Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Improved Fast Dual Gradient Methods for Embedded Model Predictive Control [⋆]

Pontus Giselsson [∗]

[∗] *Electrical Engineering, Stanford University*
(e-mail: pontusg@stanford.edu).

Abstract:
Recently, several authors have suggested the use of first order methods, such as fast dual ascent and the alternating direction method of multipliers, for embedded model predictive control. The main reason is that they can be implemented using simple arithmetic operations only. In this paper, we present results that enable for significant improvements when using fast dual ascent for embedded model predictive control. These improvements rely on a new characterization of the set of matrices that can be used to describe a quadratic upper bound to the negative dual function. For many interesting formulations, it is shown that the provided bound cannot be improved and that it is tighter than bounds previously presented in the literature. The improved quadratic upper bound is used in fast dual gradient methods as an approximation of the dual function. Since it better approximates the dual function than previous upper bounds, the convergence is improved. The performance enhancement is most significant for ill-conditioned problems. This is illustrated by a numerical evaluation on a AFTI-16 aircraft model where the algorithm outperforms previous proposals of first order methods for embedded control with one to three orders of magnitude.

1. INTRODUCTION

Several authors including O'Donoghue et al. (2013); Jerez et al. (2013); Richter et al. (2013); Patrinos and Bemporad (2014) have recently proposed first order optimization methods as appropriate for embedded model predictive control. In O'Donoghue et al. (2013); Jerez et al. (2013), the alternating direction method of multipliers (ADMM, see Boyd et al. (2011)) were used and high computational speeds were reported when implemented on embedded hardware. In Richter et al. (2013); Patrinos and Bemporad (2014), the optimal control problems arising in model predictive were solved using different formulations of fast dual gradient methods. In Richter et al. (2013), the equality constraints, i.e. the dynamic constraints, are dualized and a diagonal cost and box constraints are assumed. The resulting dual problem is solved using a fast gradient method. In Patrinos and Bemporad (2014), the same splitting as in O'Donoghue et al. (2013); Jerez et al. (2013) is used, but a fast gradient method is used to solve the resulting problem as opposed to ADMM in O'Donoghue et al. (2013); Jerez et al. (2013). In this paper, we will show how to generalize the fast dual gradient methods presented in Richter et al. (2013); Patrinos and Bemporad (2014) to achieve a faster convergence.

Fast gradient methods as used in Richter et al. (2013); Patrinos and Bemporad (2014) have been around since the early 80's when the seminal paper Nesterov (1983) was published. However, fast gradient methods did not render much attention before the mid 00's, after which an increased interest has emerged. Several extensions and generalizations of the fast gradient method have been proposed, e.g. in Nesterov (2003, 2005). In Beck and Teboulle (2009), the fast gradient method was generalized to allow for minimization of composite objective functions. Further, a unified framework for fast gradient methods and their generalizations were presented in Tseng (2008). To use fast gradient methods for composite minimization, one objective term should be convex and differentiable with a Lipschitz continuous gradient, while the other should be proper, closed, and convex. The former condition is equivalent to the existence of a quadratic upper bound to the function, with the same curvature in all directions. The curvature is specified by the Lipschitz constant to the gradient. In fast gradient methods, the quadratic upper bound serves as an approximation of the function to be minimized, since the bound is minimized in every iteration of the algorithm. If the quadratic upper bound does not well approximate the function to be minimized, slow convergence properties are expected. By instead allowing for a quadratic upper bound with different curvature in different directions, as in generalized fast gradient methods Zuo and Lin (2011), the bound can closer approximate the function to be minimized. For an appropriate choice of non-uniform quadratic upper bound, this can significantly improve the performance of the algorithm.

In (Nesterov, 2005, Theorem 1), a Lipschitz constant to the gradient of the dual function to strongly convex problems is presented. This result quantifies the curvature of a uniform quadratic upper bound to the negative dual function. This result was improved in (Richter et al., 2013,

[⋆] During the preparation of this paper, the author was a member of the LCCC Linnaeus Center at Lund University. Financial support from the Swedish Research Council for the author's Postdoctoral studies at Stanford University is gratefully acknowledged. Further, Eric Chu is gratefully acknowledged for constructive feedback and Alexander Domahidi is gratefully acknowledged for suggesting the AFTI-16 control problem as a benchmark.

Theorem 7) when the primal cost is restricted to being quadratic. Using these quadratic upper bounds, with the same curvature in all directions, as dual function approximation in a fast dual gradient method, may result in slow convergence. Especially for ill-conditioned problems where the upper bound does not well approximate the negative dual function. In this paper, the main result is a new characterization of the set of matrices that can be used to describe quadratic upper bounds to the negative dual function. This result generalizes and improves previous results in Nesterov (2005); Richter et al. (2013) to allow for a bound with different curvature in different directions. The provided set can be used to create a quadratic upper bound to the negative dual function with different curvature in different directions. Using this quadratic upper bound instead of the traditional uniform upper bound in fast dual gradient methods, significant improvements can be achieved. We also show that in many cases, the resulting quadratic upper bounds cannot be made tighter by using matrices outside this set.

In model predictive control, much offline computational effort can be devoted to improve the online execution time of the solver. This is done, e.g., in explicit MPC, see Bemporad et al. (2002), where the explicit parametric solution is computed beforehand, and found through a look-up table online. In this paper, the offline computational effort is devoted to choose a matrix from the provided set that minimizes the difference, in some metric, between the negative dual function and the resulting quadratic upper bound. The computed matrix is the same in all samples in the controller and can therefore be computed offline. The algorithm is evaluated on a pitch control problem in an AFTI-16 aircraft that has previously been studied in Kapasouris et al. (1990); Bemporad et al. (1997). This is a challenging problem for first order methods since it is very ill-conditioned. The numerical evaluation shows that the method presented in this paper outperforms the methods presented in O'Donoghue et al. (2013); Jerez et al. (2013); Richter et al. (2013); Patrinos and Bemporad (2014) with one to three orders of magnitude.

For space consideration, all proofs are omitted from this paper and can be found in the full version paper, Giselsson (2014).

