point $\lambda_0 \in \mathbb{R}^m$.
Outer iteration: For $k = 0, 1, \ldots, k_{\text{out}}$, perform:
Step 1. (*Inner loop*). For given λ_k, solve the inner problem (2.1) with accuracy ε_{in}, such that one of the stopping criterions (2.4) - (2.7) are satisfied, to obtain $\bar{z}_k$.
Step 2. Form the approximate gradient vector of d_ρ as $\nabla\bar{d}_\rho(\lambda_k) := A\bar{z}_k - b$.

Step 3. Select $\alpha_k \in [\underline{L}^{-1}, L_{\rm d}^{-1}]$ and update $S_k := \sum_{j=0}^{k} \alpha_j$.

Step 4. Update $\lambda_{k+1} := \lambda_k + \alpha_k \nabla \bar{d}_\rho(\lambda_k)$.

Output: $\hat{z}_{k_{\rm out}} := S_{k_{\rm out}}^{-1} \sum_{j=0}^{k_{\rm out}} \alpha_j \bar{z}_j$.

The penalty parameter ρ in this algorithm can be also updated adaptively by using e.g. the procedure given in [5].

3.2. Inexact dual fast gradient method. In this subsection we discuss a fast gradient scheme for solving the augmented Lagrangian dual problem $(\mathbf{D}_\rho)$. Fast gradient schemes were first proposed by Nesterov [13] and have also been discussed in the context of dual decomposition in [10]. A modification of these schemes for the case of inexact information can be also found in [4]. We shortly present such a scheme as follows. Given a positive sequence $\{\theta_k\}_{k\geq 0} \subset (0, +\infty)$ with $\theta_0 = 1$, we define $S_k := \sum_{j=0}^{k} \theta_j$. Let us assume that the sequence $\{\theta_k\}_{k\geq 0}$ satisfies $\theta_{k+1}^2 = S_{k+1}$ for all $k \geq 0$. This condition leads to:

$$\theta_{k+1} := \frac{1}{2}(1 + \sqrt{4\theta_k^2 + 1}) \;\; \forall k \geq 0 \;\; \text{and } \theta_0 := 1. \tag{3.9}$$

Note that the sequence $\{\theta_k\}_{k\geq 0}$ generated by (3.9) is increasing and satisfies:

$$0.5(k+1) \leq \theta_k \leq k+1 \;\; \forall k \geq 0. \tag{3.10}$$

We can also obtain $0.25(k+1)(k+2) < S_k < 0.5(k+1)(k+2)$ and $\sum_{j=0}^{k} S_j < (k+1)(k+2)(k+3)/3$. Now, we consider the dual fast gradient scheme as follows: given an initial point $\lambda_0 \in \mathbb{R}^m$, we define two sequences of the dual variables $\{\lambda_k\}_{k\geq 0}$ and $\{\mu_k\}_{k\geq 0}$ as:

(**IDFGM**)
$$\boxed{\left\{\begin{array}{ll} \mu_k & := \lambda_k + L_{\rm d}^{-1} \nabla \bar{d}_\rho(\lambda_k) \\ \lambda_{k+1} & := (1 - a_{k+1})\, \mu_k + a_{k+1} \Big[\lambda_0 + L_{\rm d}^{-1} \sum_{i=0}^{k} \theta_i \nabla \bar{d}_\rho(\lambda_i)\Big], \end{array}\right.}$$

where the sequence $a_{k+1} := S_{k+1}^{-1} \theta_{k+1}$.

The following lemma, which represents an extension of the results in [10, 13] to the inexact case (see also [4]), will be used to derive estimates on both primal and dual suboptimality and also primal infeasibility for the proposed method.

LEMMA 3.5. *[4, 10] Under the assumptions of Theorem 3.1, the two sequences* $\{(\lambda_k, \mu_k)\}_{k\geq 0}$ *generated by the dual fast gradient scheme* (**IDFGM**) *satisfy:*

$$\begin{aligned} S_k d_\rho(\mu_k) \geq \max_{\lambda \in \mathbb{R}^m} \left\{ \sum_{j=0}^{k} \theta_j \big[\bar{d}_\rho(\lambda_j) + \big\langle \nabla \bar{d}_\rho(\lambda_j), \lambda - \lambda_j \big\rangle\big] - \frac{L_{\rm d}}{2} \|\lambda - \lambda_0\|^2 \right\} \\ - \Big(1 + \sqrt{L_{\rm p}} R_p\Big) \varepsilon_{\rm in} \sum_{j=0}^{k} S_j \;\; \forall k \geq 0. \end{aligned} \tag{3.11}$$

The next theorem gives an estimate on dual suboptimality.

THEOREM 3.6. *Under the assumptions of Theorem 3.1, let* $\{(\lambda_k, \mu_k)\}_{k\geq 0}$ *be the two sequences generated by the scheme* (**IDFGM**). *Then, an estimate on dual suboptimality is given by the following expression:*

$$f^* - d_\rho(\mu_k) \leq \frac{2L_{\rm d} R_d^2}{(k+1)(k+2)} + \frac{4(k+3)}{3} \Big(1 + \sqrt{L_{\rm p}} R_p\Big) \varepsilon_{\rm in}.$$

Proof. By using inequality (3.1) in (3.11) we obtain:

$$S_k d_\rho(\mu_k) \geq S_k d_\rho(\lambda^*) - \frac{L_\mathrm{d}}{2}\|\lambda^* - \lambda_0\|^2 - C_Z\varepsilon_\mathrm{in}\sum_{j=0}^{k} S_j.$$

Now, using the fact that $S_k > 0.25(k+1)(k+2)$ and $\sum_{j=0}^{k} S_j < (k+1)(k+2)(k+3)/3$ and the definition of $C_Z = 1 + \sqrt{L_\mathrm{p}}R_\mathrm{p}$, we obtain our result. ☐

We further define the following primal average sequence:

$$\hat{z}_k := S_k^{-1}\sum_{j=0}^{k}\theta_j \bar{z}_i. \tag{3.12}$$

Next theorem gives an estimate on infeasibility of $\hat{z}_k$ for the original problem (**P**).

