Preconditioning in Fast Dual Gradient Methods

Pontus Giselsson and Stephen Boyd

***Abstract*— First order optimization methods often perform poorly on ill-conditioned optimization problems. However, by preconditioning the problem data and solving the preconditioned problem, the performance of the first order method can be significantly improved. In this paper, we show how to compute such preconditioners when solving the dual of strongly convex optimization problems using fast dual proximal gradient methods. The proposed preconditioning is evaluated by solving ill-conditioned optimization problems that arise from controlling the pitch angle in an aircraft using model predictive control. The numerical example shows improvements of two to three orders of magnitude in the fast dual proximal gradient method compared to when no preconditioning is used.**

I. Introduction

We consider an accelerated version of the proximal gradient method, or equivalently the forward-backward splitting method, applied to solve a dual problem. The proximal gradient method (or forward-backward splitting method) can be applied to solve composite convex optimization problems of the form

$$\text{minimize } \ell(x) + \psi(x) \tag{1}$$

where ℓ is smooth, and ψ is proper, closed, and convex. The algorithm consists of first taking a gradient step (or forward step) in ℓ, then a proximal step (or backward step) in ψ, where the prox-step is defined as

$$\text{prox}_{\psi}(y) = \underset{x}{\text{argmin}} \left\{ \psi(x) + \tfrac{1}{2}\|x - y\|_2^2 \right\}.$$

The proximal gradient method becomes

$$x^{k+1} = \text{prox}_{t\psi}\left(x^k - t\nabla\ell(x^k)\right) \tag{2}$$

where t is a step-size which is optimally chosen as $t = 1/L$, where L is a Lipschitz constant to $\nabla\ell$, see [13]. A special case of the proximal gradient method is the projected gradient method [10], which is obtained when ψ is the indicator function for a constraint set. These proximal gradient methods (or forward-backward splitting methods) do not possess an optimal convergence rate, as shown in [11]. In [12], Nesterov devised an (up to a constant) optimal gradient method that later has been extended and generalized to accelerate projected gradient methods [13] and proximal gradient methods [1], [19]. The algorithmic difference between the proximal gradient method in (2) and the fast proximal gradient method in [1] is that, in the latter case, the forward-backward step is taken from the auxiliary point $y^k = x^k + \beta^k(x^k - x^{k-1})$, where β^k grows in a specific way, instead of from the previous iterate x^k. Despite the simple algorithmic modification, the theoretical convergence rate is significantly improved, and most often also the practical convergence. For more on forward-backward splitting methods and other methods suitable for solving problems of the form (1), the reader is referred to [15].

Even though fast (proximal) gradient methods often converge significantly faster than standard (proximal) gradient methods, the performance can still be unsatisfactory when applied to ill-conditioned problems. In an attempt to resolve this, we present in this paper methods to precondition the problem data such that, when the fast (proximal) gradient method is applied to the preconditioned problem, good performance is achieved. The proposed preconditioning methods are applicable when solving the dual to a strongly convex composite optimization problem. The preconditioning is based on a result in [7] that accurately describes how the shape of the dual function changes with the preconditioning. The objective is to choose a preconditioner that reshapes the dual function to, as close as possible, resemble the first-order approximation of the dual function used in the algorithm. Computation of such a preconditioner might be expensive. Hence, the intended application is code-generation for solving multi-parametric programs, where considerable time can be spent to generate the code and to compute a preconditioner that is used in all instances of the problem.

Related work include our recent papers [7], [6], in which a dual approach to the one taken here is presented. The idea in this paper is to use preconditioning to reshape the dual function to match the first order approximation in the algorithm. In [7], [6], the approximation used in the algorithm is instead reshaped to fit the shape of the dual function. In this paper, we show that these two approaches are indeed dual to each other and that they yield equivalent algorithms. This result shows that the algorithms presented in [7], [6], can be interpreted as solving a preconditioned problem using a fast dual proximal gradient method.

The preconditioning proposed in this paper is evaluated by applying it to a (ill-conditioned) multi-parametric optimization problem arising from model predictive control (MPC) of the pitch angle in an aircraft. The evaluation reveals that for such ill-conditioned problems, the the number of iterations can be reduced by two to three orders of magnitude compared to if no preconditioning is used. We also generate C code for the multi-parametric program and compare to state-of-the-art code-generator FORCES [5] which is tailored for MPC-problems. The evaluation shows that the fast dual proximal gradient method with the proposed preconditioning performs better than FORCES on the considered problem.

P. Giselsson and S. Boyd are with the Electrical Engineering Department at Stanford University. E-mail: {pontusg,boyd}@stanford.edu.