2. PRELIMINARIES AND NOTATION

2.1 Notation

We denote by $\mathbb{R}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$, the sets of real numbers, vectors, and matrices. $\mathbb{S}^n \subseteq \mathbb{R}^{n\times n}$ is the set of symmetric matrices, and $\mathbb{S}^n_{++} \subseteq \mathbb{S}^n$, $[\mathbb{S}^n_{+}] \subseteq \mathbb{S}^n$, are the sets of positive [semi] definite matrices. Further, $L \succeq M$ and $L \succ M$ where $L, M \in \mathbb{S}^n$ denotes $L - M \in \mathbb{S}^n_+$ and $L - M \in \mathbb{S}^n_{++}$ respectively. We also use notation $\langle x, y\rangle = x^T y$, $\|x\|_2 = \sqrt{x^T x}$, and $\|x\|_H = \sqrt{x^T H x}$. Finally, $I_{\mathcal{X}}$ denotes the indicator function for the set $\mathcal{X}$, i.e. $I_{\mathcal{X}}(x) \triangleq \begin{cases} 0, & x\in\mathcal{X} \\ \infty, & \text{else}\end{cases}$.

2.2 Preliminaries

In this section, we introduce generalizations of well used concepts. We generalize the notion of strong convexity as well as the notion of Lipschitz continuity of the gradient of convex functions. We also define conjugate functions and state a known result on dual properties of a function and its conjugate.

For differentiable and convex functions $f : \mathbb{R}^n \to \mathbb{R}$ that have a Lipschitz continuous gradient with constant L, we have that

$$\|\nabla f(x_1) - \nabla f(x_2)\|_2 \leq L\|x_1 - x_2\|_2 \qquad (1)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$. This is equivalent to that

$$f(x_1) \leq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle + \frac{L}{2}\|x_1 - x_2\|_2^2 \qquad (2)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$ (Nesterov, 2003, Theorem 2.1.5). In this paper, we allow for a generalized version of the quadratic upper bound (2) to f, namely that

$$f(x_1) \leq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle + \frac{1}{2}\|x_1 - x_2\|_{\mathbf{L}}^2 \qquad (3)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$ where $\mathbf{L} \in \mathbb{S}^n_+$. The bound (2) is obtained by setting $\mathbf{L} = LI$ in (3).

Remark 1. For concave functions f, i.e. where $-f$ is convex, the Lipschitz condition (1) is equivalent to that the following quadratic lower bound

$$f(x_1) \geq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle - \frac{L}{2}\|x_1 - x_2\|_2^2 \qquad (4)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$. The generalized counterpart naturally becomes that

$$f(x_1) \geq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle - \frac{1}{2}\|x_1 - x_2\|_{\mathbf{L}}^2 \qquad (5)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$.

Next, we state a Lemma on equivalent characterizations of the condition (3).

Lemma 2. Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is convex and differentiable. The condition that

$$f(x_1) \leq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle + \frac{1}{2}\|x_1 - x_2\|_{\mathbf{L}}^2 \qquad (6)$$

holds for some $\mathbf{L} \in \mathbb{S}^n_+$ and all $x_1, x_2 \in \mathbb{R}^n$ is equivalent to that

$$\langle \nabla f(x_1) - \nabla f(x_2), x_1 - x_2\rangle \leq \|x_1 - x_2\|_{\mathbf{L}}^2. \qquad (7)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$.

The standard definition of a differentiable and strongly convex function $f : \mathbb{R}^n \to \mathbb{R}$ is that it satisfies

$$f(x_1) \geq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle + \frac{\sigma}{2}\|x_1 - x_2\|_2^2 \qquad (8)$$

for any $x_1, x_2 \in \mathbb{R}^n$, where the modulus $\sigma \in \mathbb{R}_{++}$ describes a lower bound on the curvature of the function. In this paper, the definition (8) is generalized to allow for a quadratic lower bound with different curvature in different directions.

Definition 3. A differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ is *strongly convex with matrix H* if and only if

$$f(x_1) \geq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle + \frac{1}{2}\|x_1 - x_2\|_H^2$$

holds for all $x_1, x_2 \in \mathbb{R}^n$, where $H \in \mathbb{S}^n_{++}$.

Remark 4. The traditional definition of strong convexity (8) is obtained from Definition 3 by setting $H = \sigma I$.

Lemma 5. Assume that $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable and strongly convex with matrix H. The condition that

$$f(x_1) \geq f(x_2) + \langle \nabla f(x_2), x_1 - x_2\rangle + \frac{1}{2}\|x_1 - x_2\|_H^2 \qquad (9)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$ is equivalent to that

$$\langle \nabla f(x_1) - \nabla f(x_2), x_1 - x_2 \rangle \geq \|x_1 - x_2\|_H^2 \qquad (10)$$

holds for all $x_1, x_2 \in \mathbb{R}^n$.

The condition (9) is a quadratic lower bound on the function value, while the condition (3) is a quadratic upper bound on the function value. These two properties are linked through the conjugate function

$$f^\star(y) \triangleq \sup_x \left\{ y^T x - f(x) \right\}.$$

More precisely, we have the following result.

Proposition 6. Assume that $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is closed, proper, and strongly convex with modulus σ on the relative interior of its domain. Then the conjugate function $f^\star$ is convex and differentiable, and $\nabla f^\star(y) = x^\star(y)$, where $x^\star(y) = \arg\max_x \left\{ y^T x - f(x) \right\}$. Further, $\nabla f^\star$ is Lipschitz continuous with constant $L = \frac{1}{\sigma}$.

A straight-forward generalization is given by the chain-rule and was proven in (Nesterov, 2005, Theorem 1) (which also proves the less general Proposition 6).

Corollary 7. Assume that $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is closed, proper, and strongly convex with modulus σ on the relative interior of its domain. Further, define $g^\star(y) \triangleq f^\star(Ay)$. Then $g^\star$ is convex and differentiable, and $\nabla g^\star(y) = A^T x^\star(Ay)$, where $x^\star(Ay) = \arg\max_x \left\{ (Ay)^T x - f(x) \right\}$. Further, $\nabla g^\star$ is Lipschitz continuous with constant $L = \frac{\|A\|_2^2}{\sigma}$.