THEOREM 3.7. *Under the conditions of Theorem 3.6, the point $\hat{z}_k$ defined by* (3.12) *satisfies the following estimate on primal feasibility violation:*

$$\|A\hat{z}_k - b\| \leq v(k, \varepsilon_{in}), \tag{3.13}$$

where $v(k,\varepsilon_\mathrm{in}) := \frac{8L_\mathrm{d}R_d}{(k+1)(k+2)} + 4\sqrt{\frac{2L_\mathrm{d}(k+3)\left(1+\sqrt{L_\mathrm{p}}R_p\right)\varepsilon_\mathrm{in}}{3(k+1)(k+2)}}$.

Proof. By the definition of $\bar{d}_\rho$ and $\nabla\bar{d}_\rho$, the convexity of f and $\|\cdot\|^2$, and inequality (3.12) we have:

$$\begin{aligned}\sum_{j=0}^{k}\theta_j\left[\bar{d}_\rho(\lambda_j) + \left\langle\nabla\bar{d}_\rho(\lambda_j), \lambda-\lambda_j\right\rangle\right] &= \sum_{j=0}^{k}\theta_j f(\bar{z}_j) + S_k\left\langle\lambda, A\hat{z}_k - b\right\rangle + \sum_{j=0}^{k}\theta_j\frac{\rho}{2}\|A\bar{z}_j - b\|^2\\ &\geq S_k f(\hat{z}_k) + S_k\left\langle\lambda, A\hat{z}_k - b\right)\rangle + \frac{S_k}{2L_\mathrm{d}}\|A\hat{z}_k - b\|^2.\end{aligned}$$

Substituting this inequality into (3.11) we obtain:

$$\begin{aligned}d_\rho(\mu_k) \geq f(\hat{z}_k) &+ \max_{\lambda\in\mathbb{R}^m}\left\{\left\langle\lambda, A\hat{z}_k - b)\right\rangle - \frac{L_\mathrm{d}}{2S_k}\|\lambda-\lambda_0\|^2\right\}\\ &+ \frac{\rho}{2}\|A\hat{z}_k - b\|^2 - C_Z\varepsilon_\mathrm{in}S_k^{-1}\sum_{j=0}^{k}S_j.\end{aligned} \tag{3.14}$$

On the one hand, we can write:

$$\begin{aligned}d_\rho(\mu_k) - f(\hat{z}_k) - \frac{\rho}{2}\|A\hat{z}_k - b\|^2 &\leq d_\rho(\lambda^*) - f(\hat{z}_k) - \frac{\rho}{2}\|A\hat{z}_k - b\|^2\\ &= \min_{z\in Z}\mathcal{L}_\rho(z,\lambda^*) - f(\hat{z}_k) - \frac{\rho}{2}\|A\hat{z}_k - b\|^2 \leq \left\langle\lambda^*, A\hat{z}_k - b\right\rangle.\end{aligned} \tag{3.15}$$

On the other hand, we have:

$$\max_{\lambda\in\mathbb{R}^m}\left\{-\frac{L_\mathrm{d}}{2S_k}\|\lambda-\lambda_0\|^2 + \left\langle\lambda, A\hat{z}_k - b\right\rangle\right\} = \frac{S_k}{2L_\mathrm{d}}\|A\hat{z}_k - b\|^2 + \left\langle\lambda_0, A\hat{z}_k - b\right\rangle. \tag{3.16}$$

Substituting (3.15) and (3.16) into (3.14) we obtain:

$$\frac{S_k}{2L_\mathrm{d}}\|A\hat{z}_k - b\|^2 + \left\langle\lambda_0 - \lambda^*, A\hat{z}_k - b\right\rangle \leq C_Z\varepsilon_\mathrm{in}S_k^{-1}\sum_{j=0}^{k}S_j.$$

If we define $\xi := \|A\hat{z}_k - b\|$, then the last inequality implies $\frac{(k+1)(k+2)}{8L_{\rm d}}\xi^2 - R_{\rm d}\xi \leq \frac{4(k+3)}{3}C_Z\varepsilon_{\rm in}$. Therefore, we obtain $\xi \leq \nu(k,\varepsilon_{\rm in})$, where $\nu(\cdot,\cdot)$ is defined in (3.13). ☐

Finally, we characterize the primal suboptimality for optimization problem (**P**).

THEOREM 3.8. *Under the conditions of Theorem 3.7, the following estimates hold on primal suboptimality:*

$$-\left[\|\lambda^*\|+\frac{\rho}{2}\nu(k,\varepsilon_{\rm in})\right]\nu(k,\varepsilon_{\rm in}) \leq f(\hat{\mathbf{z}}_k)-f^* \leq \frac{2L_{\rm d}\|\lambda_0\|^2}{(k+1)(k+2)} + \frac{4(k+3)}{3}\left(1+\sqrt{L_{\rm p}}R_p\right)\varepsilon_{\rm in}.$$

Proof. The left-hand side inequality can be obtained similarly as in Theorem 3.4. We now prove the right-hand side. From (3.14) and (3.16) we have:

$$\begin{aligned} d_\rho(\mu_k) &\geq f(\hat{z}_k) + \frac{S_k}{2L_{\rm d}}\|A\hat{z}_k - b\|^2 + \langle\lambda_0, A\hat{z}_k - b\rangle + \frac{\rho}{2}\|A\hat{z}_k - b\|^2 - C_Z\varepsilon_{\rm in}S_k^{-1}\sum_{j=0}^{k}S_j \\ &\geq f(\hat{z}_k) - \frac{2L_{\rm d}}{(k+1)(k+2)}\|\lambda_0\|^2 - \frac{4(k+3)}{3}C_Z\varepsilon_{\rm in}. \end{aligned}$$

Therefore, we get:

$$f(\hat{\mathbf{z}}_k) - d_\rho(\mu_k) \leq \frac{2L_{\rm d}}{(k+1)(k+2)}\|\lambda_0\|^2 + \frac{4(k+3)}{3}C_Z\varepsilon_{\rm in}.$$

Since $d_\rho(\mu_k) \leq f^*$, we obtain the right-hand side inequality from the last relation. ☐