A. Notation and definitions

We denote by $\mathbf{R}$ the set of real numbers, $\mathbf{R}^n$ the set of column real numbers of length n, $\mathbf{R}^{m\times n}$ the set of $m \times n$ real matrices. Further, $\overline{\mathbf{R}} = \mathbf{R} \cup \{\infty\}$ denotes the extended real line. Moreover, $\mathbf{S}^n$ the subset of symmetric matrices in $\mathbf{R}^{n\times n}$, $\mathbf{S}^n_+$ the subset of positive semi-definite matrices in $\mathbf{S}^n$, and $\mathbf{S}^n_{++}$ the subset of positive definite matrices in $\mathbf{S}^n$. Further, $L \succeq M$ and $L \succ M$ where $L, M \in \mathbf{S}^n$ denotes $L - M \in \mathbf{S}^n_+$ and $L - M \in \mathbf{S}^n_{++}$ respectively. We consider Euclidean spaces with inner product $\langle x, y\rangle = x^T y$ and norm $\|x\|_2 = \sqrt{x^T x}$. Moreover, $I_{\mathcal{X}}$ denotes the indicator function for the set $\mathcal{X}$, i.e. $I_{\mathcal{X}}(x) \triangleq \begin{cases} 0, & x\in\mathcal{X} \\ \infty, & \text{else} \end{cases}$. The conjugate function to f is denoted f^* and is defined by $f^*(y) = \sup_x\{y^T x - f(x)\}$. Next, we define strong convexity and smoothness.

Definition 1: A function $f : \mathbf{R}^n \to \overline{\mathbf{R}}$ is *β-strongly convex* if

$$f(x) \geq f(y) + \langle u, x - y\rangle + \tfrac{\beta}{2}\|x - y\|_2^2$$

holds for all $x, y \in \mathbf{R}^n$ and $u \in \partial f(y)$.

Definition 2: A convex function $f : \mathbf{R}^n \to \mathbf{R}$ is *β-smooth* if

$$f(x) \leq f(y) + \langle \nabla f(y), x - y\rangle + \tfrac{\beta}{2}\|x - y\|_2^2$$

holds for all $x, y \in \mathbf{R}^n$.

Remark 1: An equivalent characterization that f is β-smooth is that ∇f is β-Lipschitz continuous, i.e. that

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq \|x - y\|_2 \tag{3}$$

holds for all $x, y \in \mathbf{R}^n$.

II. Preliminaries and problem formulation

The topic of this paper is preconditioning of fast dual gradient methods for solving the dual to optimization problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) + g(y) \\ \text{subject to} & Ax = y \end{array} \tag{4}$$

where $x \in \mathbf{R}^n$, $y \in \mathbf{R}^m$, and $A \in \mathbf{R}^{m\times n}$. Throughout this paper, we assume that (4) satisfies the following assumptions:

Assumption 1:

a) The function $f : \mathbf{R}^n \to \mathbf{R}$ is such that $f - \frac{1}{2}\|\cdot\|_H^2$ is 1-strongly convex.
b) The extended valued function $g : \mathbf{R}^n \to \overline{\mathbf{R}}$ is proper, closed, and convex.
c) $A \in \mathbf{R}^{m\times n}$ has full row rank.

Remark 2: Assumption 1(a) holds, e.g., for functions $f(x) = \frac{1}{2}x^T H x + \xi^T x + h(x)$ with $H \in \mathbf{S}^n_{++}$, $\xi \in \mathbf{R}^n$, and h being any proper, closed, and convex function. Functions that satisfy Assumption 1(b) include $g = I_{\mathcal{X}}$ and $g = I^*_{\mathcal{X}}$ where $\mathcal{X}$ is a closed and convex set, $g = \|\cdot\|_1$, and $g = 0$.

Since the objective is to solve (4) using a fast gradient method applied to the dual problem, we introduce dual variables $\mu \in \mathbf{R}^m$ for the equality constraints $Ax = y$. The (negated) dual problem becomes (see [18, Corollary 31.2.1])

$$\text{minimize } d(\mu) + g^*(\mu) \tag{5}$$

where

$$d(\mu) := f^*(-A^T\mu). \tag{6}$$

The following proposition, that has been proven, e.g., in [17], states that d is continuously differentiable and that ∇d is Lipschitz continuous.

Proposition 1: Suppose that Assumption 1 holds. Then d is convex and continuously differentiable with gradient $\nabla d(\mu) = -Ax^\star(\mu)$, where

$$x^\star(\mu) := \operatorname*{argmin}_x\{f(x) + \mu^T A x\}. \tag{7}$$

Further, ∇d is Lipschitz continuous with constant $L = \|AH^{-1}A^T\|_2^2$.