For the case when $f(x) = \frac{1}{2} x^T H x + g^T x$, i.e. f is a quadratic, a tighter Lipschitz constant to $\nabla g^\star(y) = \nabla f^\star(Ay)$ was provided in (Richter et al., 2013, Theorem 7), namely $L = \|A H^{-1} A^T\|_2$.

3. PROBLEM FORMULATION

We consider optimization problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) + h(x) + g(Bx) \\ \text{subject to} & Ax = b \end{array} \qquad (11)$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^m$. We assume that the following assumption holds throughout the paper:

Assumption 8.

(a) The function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable and strongly convex with matrix H.
(b) The extended valued functions $h : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ and $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, are proper, closed, and convex.
(c) $A \in \mathbb{R}^{m \times n}$ has full row rank.

Remark 9. Examples of functions that satisfy Assumption 8(a) and 8(b) are $f(x) = \frac{1}{2} x^T H x + g^T x$ with $H \in \mathbb{S}_{++}^n$ for Assumption 8(a), and $g = I_{\mathcal{X}}$, $g = \|\cdot\|_1$, $g = I_{\mathcal{X}}^\star$, or $g = 0$ for Assumption 8(b). If Assumption 8(c) is not satisfied, redundant equality constraints can, without affecting the solution of (11), be removed to satisfy the assumption.

The optimization problem (11) can equivalently be written as

$$\begin{array}{ll} \text{minimize} & f(x) + h(x) + g(y) \\ \text{subject to} & Ax = b \\ & Bx = y \end{array} \qquad (12)$$

We introduce dual variables $\lambda \in \mathbb{R}^m$ for the equality constraints $Ax = b$ and dual variables $\mu \in \mathbb{R}^p$ for the equality constraints $Bx = y$. This gives the following Lagrange dual problem

$$\begin{aligned} &\sup_{\lambda,\mu} \inf_{x,y} \left\{ f(x) + h(x) + \lambda^T (Ax - b) + g(y) + \mu^T (Bx - y) \right\} \\ &= \sup_{\lambda,\mu} \Big[-\sup_x \left\{ (-A^T \lambda - B^T \mu)^T x - f(x) - h(x) \right\} \\ &\qquad\qquad - b^T \lambda - \sup_y \left\{ \mu^T y - g(y) \right\} \Big] \\ &= \sup_{\lambda,\mu} \left\{ -F^\star(-A^T \lambda - B^T \mu) - b^T \lambda - g^\star(\mu) \right\} \qquad (13) \end{aligned}$$

where $F^\star$ is the conjugate function to $F := f + h$ and $g^\star$ is the conjugate function to g. For ease of exposition, we introduce $\nu = (\lambda, \mu) \in \mathbb{R}^{m+p}$, $C = [A^T \ B^T]^T \in \mathbb{R}^{(m+p) \times n}$, and $c = (b, 0) \in \mathbb{R}^{m+p}$ and the following function

$$d(\nu) := -F^\star(-C^T \nu) - c^T \nu = -F^\star(-A^T \lambda - B^T \mu) - b^T \lambda. \qquad (14)$$

The function $F^\star$ is evaluated by solving an optimization problem. The minimand to this problem is denoted by

$$\begin{aligned} x^\star(\nu) &:= \arg\min_x \left\{ F(x) + \nu^T C x \right\} \qquad (15) \\ &= \arg\min_x \left\{ f(x) + h(x) + \lambda^T A x + \mu^T B x \right\}. \end{aligned}$$

From Corollary 7 we get that the function d is concave and differentiable with gradient

$$\nabla d(\nu) = C x^\star(\nu) - c$$

and that ∇d is Lipschitz continuous with constant $L = \|C\|_2^2 / \lambda_{\min}(H)$, i.e., that

$$\|\nabla d(\nu_1) - \nabla d(\nu_2)\|_2 \leq L \|\nu_1 - \nu_2\|_2 \qquad (16)$$

holds for all $\nu_1, \nu_2 \in \mathbb{R}^{m+p}$. As stated in Remark 1, (16) is equivalent to that the following quadratic lower bound to the concave function d holds for all $\nu_1, \nu_2 \in \mathbb{R}^{m+p}$

$$d(\nu_1) \geq d(\nu_2) + \langle \nabla d(\nu_2), \nu_1 - \nu_2 \rangle - \frac{L}{2} \|\nu_1 - \nu_2\|_2^2.$$

In the following section we will show that the function d satisfies the following tighter condition

$$d(\nu_1) \geq d(\nu_2) + \langle \nabla d(\nu_2), \nu_1 - \nu_2 \rangle - \frac{1}{2} \|\nu_1 - \nu_2\|_{\mathbf{L}}^2 \qquad (17)$$

for all $\nu_1, \nu_2 \in \mathbb{R}^{m+p}$ and $\mathbf{L} \in \mathbb{S}_+^{m+p}$ that satisfies $\mathbf{L} \succeq C H^{-1} C^T$.

4. DUAL FUNCTION PROPERTIES

The following theorem states that the function d defined in (14) satisfies (17) for any $\mathbf{L} \succeq C H^{-1} C^T$. The results of this section are proven in Giselsson (2014).

Theorem 10. The function d defined in (14) is concave, differentiable and satisfies

$$d(\nu_1) \geq d(\nu_2) + \langle \nabla d(\nu_2), \nu_1 - \nu_2 \rangle - \frac{1}{2} \|\nu_1 - \nu_2\|_{\mathbf{L}}^2 \qquad (18)$$

for every $\nu_1, \nu_2 \in \mathbb{R}^{m+p}$ and $\mathbf{L} \in \mathbb{S}_+^{m+p}$ that satisfies $\mathbf{L} \succeq C H^{-1} C^T$.

Next, we show that if f is a strongly convex quadratic function and h satisfies certain conditions, then Theorem 10 gives the best possible bound of the form (18).