Similar to the previous subsection, we assume that we fix the outer accuracy $\varepsilon_{\rm out}$ and the goal is to find $k_{\rm out}$ and a relation between $\varepsilon_{\rm out}$ and $\varepsilon_{\rm in}$ such that after $k_{\rm out}$ outer iterations of the scheme (**IDFGM**) relations (1.4) holds. We can take e.g.:

$$k_{\rm out} := \left\lfloor 2R_{\rm d}\sqrt{\frac{L_{\rm d}}{\varepsilon_{\rm out}}}\right\rfloor \text{ and } \varepsilon_{\rm in} := \frac{3}{8\left(1+\sqrt{L_{\rm p}}R_{\rm p}\right)(k_{\rm out}+3)}\varepsilon_{\rm out}. \tag{3.17}$$

Using now Theorems 3.6, 3.7 and 3.8 we can conclude that the following bounds for dual suboptimality, primal infeasibility and primal suboptimality hold:

$$\begin{aligned} &f^* - d_\rho(\hat{\lambda}_{k_{\rm out}}) \leq \varepsilon_{\rm out},\ \hat{z}_{k_{\rm out}} \in Z,\ \|A\hat{z}_{k_{\rm out}} - b\| \leq \frac{3}{R_{\rm d}}\varepsilon_{\rm out} \text{ and} \\ &-\left(\frac{3\|\lambda^*\|}{R_{\rm d}} + \frac{9\rho}{2R_{\rm d}^2}\varepsilon_{\rm out}\right)\varepsilon_{\rm out} \leq f(\hat{z}_{k_{\rm out}}) - f^* \leq \left(\frac{\|\lambda_0\|^2 + R_{\rm d}^2}{2R_{\rm d}^2}\right)\varepsilon_{\rm out}. \end{aligned}$$

We can summarize the above convergence rate analysis into the following algorithm.

ALGORITHM (*Inexact dual fast gradient method* (**IDFGM**)).

Initialization: Choose parameters $\rho > 0$ and $\theta_0 := 1$. Choose an initial point $\lambda_0 \in \mathbb{R}^m$ and set $S_0 := 1$.

Outer iteration: For $k = 0, 1, \ldots, k_{\rm out}$, perform:

Step 1. (*Inner loop*). For given λ_k, solve the inner problem (2.1) with accuracy $\varepsilon_{\rm in}$, such that one of the stopping criterions (2.4) - (2.7) are satisfied, to obtain $\bar{z}_k$.

Step 2. Form the approximate gradient vector of d_ρ as $\nabla\bar{d}_\rho(\lambda_k) := A\bar{z}_k - b$.

Step 3. Update $\mu_k := \lambda_k + L_{\rm d}^{-1}\nabla\bar{d}_\rho(\lambda_k)$.

Step 4. Update $\theta_{k+1} := 0.5\left(1+\sqrt{1+4\theta_k^2}\right)$, $S_{k+1} := S_k + \theta_{k+1}$ and $a_{k+1} := S_{k+1}^{-1}\theta_{k+1}$.

Step 4. Update $\lambda_{k+1} := (1-a_{k+1})\mu_k + a_{k+1}\left[\lambda_0 + L_{\rm d}^{-1}\sum_{j=0}^{k}\theta_j \nabla \bar{d}_\rho(\lambda_j)\right]$.

Output: $\hat{z}_{k_{\rm out}} := S_{k_{\rm out}}^{-1}\sum_{j=0}^{k_{\rm out}}\theta_j \bar{z}_j$.

As in previous section, the penalty parameter ρ can be also updated adaptively by using the same procedure as before.

4. Complexity certification for linear MPC problems. In this section we discuss different implementation aspects for the application of the algorithms derived in Sections 3.1 and 3.2 in the context of state-input constrained MPC for fast linear embedded systems. We first prove that for linear MPC with quadratic stage and final costs, the augmented Lagrangian function becomes strongly convex and therefore the inner problems (2.1) can be solved in linear time with a fast gradient scheme [12]. We also discuss how the different parameters, which appear in our derived complexity bounds of Algorithms (**IDGM**) and (**IDFGM**), can be computed such that tight estimates on the total number of iterations can be derived and thus to facilitate the implementation on linear embedded systems with state-input constraints.

4.1. Implementation aspects for MPC problems. We denote by X_N a subset of the region of attraction for the MPC scheme discussed in Section 1.1. A detailed discussion on the stability of suboptimal MPC schemes can be found e.g. in [20]. For a given $x \in X_N$, we denote with $z^*(x)$ an optimal solution for $(\mathbf{P}(x))$ and with $\lambda^*(x)$ an associated optimal multiplier. Usually, in MPC problems the stage and final costs are quadratic functions of the form:

$$\ell(x_i, u_i) := x_i^T Q x_i + u_i^T R u_i \;\text{ and }\; \ell_{\rm f}(x_N) := x_N^T P x_N,$$

where the matrices Q and P are positive semidefinite and R is positive definite. Note that in our formulation we do not require strongly convex stage cost, i.e. we do not impose the matrices Q and P to be positive definite. The following lemma characterizes the convexity properties of the augmented Lagrangian function.

LEMMA 4.1. *If the optimization problem* $(\mathbf{P}(x))$ *comes from a linear MPC problem with quadratic stage and final costs, then the augmented Lagrangian* $\mathcal{L}_\rho(z,\lambda,x)$ *is a strongly convex quadratic function w.r.t. variable* z.

