Optimization and Engineering manuscript No.
(will be inserted by the editor)

Soft Inequality Constraints in Gradient Method and Fast Gradient Method For Quadratic Programming

Matija Perne · **Samo Gerkšič** · **Boštjan Pregelj**

Received: date / Accepted: date

Abstract A quadratic program (QP) with soft inequality constraints with both linear and quadratic cost on constraint violation can be solved with the dual gradient method (GM) or the dual fast gradient method (FGM). The way the constraint violation is treated influences the efficiency and the usefulness of the algorithm. The classical way suggests extending the QP using a vector of slack variables. We demonstrate a way of obtaining the solution to the soft-constrained QP without explicitly introducing slack variables. The presented approach is more efficient than solving the extended QP with GM or FGM and results in an algorithm very similar to GM or FGM for the QP if the soft constraints are replaced with hard ones. The approach is intended for applications in model predictive control (MPC) with fast system dynamics, where QPs of this type are repetitively solved at every sampling time in the millisecond range.

Keywords Model Predictive Control · Quadratic Programming · First-Order Methods · soft constraints · KKT optimality conditions

Mathematics Subject Classification (2000) 90C20 · 49N05 · 93C05 · 65K10 · 49K20

Research supported by Slovenian Research Agency (P2-0001). This work has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

M. Perne
Jožef Stefan Institute, Jamova cesta 39, Ljubljana, Slovenia, Tel.: +386-1-477-3800, E-mail: matija.perne@ijs.si

Samo Gerkšič · Boštjan Pregelj
Jožef Stefan Institute, Jamova cesta 39, Ljubljana, Slovenia

1 Introduction

Real-time linear model predictive control (MPC) typically requires solving a convex quadratic program (QP) at every sample (Qin and Badgwell, 2003). In the recent decade, a number of approaches to fast online solution of MPC-derived QPs were proposed (Domahidi et al, 2012; Ferreau et al, 2008, 2014; Hartley et al, 2014; Mattingley and Boyd, 2012; Mattingley et al, 2011; Wang and Boyd, 2010) with the aim of making MPC useful for control of systems with faster dynamics. First-order methods (Giselsson, 2014a; Giselsson and Boyd, 2015; Kouzoupis, 2014; Patrinos et al, 2015; Richter, 2012) such as the fast gradient method (FGM) are a promising group of methods for fast MPC. As the necessary precision is relatively low (Mattingley and Boyd, 2012), they can calculate a sufficiently accurate solution fast in spite of their low convergence rate. For even faster computation, they are implementable using field-programmable gate arrays (FPGA) (Gerkšič et al, 2018) since the iterations are relatively simple. Their complexity certification possibilities are good (Richter, 2012). In the presence of state or output constraints, when the primal form of FGM would require inefficient projections on sets that are not simple, the dual form of FGM can still be used (Borrelli et al, 2015).

While QPs arising from MPC with hard state or output constraints are not guaranteed to be feasible (Wang and Boyd, 2010) and may not have an optimum, reasonable behaviour of an MPC controller is typically required in all circumstances (Mattingley et al, 2011; Qin and Badgwell, 2003). One of the ways of mitigating infeasibility, which is adequate in many practical control applications, is by softening the state and/or output constraints with slack variables and augmenting the cost function with penalties on the slack variables (de Oliveira and Biegler, 1994; Zafiriou and Chiou, 1993; Zheng and Morari, 1995). The augmented QP obtained in this way is guaranteed to be feasible. With suitable slack variable penalties, it can be designed in such a way that its optimum is equal or close to the optimum of the original QP when it exists (Hovd and Stoican, 2014; Kerrigan and Maciejowski, 2000). When the original QP is infeasible, the augmented one behaves in a sensible manner, e.g. violating softened constraints to a reasonable degree while enforcing physical input constraints that remain hard (Afonso and Galvo, 2012; Zafiriou and Chiou, 1993). However, the softly constrained QP with slack variables has higher QP dimensions and is computationally significantly more demanding to solve than the original QP. Because the slack variables are highly correlated with the corresponding state or output variables, the augmented QP is ill-suited for solving with first-order methods without adaptation (Jerez et al, 2014).