Proposition 11. Assume that $f(x) = \frac{1}{2} x^T H x + \zeta^T x$ with $H \in \mathbb{S}_{++}^n$ and $\zeta \in \mathbb{R}^n$ and that there exists a set $\mathcal{X} \subseteq \mathbb{R}^n$ with non-empty interior on which h is linear, i.e.

$h(x) = \xi_{\mathcal{X}}^T x + \theta_{\mathcal{X}}$ for all $x \in \mathcal{X}$. Further, assume that there exists $\widetilde{\nu}$ such that $x^\star(\widetilde{\nu}) \in \text{int}(\mathcal{X})$. Then for any matrix $\mathbf{L} \not\succeq CH^{-1}C^T$, there exist ν_1 and ν_2 such that (18) does not hold.

Proposition 11 shows that the bound in Theorem 10 is indeed the best obtainable bound of the form (18) if f is a quadratic and h specifies the stated assumptions. Examples of functions that satisfy the assumptions on h in Proposition 11 include linear functions, indicator functions of closed convex constraint sets with non-empty interior, and the 1-norm.

5. FAST DUAL GRADIENT METHODS

In this section, we will describe generalized fast gradient methods and show how they can be applied to solve the dual problem (13). Generalized fast gradient methods can be applied to solve problems of the form

$$\text{minimize } \ell(x) + \psi(x) \tag{19}$$

where $x \in \mathbb{R}^n$, $\psi : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is proper, closed and convex, $\ell : \mathbb{R}^n \to \mathbb{R}$ is convex, differentiable, and satisfies

$$\ell(x_1) \leq \ell(x_2) + \langle \nabla \ell(x_2), x_1 - x_2 \rangle + \tfrac{1}{2}\|x_1 - x_2\|_{\mathbf{L}}^2 \tag{20}$$

for all $x_1, x_2 \in \mathbb{R}^n$ and some $\mathbf{L} \in \mathbb{S}_{++}^n$. Before we state the algorithm, we define the generalized prox operator

$$\text{prox}_\psi^{\mathbf{L}}(x) := \arg\min_y \left\{\psi(y) + \tfrac{1}{2}\|y - x\|_{\mathbf{L}}^2\right\} \tag{21}$$

and note that

$$\begin{aligned} &\text{prox}_\psi^{\mathbf{L}}\left(x - \mathbf{L}^{-1}\nabla\ell(x)\right) = \\ &= \arg\min_y \left\{\tfrac{1}{2}\|y - x + \mathbf{L}^{-1}\nabla\ell(x)\|_{\mathbf{L}}^2 + \psi(y)\right\} \\ &= \arg\min_y \left\{\ell(x) + \langle\nabla\ell(x), y - x\rangle + \tfrac{1}{2}\|y - x\|_{\mathbf{L}}^2 + \psi(y)\right\}. \end{aligned} \tag{22}$$

The generalized fast gradient method is stated below.

Algorithm 1.
Generalized fast gradient method

```
Set: $y^1 = x^0 \in \mathbb{R}^n, t^1 = 1$
For $k \geq 1$
    $x^k = \text{prox}_\psi^{\mathbf{L}}\left(y^k - \mathbf{L}^{-1}\nabla\ell(y^k)\right)$
    $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$
    $y^{k+1} = x^k + \left(\frac{t^k-1}{t^{k+1}}\right)(x^k - x^{k-1})$
```

The standard fast gradient method as presented in Beck and Teboulle (2009) is obtained by setting $\mathbf{L} = LI$ in Algorithm 1, where L is the Lipschitz constant to $\nabla\ell$. The main step of the fast gradient method is to perform a prox-step, i.e., to minimize (22) which can be seen as an approximation of the function $\ell + \psi$. For the standard fast gradient method, ℓ is approximated with a quadratic upper bound that has the same curvature, described by L, in all directions. If this quadratic upper bound is a bad approximation of the function to be minimized, slow convergence properties are expected. The generalization to allow for a matrix $\mathbf{L}$ in the algorithm allows for quadratic upper bounds with different curvature in different directions. This enables for quadratic upper bounds that much better approximate the function ℓ and consequently gives improved convergence.

The generalized fast gradient method has a convergence rate of (see Zuo and Lin (2011))

$$\ell_\psi(x^k) - \ell_\psi(x^\star) \leq \frac{2\|x^\star - x^0\|_{\mathbf{L}}^2}{(k+1)^2} \tag{23}$$

where $\ell_\psi := \ell + \psi$. The convergence rate of the standard fast gradient method as given in Beck and Teboulle (2009), is obtained by setting $\mathbf{L} = LI$ in (23).

The objective here is to apply the generalized fast gradient method to solve the dual problem (13). By introducing $\widetilde{g}(\nu) = g^\star([0\ I]\nu)$, the dual problem (13) can be expressed $\max_\nu d(\nu) - \widetilde{g}(\nu)$, where d is defined in (14). As shown in Theorem 10, the function $-d$ satisfies the properties required to apply generalized fast gradient methods. Namely that (20) holds for any $\mathbf{L} \in \mathbb{S}_+^{m+p}$ such that $\mathbf{L} \succeq CH^{-1}C^T$. Further, since g is a closed, proper, and convex function so is $g^\star$, see (Rockafellar, 1970, Theorem 12.2), and by (Rockafellar, 1970, Theorem 5.7) so is $\widetilde{g}$. This implies that generalized fast gradient methods, i.e. Algorithm 1, can be used to solve the dual problem (13). We set $-d = \ell$ and $\widetilde{g} = \psi$, and restrict $\mathbf{L} = \text{blkdiag}(\mathbf{L}_\lambda, \mathbf{L}_\mu)$ to get the following algorithm.