Proof. If we consider quadratic costs in the MPC problem (1.1), then the objective function f is quadratic, i.e. $f(z) := \frac{1}{2}z^T H z$, where the Hessian $H := \mathrm{diag}(\tilde{Q}, \tilde{R})$ is positive semidefinite, with $\tilde{Q} := \mathrm{diag}(Q, \cdots, Q, P)$ and $\tilde{R} := \mathrm{diag}(R, \cdots, R)$. Note that $\tilde{R}$ is positive definite, since we assume R to be positive definite. Using these notations, we can rewrite the augmented Lagrangian in the form (see Section 1.1):

$$\mathcal{L}_\rho(z,\lambda,x) := \frac{1}{2}z^T(H + \rho A^T A)z + (A^T\lambda - \rho A^T b(x))z - b(x)^T\lambda + \frac{\rho}{2}b(x)^T b(x).$$

It is straightforward to see that since H is positive semidefinite, then $z^T(H+\rho A^T A)z > 0$ for all z which satisfy $Az \neq 0$. On the other hand, if we consider the following set $\{z \in \mathbb{R}^n |\ Az = 0\}$, which comes from the linear dynamics, we can rewrite equivalently this set as $\left\{ z \in \mathbb{R}^n \mid z = \begin{bmatrix} \tilde{A}u \\ u \end{bmatrix}, u \in \prod_{i=1}^N U \right\}$, where $u := \begin{bmatrix} u_0^T \cdots u_{N-1}^T \end{bmatrix}^T$ and the matrix $\tilde{A}$ is obtained from the matrices A_x and B_u describing the dynamics of the system. Further, since $Az = 0$, we can write $z^T(H + \rho A^T A)z = z^T H z = u^T \tilde{A}^T \tilde{Q} \tilde{A} u + u^T \tilde{R} u > 0$ for all $u \neq 0$. The last inequality follows from the fact that $\tilde{R}$ is positive definite. In conclusion, we proved that $H + \rho A^T A$ is a positive definite matrix and therefore $\mathcal{L}_\rho(z,\lambda,x)$ is a quadratic strongly convex function in z. ∎

The previous lemma shows that in the linear MPC case with quadratic costs the objective function of the inner subproblems $\mathcal{L}_\rho$ are quadratic strongly convex in the first variable z. Moreover, $\mathcal{L}_\rho$ has also Lipschitz continuous gradient. Note that the convexity parameter σ_{p} of this function can be computed easily:

$$\sigma_{\mathrm{p}} := \lambda_{\min}(H + \rho A^T A),$$

and the Lipschitz constant L_{p} of the gradient of $\mathcal{L}_\rho$ is:

$$L_{\mathrm{p}} := \lambda_{\max}(H + \rho A^T A).$$

Note that since $\mathcal{L}_\rho(z, \lambda, x)$ is strongly convex and with Lipschitz continuous gradient in the variable z, by solving the inner problem (2.1) with a fast gradient scheme we can ensure stopping criterion (2.4) in a linear number of inner iterations [12]. Since the estimate for the number of inner iterations depends on σ_{p}, L_{p} and also on the diameter R_{p} of the set Z, we can see immediately that this diameter can be computed easily for cases when the set Z has a specific structure. Note that the set Z is a Cartesian product and thus:

$$R_{\mathrm{p}} := \sqrt{(N-1)D_x^2 + D_{x_{\mathrm{f}}}^2 + ND_u^2},$$

where D_x, $D_{x_{\mathrm{f}}}$ and D_u denotes the diameters of X, X_{f} and U, respectively. These diameters can be computed explicitly for constraints sets defined e.g. by boxes or Euclidean balls, which typically appear in the context of MPC problems.

Further, the estimates for the number of outer iterations depend on the norm of the dual optimal solution. We now discuss how we can bound $\|\lambda^*\|$ in the MPC case. We make use of the result from [3]:

LEMMA 4.2. *[3] For the family of MPC problems $(\boldsymbol{P}(x))_{x\in X_N}$ we assume that there exists $r > 0$ such that $B(0, r) \subseteq \{Az - b(x) \mid z \in Z, x \in X_N\}$, where $B(0, r)$ denotes the Euclidean ball in $\mathbb{R}^{N(n_x+n_u)}$ with center 0 and radius r. Then, the following upper bound on the norm of the dual optimal solutions of MPC problems $(\boldsymbol{P}(x))$ holds:*

$$\|\lambda^*(x)\| \leq \frac{\max_{z\in Z} \langle Hz^*(x), z - z^*(x)\rangle}{\bar{r}} \quad \forall x \in X_N,$$

where $\bar{r} := \max\big\{r \mid B(0, r) \subseteq \{AZ - b(x) \mid x \in X_N\}\big\}$.

Based on the previous lemma, in [16] upper bounds are derived on $\|\lambda^*(x)\|$ for all $x \in X_N$ for linear MPC problems with X, X_{f}, U and X_N polyhedral sets:

$$\mathcal{R}_{\mathrm{d}} \geq \max_{x\in X_N} \|\lambda^*(x)\|. \tag{4.1}$$

Recall that Lipschitz constant of the gradient of augmented dual function is $L_{\mathrm{d}} = 1/\rho$.

4.2. Total complexity of solving MPC problems. Now, we assume that we know the outer accuracy $\varepsilon_{\mathrm{out}}$ and we want to estimate the total number of iterations and also the number of flops per inner and outer iterations which have to be performed by Algorithms (**IDGM**) or (**IDFGM**) in order to solve the MPC problem $(\mathbf{P}(x))$. For both algorithms we assume the initialization $\lambda_0 = 0$ and the inner problems are solved using the stopping criterion (2.4).

First, we discuss the complexity certificates in the case when problem $(\mathbf{P}(x))$ is solved using Algorithm (**IDGM**) for all $x \in X_N$. We denote by k_{in}^G the number of inner iterations which has to be performed in order to solve each inner problem and

by k_{out}^G the number of outer iterations. From the discussion in Section 3.1 an upper bound on the number of outer iterations is given by:

$$k_{\text{out}}^G := \left\lfloor \frac{L_{\text{d}}\mathcal{R}_{\text{d}}^2}{\varepsilon_{\text{out}}} \right\rfloor. \tag{4.2}$$

Since we have proved that in the MPC case $\mathcal{L}_\rho(\cdot,\lambda,x)$ is strongly convex with convexity parameter σ_{p} and has also Lipschitz continuous gradient with constant L_{p}, in order to find a point $\bar{z}_{k_{\text{in}}^G}(\lambda)$ such that $\mathcal{L}_\rho(\bar{z}_{k_{\text{in}}^G}(\lambda),\lambda,x) - \mathcal{L}_\rho(z^*(\lambda),\lambda,x) \leq \varepsilon_{\text{in}}^2$ we can apply a fast gradient scheme. From Theorem 2.2.3 in [12] and taking into account that $\varepsilon_{\text{in}} = \frac{1}{2\left(1+\sqrt{L_{\text{p}}}R_{\text{p}}\right)}\varepsilon_{\text{out}}$ (see (3.8)) we get that the number of inner iterations for finding such a point does not exceed:

$$k_{\text{in}}^G := \left\lfloor 2\sqrt{\frac{L_{\text{p}}}{\sigma_{\text{p}}}}\ln\left(\frac{3\sqrt{L_{\text{p}}}R_{\text{p}}\left(1+\sqrt{L_{\text{p}}}R_{\text{p}}\right)}{\varepsilon_{\text{out}}}\right)\right\rfloor. \tag{4.3}$$