III. Fast dual proximal gradient methods

The dual problem (5) is of the form (1). Further, Proposition 1 and [18, Theorem 12.2] show that d and g^* in (5) satisfy the properties required to apply fast proximal gradient methods. Thus, the proximal gradient method in (2) can be applied to solve the dual problem (5). The resulting algorithm becomes:

Algorithm 1: **Fast dual proximal gradient method**

Set: $\mu^0 = \mu^{-1} \in \mathbf{R}^m, \beta^0 = 0$
For $k \geq 0$
$\nu^k = \mu^k + \beta^k(\mu^k - \mu^{k-1})$
$x^k = \operatorname{argmin}_x \left\{f(x) + (\nu^k)^T A x\right\}$
$\mu^{k+1} = \operatorname{prox}_{\frac{1}{L}g^*}\left(\nu^k + \frac{1}{L}Ax^k\right)$

In the first step of the algorithm the difference between the two previous μ^k-iterates is pre-multiplied with β^k. To guarantee fast convergence of the algorithm, these β^k must grow in a controlled way. One option is to set $\beta^k = \frac{k-1}{k+2}$, which is slightly sub-optimal. The optimal choice is $\beta^k = \theta^k(\frac{1}{\theta^{k-1}} - 1)$, where consecutive θ^k satisfy $\frac{1}{(\theta^{k-1})^2} = \frac{1-\theta^k}{(\theta^k)^2}$, see [19]. Using any of these two $\{\beta^k\}$-sequences, and $L \geq \|AH^{-1}A^T\|_2$, Algorithm 1 converges with the rate

$$D(\mu^k) - D(\mu^\star) \leq \frac{2L\left\|\mu^\star - \mu^0\right\|_2^2}{(k+2)^2} \tag{8}$$

where $D := d + g^*$ and $\mu^\star$ is an optimal solution to (5), see [19].

The second step in the algorithm is used to help computing the gradient $\nabla d(\nu^k) = -Ax^\star(\nu^k) = -Ax^k$. This x^k also serves as the primal iteration which converges to the primal optimal solution. It can also be shown, see [16], that by forming a specific running average of previous primal iterates x^k, it is possible to prove a $O(1/k)$ convergence rate for the distance to the unique primal variable optimum.

Finally, the third step is the forward-backward step or proximal gradient step described in (2). Since evaluation of the prox-operation requires the solution of an optimization problem involving g^*, the function g^* must be "simple" enough for the prox-step to be readily computed. The

forward-backward step can be expressed differently by noting that

$$\begin{aligned}&\operatorname{prox}_{\frac{1}{L}g^*}\left(\nu-\tfrac{1}{L}\nabla d(\nu)\right)=\\&=\operatorname*{argmin}_{\mu}\left\{\tfrac{1}{2}\|\mu-\nu+\tfrac{1}{L}\nabla d(\nu)\|_2^2+\tfrac{1}{L}g^*(\mu)\right\}\\&=\operatorname*{argmin}_{\mu}\left\{d(\nu)+\langle\nabla d(\nu),\mu-\nu\rangle+\tfrac{L}{2}\|\mu-\nu\|_2^2+g^*(\mu)\right\}.\end{aligned}\tag{9}$$

This implies that in Algorithm 1, the function d is approximated by the r.h.s. of the smoothness definition in Definition 2, and the algorithm converges for all L such that this r.h.s. is an upper bound to d, i.e. for all Lipschitz constants to ∇d. The r.h.s. of the smoothness definition is a quadratic function with the same curvature in all directions, specified by L. For ill-conditioned problems, i.e. problems for which d has very different curvature in different directions, this serves as a bad approximation of d. Hence, for such problems a direct application of Algorithm 1 would result in poor convergence behavior. However, by appropriately preconditioning the problem data, the function d can be reshaped to better fit the approximation used in Algorithm 1 without affecting the primal optimal solution. By solving the preconditioned problem instead of the original problem, significant improvements in the convergence of the algorithm can be achieved. In the following section we show how to compute preconditioners that enable for such improvements.

IV. Preconditioning

In this section, we show how to achieve better performance of Algorithm 1 using preconditioning. It is straightforward to verify that Algorithm 1 is invariant under a linear change of primal variables, i.e. Algorithm 1 is equivalent for any primal variable preconditioning. However, by pre-multiplying the equality constraints in (4) from the left with an invertible matrix, the algorithm changes without affecting the primal optimal solution. This can be exploited to achieve faster convergence of the algorithm.

As mentioned, in Algorithm 1 the function d is approximated by a quadratic function with the same curvature in all directions, namely the r.h.s. of the smoothness definition:

$$d(\mu)\leq d(\nu)+\langle\nabla d(\nu),\mu-\nu\rangle+\frac{L}{2}\|\mu-\nu\|_2^2\tag{10}$$

where L is a Lipschitz constant to ∇d. For ill-conditioned d this serves as a bad approximation. In this section, we will see that by appropriately preconditioning the optimization data, the function d can be reshaped to be well approximated by the r.h.s. of (10).