It is possible to derive an algorithm that solves the soft-constrained QP and does not use much more resources than the dual gradient method (GM) or the dual FGM applied to the hard-constrained QP. The solver code generator *QPgen* (Giselsson, 2014b; Giselsson and Boyd, 2014) solves the soft-constrained QP by an iteration scheme that is very similar to dual GM or dual FGM applied to the original QP with a modified proximity operator, however it is limited to the special case of only linear cost on the soft constraint violation. Kouzoupis (Kouzoupis, 2014) derives the required modification of the proximity operator for a form of dual FGM with both linear and quadratic cost on the soft constraint violation for a sparsely structured QP, where the optimization vector comprises the control input, the system state and the output with box inequality constraints, where Lagrange relaxation is used for both equality and inequality constraints.

Our goal is to form an efficient implementation of soft constraints with linear and quadratic costs on the constraint violation in the dual GM and the dual FGM algorithm suitable for use with the condensed formulation as used in *QPgen*. These algorithms were found to be well-suited for the implementation of MPC for plasma magnetic control in tokamak fusion reactors where real-time MPC is challenging due to fast dynamics, where moderately-sized QPs must be solved with relatively low precision repetitively on a millisecond time-scale (Gerkšič and De Tommasi, 2016). While dual FGM with linear cost on constraint violation is implemented in *QPgen*, a quadratic cost term is regarded necessary for the intended application as well. We achieve our goal by modifying the proximity operator, prove that it gives the expected result, and illustrate it on the AFTI-16 aircraft MPC benchmark example (Giselsson, 2014a).

The paper is organised as follows: the optimization problem of MPC is described in Sect. 2. Lagrange duality is introduced and dual GM is defined in Sect. 3. A solution is proposed in Sect. 4 and proven in Sect. 5. Finally, the modification is applied to dual FGM in Sect. 6 and demonstrated on an example in Sect. 7.

2 MPC Problem description

Consider a discrete time linear system with the dynamics described as

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)\,, \tag{1}$$

where t is the time index, $\mathbf{x}$ is the system state, $\mathbf{u}$ is the system input, and the matrices $\mathbf{A}$ and $\mathbf{B}$ model the dynamics. A quadratic cost function J is introduced (Giselsson, 2014a) as

$$J = \frac{1}{2}\left(\mathbf{x}_N - \mathbf{x}_{\text{ref}}\right)^{\mathrm{T}} \mathbf{Q}\left(\mathbf{x}_N - \mathbf{x}_{\text{ref}}\right) + \frac{1}{2}\sum_{k=0}^{N-1}\left(\mathbf{x}_k - \mathbf{x}_{\text{ref}}\right)^{\mathrm{T}} \mathbf{Q}\left(\mathbf{x}_k - \mathbf{x}_{\text{ref}}\right) + \left(\mathbf{u}_k - \mathbf{u}_{\text{ref}}\right)^{\mathrm{T}} \mathbf{R}\left(\mathbf{u}_k - \mathbf{u}_{\text{ref}}\right). \tag{2}$$

The expressions $\mathbf{x}_{\text{ref}}$ and $\mathbf{u}_{\text{ref}}$ are the system state and system input setpoints, $\mathbf{x}_k$ and $\mathbf{u}_k$ are the state and input values k time steps towards the future from the current time. The signals are constrained to polyhedra $\mathbf{x} \in \mathscr{X}, \mathbf{u} \in \mathscr{U}$ where the polyhedra are defined using the constraint matrices $\mathbf{C}_x{}'$, $\mathbf{C}_u{}'$ and the constraint vectors $\mathbf{b}_x{}'$, $\mathbf{b}_u{}'$ as $\mathscr{X} = \left\{\mathbf{x} \in \mathbb{R}^j | \mathbf{C}_x{}'\mathbf{x} \preceq \mathbf{b}_x{}'\right\}$, $\mathscr{U} = \left\{\mathbf{u} \in \mathbb{R}^l | \mathbf{C}_u{}'\mathbf{u} \preceq \mathbf{b}_u{}'\right\}$. The question of finding the minimizer of the cost for a given value of $\mathbf{x}(0)$ is (Boyd and Vandenberghe, 2004)

$$\begin{aligned}
\underset{(\mathbf{x}_0,\ldots,\mathbf{x}_N,\mathbf{u}_0,\ldots,\mathbf{u}_{N-1})}{\text{minimize}} \quad & J\left(\mathbf{x}_k, \mathbf{u}_k\right) \\
\text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \\
& \mathbf{x}_k \in \mathscr{X}, \mathbf{u}_k \in \mathscr{U}, \\
& \mathbf{x}_0 = \mathbf{x}(0)\,.
\end{aligned} \tag{3}$$