In the case of Algorithm (**IDFGM**), the number of outer iterations, according to the discussion in Section 3.2, is given by:

$$k_{\text{out}}^{FG} := \left\lfloor 2\mathcal{R}_{\text{d}}\sqrt{\frac{L_{\text{d}}}{\varepsilon_{\text{out}}}}\right\rfloor. \tag{4.4}$$

Taking into account that in this case we consider that the inner accuracy is chosen as $\varepsilon_{\text{in}} = \frac{3}{8\left(1+\sqrt{L_{\text{p}}}R_{\text{p}}\right)\left(k_{\text{out}}^{FG}+3\right)}\varepsilon_{\text{out}}$ (see (3.17)), then the number of inner iterations for solving each inner problem will be given by:

$$k_{\text{in}}^{FG} := \left\lfloor 2\sqrt{\frac{L_{\text{p}}}{\sigma_{\text{p}}}}\ln\left(\frac{5\sqrt{L_{\text{d}}}\mathcal{R}_d\sqrt{L_{\text{p}}}R_{\text{p}}\left(1+\sqrt{L_{\text{p}}}R_{\text{p}}\right)}{\varepsilon_{\text{out}}\sqrt{\varepsilon_{\text{out}}}}\right)\right\rfloor. \tag{4.5}$$

Further, we are also interested in finding the total number of flops for both outer and inner iterations. For solving the inner problem we use a simple fast gradient scheme for smooth strongly convex objective functions, see e.g. [12]. For this scheme, an inner iteration will require $n_{\text{in}}^{\text{flops}} := N\left(3n_x^2 + 2n_xn_u + 2n_u^2 + 10n_x + 8n_u\right)$ flops. Regarding the number of flops required by an outer iteration, the following values can be established: $n_{\text{out}}^{\text{flops,G}} := N\left(2n_x^2 + 2n_xn_u + 5n_x\right) + k_{\text{in}}^G n_{\text{in}}^{\text{flops}}$ for Algorithm (**IDGM**) and $n_{\text{out}}^{\text{flops,FG}} := N\left(2n_x^2 + 2n_xn_u + 10n_x\right) + k_{\text{in}}^{FG} n_{\text{in}}^{\text{flops}}$ for Algorithm (**IDFGM**), respectively.

5. Numerical experiments. In order to certify the efficiency of the proposed algorithms, we consider different numerical scenarios. We first analyze the behavior of Algorithms (**IDGM**) and (**IDFGM**) in terms of CPU time and number of iterations for some practical MPC problems and then we compare the CPU time, of our algorithms and of other well known QP solvers used in the context of MPC, on randomly generated QP problems. All the simulations were performed on a Laptop with CPU Intel T6670 with 2.2GHz and 4GB RAM memory, using Matlab R2008b. In all simulations we consider $\lambda_0 = 0$.

5.1. Practical MPC problems. In this section we apply the newly developed Algorithms (**IDGM**) and (**IDFGM**) on MPC problems for some practical applications, i.e. a ball on plate system and an oscillating masses system.

5.1.1. Ball on plate system. The first application discussed in this section is the ball on plate system described in [16]. We consider box constraints for states X and X_{f}, inputs U and for the region of attraction X_N as in [16], while for the stage costs we take the matrices $Q = q_1 q_1^T$, where $q_1 = [2\ 1]^T$, $R = 1$ and we compute the terminal matrix P as the solution of the LQR problem.

For different prediction horizons ranging from $N = 5$ to $N = 20$, we analyze first the behavior of Algorithms (**IDGM**) and (**IDFGM**) in terms of the number of outer iterations. For each prediction horizon length, we consider two different estimates for the number of outer iterations depending on the way we compute the upper bound on the optimal Lagrange multipliers $\lambda^*(x)$. For Algorithm (**IDGM**), k^G_{out} is the theoretical number of iterations obtained using relation (4.2) with $\mathcal{R}_{\mathrm{d}}$ computed according to [16] (see (4.1)), while $k^G_{\mathrm{out,samp}}$ is the average number of iterations obtained using our derived bound (3.8) with R_{d} computed exactly using `Gurobi 5.0.1` solver, iterations which correspond to 50 random initial states $x \in X_N$. We also compute the average number of outer iterations $k^G_{\mathrm{out,real}}$ observed in practice, obtained by imposing stopping criteria $\left| f(\hat{z}_{k^G_{\mathrm{out,real}}}) - f^* \right|$ and $\|A\hat{z}_{k^G_{\mathrm{out,real}}} - b\|$ less than accuracy $\varepsilon_{\mathrm{out}} = 10^{-3}$. For Algorithm (**IDFGM**) we compute in a similar way k^{FG}_{out} using (4.4), $k^{FG}_{\mathrm{out,samp}}$ using (3.17) and $k^{FG}_{\mathrm{out,real}}$ observed in practice. In all simulations we take $\rho = 1$. The results for both algorithms are reported in Figure 5.1. We can observe that in practice