Algorithm 2.
Generalized fast dual gradient method

```
Set: $z^1 = \lambda^0 \in \mathbb{R}^m, v^1 = \mu^0 \in \mathbb{R}^p, t^1 = 1$
For $k \geq 1$
    $y^k = \arg\min_x \left\{f(x) + h(x) + (z^k)^T Ax + (v^k)^T Bx\right\}$
    $\lambda^k = z^k + \mathbf{L}_\lambda^{-1}(Ay^k - b)$
    $\mu^k = \text{prox}_{g^\star}^{\mathbf{L}_\mu}(v^k + \mathbf{L}_\mu^{-1}By^k)$
    $t^{k+1} = \frac{1+\sqrt{1+4(t^k)^2}}{2}$
    $z^{k+1} = \lambda^k + \left(\frac{t^k-1}{t^{k+1}}\right)(\lambda^k - \lambda^{k-1})$
    $v^{k+1} = \mu^k + \left(\frac{t^k-1}{t^{k+1}}\right)(\mu^k - \mu^{k-1})$
```

where y^k is the primal variable at iteration k that is used to help compute the gradient $\nabla d(\nu^k)$, where $\nu^k = (z^k, v^k)$. To arrive at the λ^k and μ^k iterations, we let $\xi^k = (\lambda^k, \mu^k)$, and note that

$$\begin{aligned} \xi^k &= \text{prox}_g^{\mathbf{L}}\left(\nu^k + \mathbf{L}^{-1}\nabla d(\nu^k)\right) \\ &= \arg\min_\nu \left\{\tfrac{1}{2}\|\nu - \nu^k - \mathbf{L}^{-1}\nabla d(\nu^k)\|_{\mathbf{L}}^2 + g^\star([0\ I]\nu)\right\} \\ &= \begin{bmatrix} \arg\min\limits_z \left\{\tfrac{1}{2}\|z - z^k - \mathbf{L}_\lambda^{-1}\nabla_z d(\nu^k)\|_{\mathbf{L}_\lambda}^2\right\} \\ \arg\min\limits_v \left\{\tfrac{1}{2}\|v - v^k - \mathbf{L}_\mu^{-1}\nabla_v d(\nu^k)\|_{\mathbf{L}_\mu}^2 + g^\star(v)\right\} \end{bmatrix} \\ &= \begin{bmatrix} z^k + \mathbf{L}_\lambda^{-1}(Ay^k - b) \\ \text{prox}_{g^\star}^{\mathbf{L}_\mu}(v^k + \mathbf{L}_\mu^{-1}By^k) \end{bmatrix}. \end{aligned} \tag{24}$$

In the following proposition we state the convergence rate properties of Algorithm 2.

Proposition 12. Suppose that Assumption 8 holds. If $\mathbf{L} = \text{blkdiag}(\mathbf{L}_\lambda, \mathbf{L}_\mu) \in \mathbb{S}_{++}^{m+p}$ is chosen such that $\mathbf{L} \succeq CH^{-1}C^T$. Then Algorithm 2 converges with the rate

$$D(\nu^\star) - D(\nu^k) \leq \frac{2\left\|\nu^\star - \nu^0\right\|_{\mathbf{L}}^2}{(k+1)^2}, \forall k \geq 1 \tag{25}$$

where $D = d - \widetilde{g}$ and k is the iteration number.

Remark 13. By forming a specific running average of previous primal variables, it is possible to prove a $O(1/k)$ convergence rate for the distance to the primal variable optimum and a $O(1/k^2)$ convergence rate for the worst case primal infeasibility, see Patrinos and Bemporad (2014).

For some choices of conjugate functions $g^\star$, $\text{prox}_{g^\star}^{\mathbf{L}_\mu}(x)$ can be difficult to evaluate. For standard prox operators (given by $\text{prox}_{g^\star}^{I}(x)$), Moreau decomposition (Rockafellar, 1970, Theorem 31.5) states that

$$\text{prox}_{g^\star}^{I}(x) + \text{prox}_{g}^{I}(x) = x.$$

In the following proposition, we will generalize this result to hold for the generalized prox-operator used here.

Proposition 14. Assume that $g : \mathbb{R}^n \to \mathbb{R}$ is a proper, closed, and convex function. Then

$$\text{prox}_{g^\star}^{\mathbf{L}}(x) + \mathbf{L}^{-1}\text{prox}_{g}^{\mathbf{L}^{-1}}(\mathbf{L}x) = x$$

for every $x \in \mathbb{R}^n$ and any $\mathbf{L} \in \mathbb{S}_{++}^n$.

Remark 15. If $g = I_{\mathcal{X}}$ where $I_{\mathcal{X}}$ is the indicator function, then $g^\star$ is the support function. Evaluating the prox operator (21) with $g^\star$ being a support function is difficult. However, through Proposition 14, this can be rewritten to only require the a projection operation onto the set $\mathcal{X}$. If $\mathcal{X}$ is a box constraint and $\mathbf{L}$ is diagonal, then the projection becomes a max-operation and hence very cheap to implement.

Remark 16. We are not restricted to have one auxiliary term g only. We can have any number of auxiliary terms g_i that all decompose according to the computations in (24), i.e., we get one prox-operation in the algorithm for every auxiliary term g_i.

6. MODEL PREDICTIVE CONTROL

In this section, we pose some standard model predictive control problems and show how they can be solved using the methods presented in this paper. The resulting algorithms will have simple arithmetic operations only which allows for easier implementation in embedded systems. We also show how to choose the $\mathbf{L}$-matrix in each case.

Example 17. We consider MPC optimization problems of the form

$$\begin{array}{lll} \text{minimize} & \sum_{t=0}^{N-1} \frac{1}{2}\left(x_t^T Q x_t + u_t^T R u_t\right) + \frac{1}{2} x_N^T Q_f x_N & \\ \text{subject to} & x_{t+1} = \Phi x_t + \Gamma u_t, & t = 0, \ldots, N-1 \\ & x_{\min} \leq x_t \leq x_{\max}, & t = 0, \ldots, N \\ & u_{\min} \leq u_t \leq u_{\max}, & t = 0, \ldots, N-1 \\ & x_0 = \bar{x} & \end{array}$$

where $\bar{x}, x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$, $\Phi \in \mathbb{R}^{n_x \times n_x}$, $\Gamma \in \mathbb{R}^{n_x \times n_u}$ and $Q \in \mathbb{S}_{++}^{n_x}$, $R \in \mathbb{S}_{++}^{n_u}$, $Q_f \in \mathbb{S}_{++}^{n_x}$ are all diagonal. Letting $y = (x_0, \ldots, x_N, u_0, \ldots, u_{N-1})$, this can be cast as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} y^T H y \\ \text{subject to} & Ay = b\bar{x} \\ & y_{\min} \leq y \leq y_{\max} \end{array}$$

where H, A, b, $y_{\min}$, and $y_{\max}$ are structured according to y. We choose $f(y) = \frac{1}{2} y^T H y$, $g = 0$, and $h = I_{\mathcal{Y}}$ where $I_{\mathcal{Y}}$ is the indicator function to

$$\mathcal{Y} = \{y \in \mathbb{R}^{(N+1)n_x + N n_u} \mid y_{\min} \leq y \leq y_{\max}\}.$$