By substituting $\mathbf{x}_1,\dots,\mathbf{x}_N$ via (1) (Ullmann and Richter, 2012), (3) can be transformed into the *condensed* QP form

$$\begin{aligned} \underset{\mathbf{z}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{z}^\mathrm{T}\mathbf{H}\mathbf{z}+\mathbf{c}^\mathrm{T}\mathbf{z} && \text{(4a)}\\ \text{subject to} \quad & \mathbf{C}\mathbf{z}\preceq\mathbf{b} && \text{(4b)} \end{aligned}$$

with the Hessian $\mathbf{H}\in\mathbb{R}^{n\times n}$ positive definite ($n = l\times N$), the gradient vector $\mathbf{c}\in\mathbb{R}^n$, the constraint matrix $\mathbf{C}\in\mathbb{R}^{m\times n}$, the constraint vector $\mathbf{b}\in\mathbb{R}^m$, the optimization vector $\mathbf{z}\in\mathbb{R}^n$. Using the condensed form of the QP, only the system inputs $\mathbf{u}_0,\dots,\mathbf{u}_{N-1}$ are assembled into the optimization variable $\mathbf{z}$.

A solution to an optimization problem of the type (4) arising from MPC with hard state constraints may be infeasible. In order to avoid the problem of infeasibility, the augmented form of the QP with softened state constraints is introduced

$$\begin{aligned} \underset{\mathbf{z},\mathbf{s}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{z}^\mathrm{T}\mathbf{H}\mathbf{z}+\mathbf{c}^\mathrm{T}\mathbf{z}+\frac{1}{2}\mathbf{s}^\mathrm{T}\mathbf{W}\mathbf{s}+\mathbf{w}^\mathrm{T}\mathbf{s} && \text{(5a)}\\ \text{subject to} \quad & \mathbf{C}_x\mathbf{z}\preceq\mathbf{b}_x+\mathbf{s}, && \text{(5b)}\\ & \mathbf{C}_u\mathbf{z}\preceq\mathbf{b}_u, && \text{(5c)}\\ & \mathbf{s}\succeq\mathbf{0}. && \text{(5d)} \end{aligned}$$

We have split

$$\mathbf{C}=\begin{bmatrix}\mathbf{C}_x\\ \mathbf{C}_u\end{bmatrix},\quad \mathbf{b}=\begin{bmatrix}\mathbf{b}_x\\ \mathbf{b}_u\end{bmatrix}, \tag{6}$$

so that (5b) describes the system state constraints that are softened and (5c) describes the system input (actuator) constraints that are hard. The dimensions are $\mathbf{C}_x\in\mathbb{R}^{p\times n}$, $\mathbf{C}_u\in\mathbb{R}^{(m-p)\times n}$, $\mathbf{b}_x\in\mathbb{R}^p$, $\mathbf{b}_u\in\mathbb{R}^{m-p}$. The vector $\mathbf{s}\in\mathbb{R}^p$ is the vector of slack variables, the linear cost on the slack vector $\mathbf{w}\in\mathbb{R}^p$ only has positive components, the matrix $\mathbf{W}\in\mathbb{R}^{p\times p}$ is diagonal positive semi-definite. The QP (5) always has a solution. If the QP (4) is feasible, the QP (5) has the same optimum in $\mathbf{z}$ if the components of $\mathbf{w}$ are big enough (Hovd and Stoican, 2014; Kerrigan and Maciejowski, 2000). If (4) is infeasible, the optimum of (5) violates the constraints of (4) in a predictable way that is determined by the choice of $\mathbf{w}$ and $\mathbf{W}$.