We consider the following equivalent formulation of (4)

$$\begin{array}{ll}\text{minimize} & f(x)+g(y)\\ \text{subject to} & EAx=Ey\end{array}\tag{11}$$

where $E\in\mathbf{R}^{m\times m}$ is an invertible preconditioner. The dual problem to (11) becomes

$$\text{minimize } d(E^T\mu)+g^*(E^T\mu).$$

By introducing

$$d_E(\mu):=d(E^T\mu)\tag{12}$$

$$g_E^*(\mu):=g^*(E^T\mu)\tag{13}$$

the dual problem to (11) can equivalently be written as

$$\text{minimize } d_E(\mu)+g_E^*(\mu).\tag{14}$$

Applying the fast proximal gradient method to (14) gives the following algorithm:

Algorithm 2:
Preconditioned fast dual proximal gradient method

Set: $\mu^0=\mu^{-1}\in\mathbf{R}^m,\beta^0=0$
For $k\geq 0$
$\nu^k=\mu^k+\beta^k(\mu^k-\mu^{k-1})$
$x^k=\operatorname{argmin}_x\left\{f(x)+(\nu^k)^TEAx\right\}$
$\mu^{k+1}=\operatorname{prox}_{\frac{1}{L}g_E^*}\left(\nu^k+\frac{1}{L}EAx^k\right)$

where L here is a Lipschitz constant to ∇d_E. Before we proceed on how to choose E to make Algorithm 2 efficient, we show that E can always be scaled to get $L=1$ without changing the algorithm.

Proposition 2: Assume that (14) is preconditioned with E and that the sequence $\{\nu^k,x^k,\mu^k\}$ is generated by Algorithm 2 with step-size $\frac{1}{L}$, where $L=\|EAH^{-1}A^TE^T\|$. Further assume that the sequence $\{\hat\nu^k,\hat x^k,\hat\mu^k\}$ is generated by Algorithm 2 using preconditioner $\hat E=\frac{1}{\sqrt{L}}E$ and with $\hat L=1$. Then, if additionally $\mu^0=\frac{1}{\sqrt{L}}\hat\mu^0$ and $\mu^{-1}=\frac{1}{\sqrt{L}}\hat\mu^{-1}$, we get $\nu^k=\frac{1}{\sqrt{L}}\hat\nu^k$, $x^k=\hat x^k$, and $\mu^k=\frac{1}{\sqrt{L}}\hat\mu^k$ for all $k\geq 0$.

Proof. Since $\sqrt{L}\mu^0=\hat\mu^0$ and $\sqrt{L}\mu^{-1}=\hat\mu^{-1}$, we have that $\nu^1=\frac{1}{\sqrt{L}}\hat\nu^1$. Further,

$$(\nu^1)^TEAx=\tfrac{1}{\sqrt{L}}(\hat\nu^1)^T\sqrt{L}\hat EAx=(\hat\nu^1)^T\hat EAx$$

which implies that $x^1=\hat x^1$. Finally,

$$\begin{aligned}\mu^1&=\operatorname{prox}_{\frac{1}{L}g_E^*}\left(\nu^1-\tfrac{1}{L}EAx^1\right)=\\&=\operatorname*{argmin}_v\left\{\tfrac{L}{2}\|v-\nu^1+\tfrac{1}{L}EAx^1\|_2^2+g_E^*(v)\right\}\\&=\operatorname*{argmin}_v\left\{\tfrac{L}{2}\|v-\tfrac{1}{\sqrt{L}}\hat\nu^1+\tfrac{1}{\sqrt{L}}\hat EA\hat x^1\|_2^2+g_E^*(v)\right\}\\&=\operatorname*{argmin}_v\left\{\tfrac{1}{2}\|\sqrt{L}v-\hat\nu^1+\hat EA\hat x^1\|_2^2+g^*(E^Tv)\right\}\\&=\tfrac{1}{\sqrt{L}}\operatorname*{argmin}_{\widetilde v}\left\{\tfrac{1}{2}\|\widetilde v-\hat\nu^1+\hat EA\hat x^1\|_2^2+g^*(\tfrac{1}{\sqrt{L}}E^T\widetilde v)\right\}\\&=\tfrac{1}{\sqrt{L}}\operatorname*{argmin}_{\widetilde v}\left\{\tfrac{1}{2}\|\widetilde v-\hat\nu^1+\hat EA\hat x^1\|_2^2+g^*(\hat E^T\widetilde v)\right\}\\&=\tfrac{1}{\sqrt{L}}\operatorname{prox}_{g_{\hat E}^*}\left(\hat\nu^1-\hat EA\hat x^1\right)=\tfrac{1}{\sqrt{L}}\hat\mu^1.\end{aligned}$$

Recursive application of these arguments gives the result. □

The preceding result shows that it is sufficient to consider preconditioners E that give a Lipschitz constant to ∇d_E that is $L=1$. Thus, it is sufficient to consider preconditioners E that satisfy

$$\|EAH^{-1}A^TE^T\|_2=1.$$

However, this relation does not indicate how to choose a preconditioner E. In the following theorem, we present a result that will indicate how this can be done. The result follows directly from [7, Theorem 11] when applied to d_E instead of d.