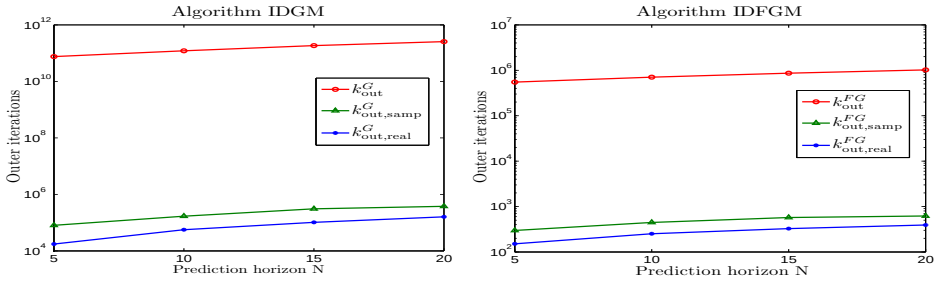

FIG. 5.1. *Variation of k^i_{out}, $k^i_{\mathrm{out,samp}}$ and $k^i_{\mathrm{out,real}}$ ($i = \{G; FG\}$) for Algorithm* (**IDGM**) *(left) and Algorithm* (**IDFGM**) *(right) w.r.t the prediction horizon N, with accuracy $\varepsilon_{\mathrm{out}} = 10^{-3}$.*

Algorithm (**IDFGM**) performs much better than Algorithm (**IDGM**). Also, we can notice that the expected number of outer iterations $k^G_{\mathrm{out,samp}}$ and $k^{FG}_{\mathrm{out,samp}}$ obtained from our derived bounds in Section 3 offer a good approximation for the real number of iterations performed by the two algorithms. Thus, these simulations show that our derived bounds in Section 3 are tight. On the other hand, k^{FG}_{out} is about three orders of magnitude, while k^G_{out} is about six orders of magnitude greater than the real number of iterations.

In Figure 5.2 we also plot the evolution of the states and inputs over the simulation horizon for a prediction horizon $N = 5$ and an outer accuracy $\varepsilon_{\mathrm{out}} = 10^{-3}$. We observe that the system is driven to the equilibrium point. Since we obtained similar trajectories for the states and inputs with both algorithms, we present only the results for Algorithm (**IDFGM**).

Since the number of outer iterations is also dependent on the way the inner accuracy $\varepsilon_{\mathrm{in}}$ is chosen, we are also interested in the behavior of the two algorithms w.r.t to $\varepsilon_{\mathrm{in}}$. For this purpose we apply Algorithms (**IDGM**) and (**IDFGM**) for solving the optimization problem ($\mathbf{P}(x)$) with a prediction horizon $N = 20$, a fixed

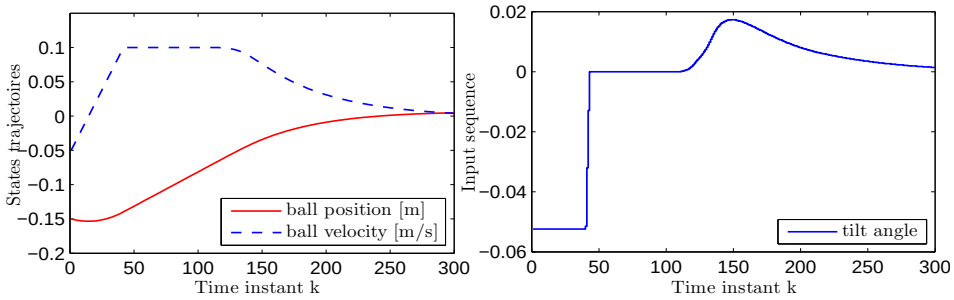

FIG. 5.2. *The trajectories of the states and inputs for a prediction horizon* $N = 5$ *obtained using Algorithm* (**IDFGM**) *with accuracy* $\varepsilon_{\text{out}} = 10^{-3}$.

outer accuracy $\varepsilon_{\text{out}} = 10^{-3}$ and varying ε_{in}. In Figure 5.3 we plot the average number of outer iterations performed by the algorithms by taken 10 random samples for the initial state $x \in X_N$. We observe that we can increase the inner accuracy ε_{in} derived in

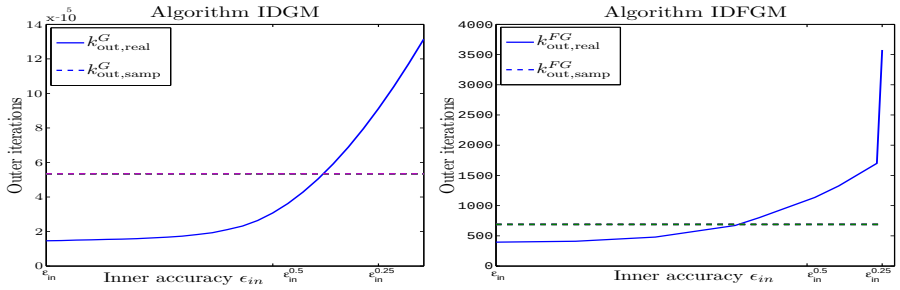

FIG. 5.3. *The number of outer iterations performed by Algorithm* (**IDGM**) *(left) and Algorithm* (**IDFGM**) *(right) with fixed outer accuracy* $\varepsilon_{\text{out}} = 10^{-3}$ *and different inner accuracies* ε_{in}.

Section 3 up to a certain value and the algorithms still perform a number of iterations less than the theoretical bounds derived in Section 3 for finding a suboptimal solution. On the other hand, if the inner accuracy is too large, the desired suboptimality cannot be ensured in a finite number of iterations. We see that Algorithm (**IDGM**) is less sensitive to the choice of the inner accuracy ε_{in} than Algorithm (**IDFGM**) due to the fact that Algorithm (**IDFGM**) accumulates errors (see Theorems 3.4 and 3.8).

In conclusion, we notice from simulations that on the one hand the Algorithm (**IDFGM**) is faster than (**IDGM**), but on the other hand that it is less robust. Thus, depending on the requirements of the application, one can choose between the two algorithms.

5.1.2. Oscillating masses. The second example is a system comprised of M oscillating masses connected by springs to each other and to walls on either sides, having $2M$ states and $M - 1$ inputs. For a detailed description of the system, its parameters and constraints see [23]. We choose a quadratic stage cost with randomly generated positive semidefinite matrices $Q \in \mathbb{R}^{2M \times 2M}$, having $\text{rank}(Q) = M$, $R = 0.1 I_M$ and the final cost $P = Q$.

For this application we are interested in the CPU time. Thus, we consider only the Algorithm (**IDFGM**), which is usually faster than Algorithm (**IDGM**). In simulations we vary the number M of masses and also the prediction horizon length N.