It is possible to rewrite the QP (5) to the general form of QP (4) by augmentation of $\mathbf{z}$, $\mathbf{H}$, $\mathbf{c}$, and $\mathbf{C}$ of (4) with $\mathbf{s}$, $\mathbf{w}$, $\mathbf{W}$. The resulting form is, for example,

$$\begin{aligned} \underset{\left[\begin{smallmatrix}\mathbf{z}\\ \mathbf{s}\end{smallmatrix}\right]}{\text{minimize}} \quad & \frac{1}{2}\begin{bmatrix}\mathbf{z}\\ \mathbf{s}\end{bmatrix}^\mathrm{T}\begin{bmatrix}\mathbf{H} & \mathbf{0}\\ \mathbf{0} & \mathbf{W}\end{bmatrix}\begin{bmatrix}\mathbf{z}\\ \mathbf{s}\end{bmatrix}+\begin{bmatrix}\mathbf{c}\\ \mathbf{w}\end{bmatrix}^\mathrm{T}\begin{bmatrix}\mathbf{z}\\ \mathbf{s}\end{bmatrix}\\ \text{subject to} \quad & \begin{bmatrix}\mathbf{C}_x & -\mathbf{I}\\ \mathbf{C}_u & \mathbf{0}\\ \mathbf{0} & -\mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{z}\\ \mathbf{s}\end{bmatrix}\preceq\begin{bmatrix}\mathbf{b}_x\\ \mathbf{b}_u\\ \mathbf{0}\end{bmatrix}, \end{aligned} \tag{7}$$

where the symbol $\mathbf{I}$ stands for an identity matrix of the appropriate size and each $\mathbf{0}$ is a matrix of zeros of the appropriate size.

3 Lagrange duality and dual methods

For input-constrainted MPC, the inequality (4b) defines a set that is *simple*, meaning that a projection on it can be carried out efficiently. A GM can then be used to solve the QP (4) in primal domain as explained in Nesterov (2003); Patrinos and Bemporad (2014); Richter (2012). In contrast, state constraints in the MPC problem result in a set defined by (4b) that is not simple, necessitating the use of a dual method.

We define the Lagrangian associated with (4) by relaxing the inequality constraints (4b) to obtain the expression

$$L(\mathbf{z},\mu) = \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z} + \mathbf{c}^{\mathrm{T}}\mathbf{z} + \mu^{\mathrm{T}}(\mathbf{C}\mathbf{z} - \mathbf{b}). \tag{8}$$

The vector $\mathbf{z}$ comprises the primal variable while $\mu \in \mathbb{R}^m$ is the dual variable or the *Lagrange multiplier*.

One can define the *Lagrange dual function* as

$$g(\mu) = \inf_{\mathbf{z}} L(\mathbf{z},\mu). \tag{9}$$

The problem

$$\begin{aligned} &\underset{\mu}{\text{maximize}} && g(\mu) && \text{(10a)}\\ &\text{subject to} && \mu \succeq 0 && \text{(10b)} \end{aligned}$$

is called the *dual problem* to the quadratic program (4) and its optimum can be labelled μ^*. Since $g(\mu)$ may not be strongly concave (Bemporad et al, 2002), the maximizer may not be unique and μ^* labels an arbitrary one. An important attribute of the optimal Lagrange multiplier is that the unconstrained optimization

$$\underset{\mathbf{z}}{\text{minimize}} \qquad L(\mathbf{z},\mu^*) \tag{11}$$

has the same optimum in $\mathbf{z}$ as the quadratic program (4) (Boyd and Vandenberghe, 2004).

The dual problem (10) is a QP (Dorn, 1960) as the dual function is $g(\mu) = -\frac{1}{2}\left(\mathbf{C}^{\mathrm{T}}\mu - \mathbf{c}\right)^{\mathrm{T}}\mathbf{H}^{-1}\left(\mathbf{C}^{\mathrm{T}}\mu - \mathbf{c}\right) - \mu^{\mathrm{T}}\mathbf{b}$. The non-negative orthant defined by (10b) is a simple set so the QP (10) can be solved using a GM and (11) can be used to reconstruct the primal solution. Similarly, the dual GM solves a QP of the type (4) through the iterations of (Giselsson and Boyd, 2014)

$$\begin{aligned} \mathbf{y}^k &= -\mathbf{H}^{-1}\left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^k + \mathbf{c}\right)\\ \mathbf{v}^{k+1} &= \mathbf{v}^k + \mathbf{C}\mathbf{y}^k - \operatorname{prox}_h\left(\mathbf{v}^k + \mathbf{C}\mathbf{y}^k\right). \end{aligned} \tag{12}$$