This implicitly implies that we introduce dual variables λ for the equality constraints $Ay = b\bar{x}$. The algorithm becomes:

$$y^k = \arg\min_y \left\{\tfrac{1}{2} y^T H y + I_{\mathcal{Y}}(y) + z^T A x\right\} \tag{26}$$

$$\lambda^k = z^k + \mathbf{L}_\lambda^{-1}(Ay^k - b\bar{x}) \tag{27}$$

$$t^{k+1} = \tfrac{1 + \sqrt{1 + 4(t^k)^2}}{2} \tag{28}$$

$$z^{k+1} = \lambda^k + \left(\tfrac{t^k - 1}{t^{k+1}}\right)(\lambda^k - \lambda^{k-1}) \tag{29}$$

where the first step (26) can be implemented as

$$y^k = \max\left(\min\left(-H^{-1}A^T z^k, y_{\max}\right), y_{\min}\right) \tag{30}$$

due to the structure of the problem. From Theorem 10, we know that $\mathbf{L}_\lambda$ must satisfy $\mathbf{L}_\lambda \succeq AH^{-1}A^T$. By Assumption 8 we have that A has full row rank, which is common in MPC. Further, A is sparse in MPC which renders $\mathbf{L}_\lambda = AH^{-1}A^T$ a good choice. The algorithm requires the computation of $\mathbf{L}_\lambda^{-1}z$, where $z = Ay^k - b\bar{x}$, in each iteration. Since $\mathbf{L}_\lambda = AH^{-1}A$ is sparse, this can efficiently be implemented by offline storing the sparse Cholesky factorization $R^T R = S^T \mathbf{L}_\lambda S$, where R is sparse and upper triangular, and S is a permutation matrix. The online computation of $\mathbf{L}_\lambda^{-1}z$ then reduces to one forward and one backward solve, which can be very efficiently implemented.

The algorithm in this example is a generalization of the algorithm in Richter et al. (2013), where the matrix $\mathbf{L}$ is chosen as $\mathbf{L} = \|AH^{-1}A^T\|_2 I$. In the numerical section we will see that this generalization can significantly improve the convergence rate.

Next, we present an algorithm that works for arbitrary positive definite cost matrices, and arbitrary linear constraints.

Example 18. We consider the MPC optimization problems of the form

$$\begin{array}{lll} \text{minimize} & \sum_{t=0}^{N-1} \frac{1}{2}\left(x_t^T Q x_t + u_t^T R u_t\right) + \frac{1}{2} x_N^T Q_f x_N & \\ \text{subject to} & x_{t+1} = \Phi x_t + \Gamma u_t, & t = 0, \ldots, N-1 \\ & B_x x_t \leq d_x, & t = 0, \ldots, N-1 \\ & B_u u_t \leq d_u, & t = 0, \ldots, N-1 \\ & x_0 = \bar{x}, B_N x_N \leq d_N & \end{array}$$

where $\bar{x}, x_t \in \mathbb{R}^{n_x}$, $u_t \in \mathbb{R}^{n_u}$, $\Phi \in \mathbb{R}^{n_x \times n_x}$, $\Gamma \in \mathbb{R}^{n_x \times n_u}$, $B_x \in \mathbb{R}^{p_x \times n_x}$, $B_u \in \mathbb{R}^{p_u \times n_u}$, $B_N \in \mathbb{R}^{p_N \times n_x}$, $d_x \in \mathbb{R}^{p_x}$, $d_u \in \mathbb{R}^{p_u}$, $d_N \in \mathbb{R}^{p_N}$, $Q \in \mathbb{S}_{++}^{n_x}$,$R \in \mathbb{S}_{++}^{n_u}$, and $Q_f \in \mathbb{S}_{++}^{n_x}$. We let $y = (x_0, \ldots, x_N, u_0, \ldots, u_{N-1})$ and define $B = \text{blkdiag}(\bar{B}_x, B_N, \bar{B}_u)$ where $\bar{B}_x = \text{blkdiag}(B_x, \ldots, B_x)$ and $\bar{B}_u = \text{blkdiag}(B_u, \ldots, B_u)$. We also introduce $d = (d_x, \ldots, d_x, d_N, d_u, \ldots, d_u)$. This implies that all inequality constraints are described by $By \leq d$. Using this notation, the optimization problem can be rewritten as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} y^T H y \\ \text{subject to} & Ay = b\bar{x} \\ & By = v \\ & v \leq d \end{array}$$

We let $f(y) = \frac{1}{2} y^T H y$, $h = I_{Ay = b\bar{x}}$, and $g = I_{\mathcal{Y}}$ where $\mathcal{Y} = \{y \in \mathbb{R}^{(N+1)n_x + N n_u} \mid By \leq d\}$. Since h is the indicator function for the equality constraints $Ay = b\bar{x}$, we do not need to introduce dual variables for those

Table 1. Numerical evaluation of the dual gradient method proposed in this paper.