Theorem 1: The function d_E defined in (12) (with d defined in (6)) is convex, differentiable and satisfies

$$d_E(\mu) \leq d_E(\nu) + \langle \nabla d_E(\nu), \mu - \nu \rangle + \frac{1}{2}\|\mu - \nu\|^2_{\mathbf{L}} \quad (15)$$

for every $\mu, \nu \in \mathbf{R}^m$ and $\mathbf{L} \in \mathbf{S}^m_+$ that satisfies $\mathbf{L} \succeq EAH^{-1}A^TE^T$.

Before we proceed on how to use this result to compute preconditioners, we state two more results that are also proven in [7]. The first results shows that if f is a quadratic function plus a proper, closed, and convex function h that satisfies a certain condition, then Theorem 1 gives the best possible upper bound to d_E of the form (15). The second result gives an improved bound in the case where f is a quadratic function plus h, where h is the indicator function for an affine subspace.

Proposition 3: Assume that $f(x) = \frac{1}{2}x^THx + \xi^Tx + h(x)$ with $H \in \mathbf{S}^n_{++}$ and $\xi \in \mathbf{R}^n$ and that there exists a set $\mathcal{X} \subseteq \mathbf{R}^n$ with non-empty interior on which h (besides being proper, closed, and convex) is linear, i.e. $h(x) = \xi^T_{\mathcal{X}}x + \theta_{\mathcal{X}}$ for all $x \in \mathcal{X}$. Further, assume that there exists $\widetilde{\nu}$ such that $x^\star(\widetilde{\nu}) \in \operatorname{int}(\mathcal{X})$. Then for any matrix $\mathbf{L} \not\succeq EAH^{-1}A^TE^T$, there exist μ and ν such that (15) does not hold.

Proposition 3 shows that the bound in Theorem 1 is indeed the best obtainable bound of the form (15) if f is a quadratic+h where h specifies the stated assumptions. Examples of functions that satisfy the assumptions on h in Proposition 3 include linear functions, indicator functions of closed convex constraint sets with non-empty interior, and the 1-norm. However, indicator functions for affine subspaces do not satisfy the the assumptions of Proposition 3 since their interiors are empty (except for the trivial subspace $\mathbf{R}^n$). In the following proposition we present another result from [7] that shows how Theorem 1 can be improved in that case.

Proposition 4: Assume that $f(x) = \frac{1}{2}x^THx + \zeta^Tx + h(x)$ with $H \in \mathbf{S}^n_+$ and $\zeta \in \mathbf{R}^n$, and that $h = I_{Bx=b}$. Further assume $x^THx > 0$ whenever $x \neq 0$ and $Bx = 0$, i.e. that H is positive definite on the null-space of B. Then (15) holds for all $\mathbf{L} \in \mathbf{S}^m_+$ such that $\mathbf{L} \succeq EAK_{11}A^TE^T$ where

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} = \begin{bmatrix} H & B^T \\ B & 0 \end{bmatrix}^{-1}. \quad (16)$$

Further, for any matrix $\mathbf{L} \not\succeq EAK_{11}E^TA^T$ there exist $\mu, \nu \in \mathbf{R}^m$ such that (15) does not hold.

The preceding results provide guidance on how to choose the preconditioner E. Theorem 1 and Proposition 3 state that the (in many cases) best quadratic approximation of the form (15) to d_E is

$$d_E(\mu) \leq d_E(\nu) + \langle \nabla d_E(\nu), \mu - \nu \rangle + \tfrac{1}{2}\|\mu - \nu\|^2_{\mathbf{L}} \quad (17)$$

where $\mathbf{L} = EAH^{-1}A^TE^T$ (or $\mathbf{L} = EAK_{11}A^TE^T$). However, in Algorithm 1, the function d_E (with E scaled such that $\|EAH^{-1}A^TE^T\|_2 = 1$) is approximated by the r.h.s. of the inequality

$$d_E(\mu) \leq d_E(\nu) + \langle \nabla d_E(\nu), \mu - \nu \rangle + \tfrac{1}{2}\|\mu - \nu\|^2_2. \quad (18)$$