Further, we consider both formulations of the MPC problem: sparse QP (i.e. we keep the states as variables) and dense QP (i.e. we eliminate the states using the dynamics of the system). Our goal is to compare the performances of Algorithm (**IDFGM**) and other methods used in the framework of linear MPC. Algorithm (**IDFGM**) and `Gurobi 5.0.1` solver (**Gur1**) are used for solving the sparse formulation of the MPC problem. **Alg. 1** in [14] and `Gurobi 5.0.1` (**Gur2**) are used for solving the dense formulation of the MPC problem. In the implementation of the Algorithm (**IDFGM**) we consider an adaptive scheme in order to update the penalty parameter ρ, similar to the one presented in [5]. Since the number of iterations is sensitive to the choice of penalty parameter, we have also tuned the initial guess of ρ. For each number of masses and prediction horizon, 50 simulations were run starting from different random initial states, We have considered the accuracy $\varepsilon_{\text{out}} = 10^{-3}$ and the stopping criteria $|f(\hat{z}_k) - f^*|$ and $\|A\hat{z}_k - b\|$ less than accuracy ε_{out}.

TABLE 5.1
The average and maximum CPU time [s] (number of iterations) for Algorithm (**IDFGM**)*,* ***Alg. 1*** *in [14],* `Gurobi` *solver for sparse form (**Gur1**) and* `Gurobi` *solver for condensed form (**Gur2**).*

M	N	**IDFGM**		**Gur1**		**Alg. 1**		**Gur2**	
		avg	max	avg	max	avg	max	avg	max
5	5	0.03 (31)	0.04 (33)	0.007 (10)	0.008 (11)	0.05 (441)	0.07 (604)	0.005 (9)	0.008 (11)
5	10	0.07 (36)	0.10 (51)	0.009 (11)	0.010 (12)	0.13 (924)	0.18 (1331)	0.007 (12)	0.008 (13)
5	20	0.13 (65)	0.22 (110)	0.016 (11)	0.017 (12)	0.33 (1199)	0.65 (2383)	0.038 (12)	0.043 (13)
10	5	0.10 (28)	0.12 (30)	0.013 (10)	0.014 (12)	0.24 (1611)	0.33 (2193)	0.007 (10)	0.008 (11)
10	10	0.25 (47)	0.38 (72)	0.027 (11)	0.028 (13)	0.77 (2552)	1.34 (4449)	0.037 (12)	0.041 (13)
10	20	0.64 (70)	1.25 (135)	0.051 (12)	0.055 (13)	2.75 (2331)	5.69 (4698)	0.156 (10)	0.168 (12)
20	5	0.23 (42)	0.34 (64)	0.039 (11)	0.043 (12)	0.99 (3066)	1.45 (4481)	0.020 (11)	0.025 (13)
20	10	1.54 (98)	2.90 (193)	0.078 (12)	0.084 (13)	7.60 (5067)	18.30 (11953)	0.105 (12)	0.115 (13)
20	20	8.2 (356)	14.9 (646)	0.230 (12)	0.770 (13)	57.6 (12581)	84.7 (18504)	1.300 (12)	2.120 (12)

We can observe from Table 5.1 that Algorithm (**IDFGM**) outperforms **Alg. 1** in [14], especially when the dimension of the problem increases. On the other hand, we can notice that the solver `Gurobi 5.0.1` performs much faster than our algorithm, since the MPC problem is sparse. However, the CPU times of the two algorithms are comparable in the case of dense QP problems (see next section).

5.2. Random quadratic programming problems. In this section we compare the performance, in terms of CPU time, of Algorithms (**IDGM**) and (**IDFGM**) against some well known QP solvers used for solving MPC problems: `quadprog` (Matlab R2008b), `Sedumi` 1.3, `Cplex` 12.4 (IBM ILOG) and `Gurobi` 5.0.1.

We consider random QP problems of the form

$$\min_{\text{lb} \le z \le \text{ub}} \left\{ 0.5 z^T Q z + q^T z : \quad \text{s.t. } Az = b \right\},$$

where matrices $Q \in \mathbb{R}^{r \times n}$ and $A \in \mathbb{R}^{\lceil \frac{n}{2} \rceil \times n}$ are taken from a normal distribution with zero mean and unit variance. Matrix Q is then made positive semidefinite by transformation $Q \leftarrow Q^T Q$, having rank(Q) ranging from $0.5n$ to $0.9n$. Further, $\text{ub} = -\text{lb} = 1$ and b is taken from a uniform distribution.

We plot in Figure 5.4 the average CPU time for each solver, obtained by solving 50 random QP's for each dimension n, with an accuracy $\varepsilon_{\text{out}} = 10^{-3}$ and the stopping criteria $|f(\hat{z}_k) - f^*|$ and $\|A\hat{z}_k - b\|$ less than accuracy ε_{out}. In the case of Algorithm (**IDGM**), at each outer iteration we let the algorithm perform only $k_{\text{in}}^G = 50$ inner iterations. For the Algorithm (**IDFGM**) we consider two scenarios: in the first one, we let the algorithm to perform only $k_{\text{in}}^{FG} = 100$ inner iterations, while in the

second one we use the theoretic number of inner iterations obtained in Section 4.2 (see (4.5)). As described previously, in our algorithms we consider an adaptive scheme for updating the penalty parameter ρ, similar to the one presented in [5]. We can observe that even if the Algorithms (**IDGM**) and (**IDFGM**) are well suited for embedded applications, i.e. the implementation of the iterates is very simple, the iteration complexity is low and also the number of iterations for finding an approximate solution can be easily predicted, the computational time is comparable with the one of the other solvers used in the context of MPC. Although the obtained averaged CPU times

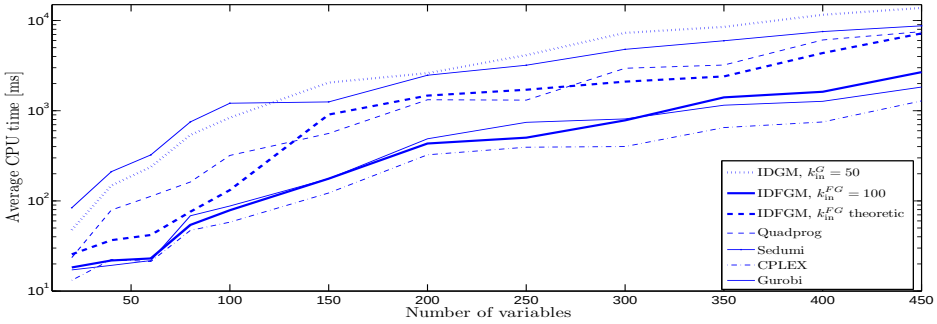

FIG. 5.4. *Average CPU time for solving QP problems of different sizes.*

are comparable, we cannot compare the exact computation complexity since for this purpose an equivalence between different stopping criteria of each solver should be studied, like e.g. the maximum violation of the constraints.