		exec time (ms)		nbr iters	
Algorithm	Parameters	avg.	max	avg.	max
(26)-(29)	$\mathbf{L}_\lambda = AH^{-1}A^T$	2.3	12.1	21.7	102
Richter et al. (2013)	$\mathbf{L}_\lambda = \|AH^{-1}A^T\|_2 I$	4713.9	28411	50845	308210
(31)-(34)	$\mathbf{L}_\mu = BH^{-1}B^T + 10^{-4}I$	1.4	7.2	19.0	102
Patrinos and Bemporad (2014)	$\mathbf{L}_\mu = \|BH^{-1}B^T\|_2 I$	180.3	1225	2626.7	18093
O'Donoghue et al. (2013); Jerez et al. (2013)	$\rho = 0.4$	205.8	974.3	2351.7	11229
O'Donoghue et al. (2013); Jerez et al. (2013)	$\rho = 4$	56.9	314.0	638.3	3593
O'Donoghue et al. (2013); Jerez et al. (2013)	$\rho = 40$	90.7	692.7	1027.2	7994

constraints. However, we introduce dual variables μ for $By = v$. Letting $H_A = AH^{-1}A^T$, the algorithm becomes

$$y^k = H^{-1}(A^T H_A^{-1}(AH^{-1}B^T v^k + b\bar{x}) - B^T v^k) \quad (31)$$

$$\mu^k = \operatorname{prox}_{g^\star}^{\mathbf{L}_\mu}(v^k + \mathbf{L}_\mu^{-1} B y^k) \quad (32)$$

$$t^{k+1} = \tfrac{1+\sqrt{1+4(t^k)^2}}{2} \quad (33)$$

$$v^{k+1} = \mu^k + \left(\tfrac{t^k - 1}{t^{k+1}}\right)(\mu^k - \mu^{k-1}) \quad (34)$$

where the y^k iterate follows from solving $\min_x \{f(x) + I_{Ax=b\bar{x}}(x) + (v^k)^T Bx\}$. In an implementation, the y^k-update can be implemented as in (31). Then, for efficiency, the matrix multiplications should be computed offline and stored for online use. Depending on the sparsity of H, A, and B, it might be more efficient to use the KKT-system from which (31) is deduced, namely

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} y^k \\ \xi \end{bmatrix} = \begin{bmatrix} -B^T v^k \\ b\bar{x} \end{bmatrix}.$$

Then, a sparse LDL-factorization of the KKT-matrix $\left[\begin{smallmatrix} H & A^T \\ A & 0 \end{smallmatrix}\right]$ is computed offline for online use. The online computational burden to compute the y^k-update then becomes one forward and one backward solve. Whichever method that has the lower number of flops should be chosen.

By restricting $\mathbf{L}_\mu$ to be diagonal, the second step, i.e. (32), can be implemented as

$$\mu^k = \max\left(0, v^k + \mathbf{L}_\mu^{-1}(By^k - d)\right).$$

To implement the algorithm, the matrix $\mathbf{L}_\mu$ must also be chosen. From Theorem 10 we know that $\mathbf{L}_\mu \succeq BH^{-1}B^T$. Usually in MPC, B is a thin matrix which implies that $BH^{-1}B^T$ is positive semi-definite only. One option is to choose $\mathbf{L}_\mu = BH^{-1}B^T + \epsilon I$, and to solve $\operatorname{prox}_{g^\star}^{\mathbf{L}_\mu}(v^k + \mathbf{L}_\mu^{-1}By^k)$ parametrically for fast execution. Another option is to choose $\mathbf{L}_\mu$ by solving the following semi-definite program:

$$\begin{array}{ll} \text{minimize} & \operatorname{tr} \mathbf{L}_\mu \\ \text{subject to} & \mathbf{L}_\mu \succeq BH^{-1}B^T \\ & \mathbf{L}_\mu \in \mathcal{L}_\mu \end{array}$$

where $\mathcal{L}_\mu$ describes the sparsity structure of $\mathbf{L}_\mu$, e.g., diagonal.

The splitting method used here is the same as the one used in Patrinos and Bemporad (2014). However, this is more general since we allow for $\mathbf{L}_\mu$-matrices that are not a multiple of the identity matrix. Also, the same splitting is used in O'Donoghue et al. (2013); Jerez et al. (2013), where ADMM (see Boyd et al. (2011)) is used to solve the optimization problem.

7. NUMERICAL EXAMPLE

The proposed algorithms are evaluated by applying them to the AFTI-16 aircraft model in Kapasouris et al. (1990); Bemporad et al. (1997). This problem is also a tutorial example in the MPC toolbox in MATLAB. As in Bemporad et al. (1997) and the MPC toolbox tutorial, the continuous time model from Kapasouris et al. (1990) is sampled using zero-order hold every 0.05 s. The system has four states $x = (x_1, x_2, x_3, x_4)$, two outputs $y = (y_1, y_2)$, two inputs $u = (u_1, u_2)$, and obeys the following dynamics

$$x^+ = \begin{bmatrix} 0.999 & -3.008 & -0.113 & -1.608 \\ -0.000 & 0.986 & 0.048 & 0.000 \\ 0.000 & 2.083 & 1.009 & -0.000 \\ 0.000 & 0.053 & 0.050 & 1.000 \end{bmatrix} x + \begin{bmatrix} -0.080 & -0.635 \\ -0.029 & -0.014 \\ -0.868 & -0.092 \\ -0.022 & -0.002 \end{bmatrix} u,$$

$$y = \left[\begin{smallmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{smallmatrix}\right] x$$

where x^+ denotes the state in the next time step. The dynamics, input, and output matrices are denoted by Φ, Γ, C respectively, i.e. we have $x^+ = \Phi x + \Gamma u, y = Cx$. The system is unstable, the magnitude of the largest eigenvalue of the dynamics matrix is 1.313. The outputs are the attack and pitch angles, while the inputs are the elevator and flaperon angles. The inputs are physically constrained to satisfy $|u_i| \leq 25°$, $i = 1, 2$. The outputs are soft constrained to satisfy $-s_1 - 0.5 \leq y_1 \leq 0.5 + s_2$ and $-s_3 - 100 \leq y_2 \leq 100 + s_4$ respectively, where $s = (s_1, s_2, s_3, s_4) \geq 0$ are slack variables. The cost in each time step is

$$\ell(x, u, s) = \frac{1}{2}\left((x - x_r)^T Q (x - x_r) + u^T R u + s^T S s\right)$$

where $Q = C^T Q_y C + Q_x$, where $Q_y = 10^2 I$ and $Q_x = \operatorname{diag}(10^{-4}, 0, 10^{-3}, 0)$, x_r is such that $y_r = Cx_r$ where y_r is the output reference that can vary in each step, $R = 10^{-2} I$, and $S = 10^6 I$. This gives condition number 10^{10} of the full cost matrix. Further, the terminal cost is Q, and the control and prediction horizon is $N = 10$. The numerical data in Table 1 is obtained by following a reference trajectory on the output. The objective is to change the pitch angle from 0° to 10° and then back to 0° while the angle of attack satisfies the output constraints $-0.5° \leq y_1 \leq 0.5°$. The constraints on the angle of attack limits the rate on how fast the pitch angle can be changed.