Thus, to get a close fit between the preconditioned dual function (17) and the first order approximation used in the algorithm, namely (18), E should be chosen such that $EAMA^TE^T = I$, where M is either $M = H^{-1}$ or $M = K_{11}$ depending on the structure of the considered problem (see Proposition 3 and Proposition 4). However, in most cases this choice is either infeasible (if A does not have full row rank, or if M is rank deficient) or it leads to too expensive prox-evaluations in the algorithm. To get as close fit between $EAMA^TE^T$ and I as possible, we propose to choose E to minimize the ratio between the largest and smallest non-zero eigenvalues of $EAMA^TE^T$ subject to structural constraints on E. For $M = H^{-1}$ and A having full row rank, all eigenvalues of $EAH^{-1}A^TE^T$ are positive, and the problem becomes to minimize the condition number

$$\text{minimize } \frac{\lambda_{\max}(EAH^{-1}A^TE^T)}{\lambda_{\min}(EAH^{-1}A^TE^T)} \quad (19)$$

subject to $E \in \mathcal{E}$, where $\mathcal{E}$ is a structure imposing set. (For many problems, the prox-operation is separable in each element, then a suitable structure in $\mathcal{E}$ is to allow for diagonal preconditioners since they does not increase the complexity when evaluating the prox-operator.) Problems of the form (19) can be posed a semi-definite programs and can therefore be readily solved for small to medium scaled problems, see [3, Section 3.1]. For cases where $EAMA^TE^T$ is rank deficient (i.e. if $M = K_{11}$, and/or A has full column rank), finding E that minimizes the ratio between the largest and smallest non-zero eigenvalues can also be posed as a semi-definite program by minimizing the condition number of QPQ^T, where $Q \in \mathbb{R}^{q\times n}$ has rank q and satisfies $AMA^T = Q^TQ$, and $P = E^TE$. This can be seen by letting $T = EQ^T \in \mathbb{R}^{n\times q}$, which implies that $TT^T = EAMA^TE^T$. Further, the non-zero eigenvalues of TT^T coincide with the non-zero eigenvalues of $T^TT = QPQ^T$, which is positive definite since P is positive definite and Q has full row rank. Thus, minimizing the condition number of $T^TT = QPQ^T$ is equivalent to minimizing the ratio between the largest and smallest non-zero eigenvalues of $TT^T = EAMA^TE^T$. How to minimize the condition number of $T^TT = QPQ^T$ using semi-definite programming is shown in [3, Section 3.1].

The proposed preconditioners require the solution of a semi-definite program, which can be time consuming to find and restricted to small to medium scale problems. In the contexts of code generation and model predictive control, this is a viable approach since the computation time needed to compute a preconditioner is often irrelevant.

V. Relation to other methods

In this paper, the optimization problem (4) is preconditioned such that the dual function d_E closely resembles the approximation used in Algorithm 2, namely the right hand

side of

$$d_E(\mu) \le d_E(\nu) + \langle \nabla d_E(\nu), \mu - \nu \rangle + \tfrac{1}{2}\|\mu - \nu\|_2^2.$$

In [6], [7], a dual approach is taken in which no preconditioning is used, i.e. the function d remains the same, but the approximation of d used in the algorithm is instead the right hand side of

$$d(\mu) \le d(\nu) + \langle \nabla d(\nu), \mu - \nu \rangle + \tfrac{1}{2}\|\mu - \nu\|_{\mathbf{L}}^2.$$

Thus, in this paper, the function d is reshaped to fit the quadratic approximation used in the algorithms, while in [6], [7], the quadratic upper bound used in the algorithm is reshaped to fit the dual function d. In this section, we will show that the generalized fast dual proximal gradient method in [6], [7] is equivalent to the preconditioned fast dual gradient method in Algorithm 2. Before we state the equivalence, we introduce the generalized prox-operator

$$\operatorname{prox}_g^{\mathbf{L}}(y) := \operatorname*{argmin}_x \left\{ g(x) + \tfrac{1}{2}\|x - y\|_{\mathbf{L}}^2 \right\}$$

and present the algorithm from [6], [7], which in [6], [7] is shown to converge for any $\mathbf{L} \succeq AH^{-1}A^T$.

Algorithm 3:

Generalized fast dual proximal gradient method

Set: $\mu^0 = \mu^{-1} \in \mathbf{R}^m, \beta^0 = 0$

For $k \ge 0$

$\nu^k = \mu^k + \beta^k(\mu^k - \mu^{k-1})$

$x^k = \operatorname{argmin}_x \left\{ f(x) + (\nu^k)^T A x \right\}$

$\mu^k = \operatorname{prox}_{g^*}^{\mathbf{L}} \left(\nu^k + \mathbf{L}^{-1} A x^k \right)$

Proposition 5: Assume that the sequence $\{\nu^k, x^k, \mu^k\}$ is generated by Algorithm 2 and that E is such that $L = \|EAH^{-1}A^TE^T\|_2 = 1$. Further assume the sequence $\{\hat{\nu}^k, \hat{x}^k, \hat{\mu}^k\}$ is generated by Algorithm 3 with $\mathbf{L} = E^{-1}E^{-T}$. Then, if additionally, $\mu^0 = E^{-T}\hat{\mu}^0$ and $\mu^{-1} = E^{-T}\hat{\mu}^{-1}$, we get $\nu^k = E^{-T}\hat{\nu}^k$, $x^k = \hat{x}^k$, and $\mu^k = E^{-T}\hat{\mu}^k$ for all $k \ge 0$.