6. Conclusions. Motivated by MPC problems for fast embedded linear systems, we have proposed two dual gradient based methods for solving the augmented Lagrangian dual of a primal convex optimization problem with complicating linear constraints. We have moved the complicating constraints in the cost using augmented Lagrangian framework and solved the dual problem using gradient and fast gradient methods with inexact information. We have solved the inner subproblems only up to a certain accuracy, discussed the relations between the inner and the outer accuracy of the primal and dual problems and derived tight estimates on both primal and dual suboptimality and also on feasibility violation. We have also discussed some implementation issues of the new algorithms for embedded linear MPC problems and tested them on several examples.

Acknowledgements. The research leading to these results has received funding from: the European Union, Seventh Framework Programme (FP7-EMBOCON/2007–2013) under grant agreement no 248940; CNCS-UEFISCDI (project TE-231, no. 19/11.08.2010); ANCS (project PN II, no. 80EU/2010); Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labor, Family and Social Protection through the Financial Agreement POSDRU/89/1.5/S/62557 and POSDRU/107/1.5/S/76909; Research Council KUL: PFV/10/002 (OPTEC), GOA/10/09 - MaNet, GOA/10/11; IOF/KP/SCORES4CHEM, G.0320.08, G.0377.09; IUAP - P7 (DYSCO); FP7 - SADCO (MC ITN-264735), ERC - ST - HIGHWIND (259-166), Eurostars - SMART.

REFERENCES

[1] D.P. BERTSEKAS, *Nonlinear Programming*, 2nd ed., Athena Scientific, 1999.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.
[3] O. Devolder, F. Glineur and Y. Nesterov, *Double Smoothing Technique for Large Scale Linearly Constrained Convex Optimization*, SIAM J. Optim., 22 (2012), pp. 702–727.
[4] O. Devolder, F. Glineur and Y. Nesterov, *First-order methods of smooth convex optimization with inexact oracle*, CORE Discussion Papers 02/2011, Universite Catholique de Louvain, 2011.
[5] A. Hamdi, *Two-level primaldual proximal decomposition technique to solve large scale optimization problems*, Appl. Math. Comput., 160 (2005), pp. 921–938.
[6] M.R. Hestenes, *Multiplier and gradient methods*, Journal of Optimization and Application, 4 (1969), pp. 303–320.
[7] B. Houska, H.J. Ferreau and M. Diehl, *An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range*, Automatica J. IFAC, 47 (2011), pp. 2279–2285.
[8] J.L. Jerez, K.-V. Ling, G.A. Constantinides and E.C. Kerrigan, *Model predictive control for deeply pipelined field-programmable gate array implementation: Algorithms and circuitry*, IET Control Theory Appl., 6 (2012), pp. 1029–1041.
[9] G. Lan and R.D.C. Monteiro, *Iteration-complexity of first-order augmented Lagrangian methods for convex programming*, Technical Report, School of Industrial and Systems Engineering, Georgia Institute of Technology, 2008.
[10] I. Necoara, J. Suykens, *Application of a Smoothing Technique to Decomposition in Convex Optimization*, IEEE Trans. Automat. Control, 53 (2008), pp. 2674–2679.
[11] A. Nedic and A. Ozdaglar, *Approximate primal solutions and rate analysis for dual subgradient methods*, SIAM J. Optim, 19 (2009), pp. 1757–1780.
[12] Y. Nesterov, *Introductory lectures on convex optimization*, Springer, 2004.
[13] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program., 103 (2005), pp. 127–152.
[14] P. Patrinos and A. Bemporad, *Simple and Certifiable Quadratic Programming Algorithms for Embedded Linear Model Predictive Control*, Proc. of IFAC Nonlinear Model Predictive Control Conference, 2012.
[15] C. V. Rao, S. J. Wright and J. B. Rawlings, *Application of interior-point methods to model predictive control*, J. Optim. Theory Appl., 99 (1998), pp. 723–757.
[16] S. Richter, M. Morari and C.N. Jones, *Towards Computational Complexity Certification for Constrained MPC Based on Lagrange Relaxation and the Fast Gradient Method*, Proc. of IEEE Conference on Decision and Control, 2011.
[17] S. Richter, C.N. Jones and M. Morari, *Computational Complexity Certification for Real-Time MPC With Input Constraints Based on the Fast Gradient Method*, IEEE Trans. Automat. Control, 57 (2012), pp. 1391–1403.
[18] R.T. Rockafellar and R. Wetz, *Variational Analysis*, Springer-Verlag, 1998.
[19] R.T. Rockafellar, *Augmented Lagrangian and Applications of The Proximal Point Algorithm In Convex Programming*, Math. Oper. Res., 1 (1976), pp. 97–116.
[20] P.O.M Scokaert, D.Q. Mayne and J.B. Rawlings, *Suboptimal model predictive control (feasibility implies stability)*, IEEE Trans. Automat. Control, 44 (1999), pp. 648–654.
[21] G. Valencia-Palomo and J.A. Rossiter, *Programmable logic controller implementation of an auto-tuned predictive control based on minimal plant information*, ISA Transactions, 50 (2011), pp. 92–100.
[22] P. Zometa, M. Kogel, T. Faulwasser and R. Findeisen, *Implementation Aspects of Model Predictive Control for Embedded Systems*, Proc. of American Control Conference, 2012.
[23] Y. Wang and S. Boyd, *Fast Model Predictive Control Using Online Optimization*, IEEE Trans. Control Syst. Tech., 18 (2010), pp. 267–278.