All algorithms in the comparison are implemented in MATLAB on a Linux machine using a single core running at 2.9 GHz. To create an easily transferable and fair termination criterion, the optimal solution to each optimization problem $y^\star$ is computed to high accuracy using an interior point solver. The optimality condition is $\|y^k - y^\star\|_2 / \|y^\star\|_2 \leq 0.005$, where y^k is the primal iterate in the algorithm. This implies that a relative accuracy of 0.5% of the primal solution is required.

The algorithms in Example 17, i.e. (26)-(29), and Example 18, i.e. (31)-(34), have been applied to this problem. Due to the slack variables, (30) cannot replace (26) for the y^k update. However, the y^k minimization is separable in the constraints and each of the projections can be solved by a multiparametric program with two regions. This is almost as computationally inexpensive as the y^k update in (30). Further, we use $\mathbf{L}_\lambda = AH^{-1}A^T$. Algorithm (26)-(29) is a generalization of Richter et al. (2013) that allows for general matrices $\mathbf{L}_\lambda$. The algorithm in Richter et al. (2013) is obtained by setting $\mathbf{L}_\lambda = \|AH^{-1}A^T\|_2 I$. The numerical evaluation in Table 1 reveals that this generalization improves the execution time with more than three orders of magnitude for this problem. The formulation in Example 18, i.e. (31)-(34), directly covers this MPC formulation with soft constraints. For this algorithm, we choose $\mathbf{L}_\mu = BH^{-1}B^T + 10^{-4}I$ and solve (32), i.e. $\text{prox}_{g^\star}^{\mathbf{L}_\mu}(v^k + \mathbf{L}_\mu^{-1}By^k)$, parametrically. It turns out that also this parametric program is cheap to implement as it requires one max-operation only. The resulting algorithm is a generalization of the algorithm in Patrinos and Bemporad (2014). The algorithm in Patrinos and Bemporad (2014) is given by setting $\mathbf{L}_\mu = \|BH^{-1}B^T\|_2 I$ in the iterations (31)-(34). Table 1 indicates that this generalization improves the algorithm by one to two orders of magnitude compared to Patrinos and Bemporad (2014). Further, (31)-(34) is based on the same splitting as the method in O'Donoghue et al. (2013); Jerez et al. (2013). The difference is that here, the problem is solved with a generalized dual gradient method, while in O'Donoghue et al. (2013); Jerez et al. (2013) it is solved using ADMM. In ADMM, the ρ-parameter need to be chosen. However, no exact guidelines are yet known for this choice, and the performance of the algorithm often relies heavily on this parameter. We compare our algorithm with ADMM using the best ρ that we found, $\rho = 4$, and with one larger and one smaller ρ. Table 1 reports that the execution time for our method is one to two orders of magnitude smaller (or more if the ρ-parameter in O'Donoghue et al. (2013); Jerez et al. (2013) is chosen suboptimally) than the proposed algorithm in O'Donoghue et al. (2013); Jerez et al. (2013).

8. CONCLUSIONS

We have proposed a generalization of dual fast gradient methods. This generalization allows the algorithm to, in each iteration, minimize a quadratic upper bound to the negative dual function with different curvature in different directions. This is in contrast to the standard fast dual gradient method where a quadratic upper bound to the negative dual with the same curvature in all directions is minimized in each iteration. This generalization is made possible by the main contribution of this paper that characterizes the set of matrices that can be used to describe a quadratic upper bound to the negative dual function. The numerical evaluation on an ill-conditioned aircraft problem reveals that the proposed algorithms outperform, with one to three orders of magnitude, other first order algorithms that have recently been proposed to be suitable for embedded model predictive control.

REFERENCES

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, October 2009.

A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997. ISSN 0018-9286. doi: 10.1109/9.557577.

A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, January 2002.

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

P. Giselsson. Improving fast dual ascent for MPC - Part II: The embedded case. *Automatica*, 2014. Submitted. Available `http://arxiv.org/abs/1312.3013`.

J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari. Embedded online optimization for model predictive control at megahertz rates. *IEEE Transactions on Automatic Control*, 2013. Submitted.

P. Kapasouris, M. Athans, and G. Stein. Design of feedback control systems for unstable plants with saturating actuators. In *Proceedings of the IFAC Symposium on Nonlinear Control System Design*, pages 302–307. Pergamon Press, 1990.

Y. Nesterov. A method of solving a convex programming problem with convergence rate O $(1/\mathrm{k}^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Springer Netherlands, 1st edition, 2003. ISBN 1402075537.

Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, May 2005.

B. O'Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6):2432–2442, 2013.

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

S. Richter, C. N. Jones, and M. Morari. Certification aspects of the fast gradient method for solving the dual of parametric convex programs. *Mathematical Methods of Operations Research*, 77(3):305–321, 2013.

K. T. Rockafellar. *Convex Analysis*, volume 28. Princeton Univercity Press, Princeton, NJ, 1970.

P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. Technical report. Available: `http://www.csie.ntu.edu.tw/~b97058/tseng/papers/apgm.pdf`, May 2008.

W. Zuo and Z. Lin. A generalized accelerated proximal gradient approach for total-variation-based image restoration. *IEEE Transactions on Image Processing*, 20(10):2748–2759, October 2011.