Proof. Since $\mu^0 = E^{-T}\hat{\mu}^0$ and $\mu^{-1} = E^{-T}\hat{\mu}^{-1}$, we have that $\nu^1 = E^{-T}\hat{\nu}^1$. Further,

$$(\nu^1)^T EAx = (\hat{\nu}^1)^T E^{-T} EAx = (\hat{\nu}^1)^T Ax$$

which implies that $x^1 = \hat{x}^1$. Finally,

$$\begin{aligned}
\mu^1 &= \operatorname{prox}_{g_E^*}\left(\nu^1 - EAx^1\right) = \\
&= \operatorname*{argmin}_v \left\{ \tfrac{1}{2}\|v - \nu^1 + EAx^1\|_2^2 + g_E^*(v) \right\} \\
&= \operatorname*{argmin}_v \left\{ \tfrac{1}{2}\|E^T(v - \nu^1 + EA\hat{x}^1)\|_{\mathbf{L}}^2 + g^*(E^T v) \right\} \\
&= \operatorname*{argmin}_v \left\{ \tfrac{1}{2}\|E^T v - \hat{\nu}^1 + E^T EA\hat{x}^1\|_{\mathbf{L}}^2 + g^*(E^T v) \right\} \\
&= E^{-T} \operatorname*{argmin}_{\widetilde{v}} \left\{ \tfrac{1}{2}\|\widetilde{v} - \hat{\nu}^1 + \mathbf{L}^{-1}A\hat{x}^1\|_{\mathbf{L}}^2 + g^*(\widetilde{v}) \right\} \\
&= E^{-T}\operatorname{prox}_{g^*}^{\mathbf{L}}\left(\hat{\nu}^1 - \mathbf{L}^{-1}A\hat{x}^1\right) = E^{-T}\hat{\mu}^1.
\end{aligned}$$

Recursive application of these arguments gives the result. □

This result implies that the generalized fast dual proximal gradient methods developed in [6], [7] can be interpreted as a fast dual proximal gradient method applied to a preconditioned problem.

VI. Numerical example

The proposed preconditioning methods are evaluated by applying them to the AFTI-16 aircraft model in [9], [2]. As in [2], the continuous time model from [9] 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

$$\begin{aligned}
x^+ &= \begin{bmatrix} 0.999 & -3.008 & -0.113 & -1.608 \\ -0.000 & 0.986 & 0.048 & 0.000 \\ 0.000 & 2.083 & 1.009 & -0.000 \\ 0.000 & 0.053 & 0.050 & 1.000 \end{bmatrix} x + \begin{bmatrix} -0.080 & -0.635 \\ -0.029 & -0.014 \\ -0.868 & -0.092 \\ -0.022 & -0.002 \end{bmatrix} u, \\
y &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} x
\end{aligned}$$

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 $|u_i| \le 25^\circ$, $i = 1, 2$. The outputs are soft constrained to satisfy $-s_1 - 0.5 \le y_1 \le 0.5 + s_2$ and $-s_3 - 100 \le y_2 \le 100 + s_4$ respectively, where $s = (s_1, s_2, s_3, s_4) \ge 0$ are slack variables. The cost in each time step is

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

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 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 $-0.5^\circ \le y_1 \le 0.5^\circ$. The constraints on the angle of attack limits the rate on how fast the pitch angle can be changed.

The optimization problem can compactly be written as

$$\begin{array}{ll} \text{minimize} & \tfrac{1}{2} z^T H z + \xi_r^T z \\ \text{subject to} & Bz = b\bar{x} \\ & \underline{d} \le Cz \le \bar{d} \end{array} \tag{20}$$

where ξ_r varies with the reference, and $\bar{x}$ is (a measurement of) the current state. We will consider two different splittings to make the MPC problem (20) fit the format in (4). In the first splitting, we set f, g, and A in (4) to be

$$\begin{aligned} f(x) &:= \tfrac{1}{2} x^T H x + \xi_r^T x + I_{Bx = b\bar{x}}(x) \\ g(y) &:= I_{\underline{d} \le y \le \bar{d}}(y) \\ A &:= C \end{aligned} \tag{21}$$

This splitting has also been used in [16] in the context of fast dual gradient methods, and in [14], [8] in the context of the alternating direction method of multipliers (ADMM). For this splitting, we compare the performance of the fast dual proximal gradient method when applied to the preconditioned problem (where the preconditioner E minimizes the

TABLE I

MATLAB COMPARISON WITH AND WITHOUT PRECONDITIONING.

		exec time (ms)		nbr iters	
Alg./Split.	Precond	avg.	max	avg.	max
Alg. 2/(22)	y	2.3	12.1	21.7	102
Alg. 2/(22)	n	4713.9	28411	50845	308210
Alg. 2/(21)	y	1.2	5.8	20.0	105
Alg. 2/(21)	n	98.5	673.0	1835.8	12686

TABLE II

C COMPARISON TO STATE-OF-THE-ART SOLVERS.

		exec time (ms)	
Alg./Split	Precond	avg.	max
Alg. 2/(22)	y	0.079	0.232
Alg. 2/(21)	y	0.061	0.196
FORCES	-	0.347	0.592
MOSEK	-	4.9	5.4

condition number of $ECH^{-1}C^TE^T$) with the performance of the algorithm when applied to the non-preconditioned problem. In the second splitting, we set f, g, and A in (4) to be

$$\begin{aligned} f(x) &:= \tfrac{1}{2}x^THx + \xi_r^Tz + I_{\underline{d}\leq Cx\leq \bar{d}}(x) \\ g(y) &:= I_{y=b\bar{x}}(y) \\ A &:= B \end{aligned} \tag{22}$$

which has been used in [17]. Also here, we compare the algorithm performance when applied to the preconditioned problem (where the preconditioner E minimizes the condition number of $EBH^{-1}B^TE^T$) with the performance when solving the original problem. In the second splitting, we have no structural constraints on E, since $\text{prox}_{g_E^*}$ can be computed explicitly with the same computational cost for any structure of E. Thus we choose E to minimize the condition number of $EBH^{-1}B^TE^T$, i.e., E is chosen to satisfy $EBH^{-1}B^TE^T = I$.

In Table I, the proposed preconditioning is evaluated by comparing it to the original, not preconditioned, case. In this example, improvements of two to three orders of magnitude (depending on which splitting that is used) are achieved when solving the preconditioned problems compared to when solving the original, not preconditioned, problems. In Table II, C implementations of Algorithm 2 that solve the preconditioned problems using splitting (21) and (22) respectively are compared to other C solvers. The table reveals that the first order methods are two to three times faster than the code-generator FORCES [5], which uses an interior point method that is tailored to the specific structure that arise in MPC optimization problems. Table II also shows that the C implementations of Algorithm 2 significantly outperform the C solver MOSEK, which does not have the advantage of generating problem specific code before the problems are solved.

VII. CONCLUSIONS

We have proposed a preconditioning method for fast dual proximal gradient methods that is suitable for solving optimization problems arising in model predictive control. Our numerical evaluation shows that by preconditioning the problem to be solved, the algorithm performance can be improved with up to two to three orders of magnitude, compared to when no preconditioning is used. The numerical evaluation further shows that a C implementation of the fast dual gradient method when solving the preconditioned problem can perform better than state of the art code generators for MPC problems.

REFERENCES

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

[2] 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.

[3] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, June 1994.

[4] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM journal on Multiscale Modeling and Simulation*, 4(4):1168–1200, 2005.

[5] A. Domahidi, A. Zgraggen, M.N. Zeilinger, M. Morari, and C.N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *IEEE Conference on Decision and Control (CDC)*, pages 668–674, Maui, HI, USA, December 2012.

[6] P. Giselsson. Improved dual decomposition for distributed model predictive control. In *Proceedings of 2014 IFAC World Congress*, Cape Town, South Africa, August 2014. Accepted for publication. Available `https://www.control.lth.se/Staff/PontusGiselsson.html`.

[7] P. Giselsson. Improved fast dual gradient methods for embedded model predictive control. In *Proceedings of 2014 IFAC World Congress*, Cape Town, South Africa, 2014. Accepted for publication. Available `https://www.control.lth.se/Staff/PontusGiselsson.html`.

[8] 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.

[9] 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.

[10] E. S. Levitin and B. T. Polyak. Constrained minimization problems. *USSR Comput. Math and Math. Phys.*, 6:1–50, 1966. (Engl. transl. of paper i Zh. Vychisl. Mat. i Mat. Fiz., vol. 6, pp. 787–823, 1965).

[11] A Nemirovsky and D Yudin. *Informational Complexity and Efficient Methods for Solution of Convex Extremal Problems*. Wiley, New York, NY, 1983.

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

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

[14] 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.

[15] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[16] 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.

[17] 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.

[18] R. T. Rockafellar. *Convex Analysis*, volume 28. Princeton Univercity Press, Princeton, NJ, 1970.

[19] 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.