The vector $\mathbf{y}^k \in \mathbb{R}^n$ in (12) has the role of the approximate primal solution and $\mathbf{v}^k \in \mathbb{R}^m$ is the dual variable. The proximity operator of a closed convex function $f : \mathbb{R}^r \to \mathbb{R} \cup \{+\infty\}$ that is not identical to $\{+\infty\}$, is defined as

$$\operatorname{prox}_f(\mathbf{t}) = \underset{\mathbf{r}\in\mathbb{R}^r}{\operatorname{argmin}}\, f(\mathbf{r}) + \frac{1}{2}\|\mathbf{t} - \mathbf{r}\|^2 \tag{13}$$

(Giselsson, 2014a; Richter, 2012). In (12), $h(\mathbf{t})$ symbolises the indicator function

$$h(\mathbf{t}) = \begin{cases} 0 \text{ if } \mathbf{t} \preceq \mathbf{b} \\ \infty \text{ if } \mathbf{t} \npreceq \mathbf{b}. \end{cases} \tag{14}$$

It follows from (13) that $\text{prox}_h(\mathbf{t})$ where $h(\mathbf{t})$ is an indicator function is the projection onto the cone $\mathbf{t} \preceq \mathbf{b}$. Since the cone is a translated orthant and projection can be done by component, the whole iteration cycle is straightforward to perform. The variable $\mathbf{y}^k$ converges to the solution of the QP when $k \to \infty$ if all eigenvalues of $\mathbf{C}\mathbf{H}^{-1}\mathbf{C}^{\mathrm{T}}$ are smaller than or equal to 1 (Giselsson and Boyd, 2014) and this condition can be fulfilled by scaling the cost function.

4 Proposed solution

The main idea of the paper is as follows. What we want is to use the method (12) for the problem (5) in an efficient way. In particular, we avoid rewriting (5) into (7) and applying (12) because of the bigger dimensions and slower convergence that would result. Our goal is deriving a scheme that is similar to and similarly efficient than (12) applied to (4) and results in the solution of (5). We proceed similarly as Giselsson (2014a); Kouzoupis (2014), modifying the proximity operator so it is not a simple projection anymore.

We write the iteration scheme used to solve the QP (5) as

$$\mathbf{y}^k = -\mathbf{H}^{-1}\left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^k + \mathbf{c}\right) \tag{15a}$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \mathbf{C}\mathbf{y}^k - \widetilde{\text{prox}}_{h,W,w}\left(\mathbf{v}^k + \mathbf{C}\mathbf{y}^k\right), \tag{15b}$$

where we define $\widetilde{\text{prox}}_{h,W,w}\left(\mathbf{v}^k + \mathbf{C}\mathbf{y}^k\right)$ by components. The i-th component of $\widetilde{\text{prox}}_{h,W,w}\left(\mathbf{v}^k + \mathbf{C}\mathbf{y}^k\right)$ is

$$\widetilde{\text{prox}}_{h,\mathbf{W},\mathbf{w}}(\mathbf{t})_i := \begin{cases} t_i & \text{if } t_i \le b_i \\ b_i & \text{if } t_i > b_i \text{ and } i \text{ hard} \\ b_i & \text{if } b_i < t_i \le b_i + w_i \text{ and } i \text{ soft} \\ \frac{t_i + W_{ii}b_i - w_i}{W_{ii}+1} & \text{if } t_i > b_i + w_i \text{ and } i \text{ soft} \end{cases} \tag{16}$$

and all the eigenvalues of $\mathbf{C}\mathbf{H}^{-1}\mathbf{C}^{\mathrm{T}}$ are smaller than or equal to 1. We will see that $\mathbf{y}^k$ converges to the optimum of (5) as $k \to \infty$.

Lemma 1 *The algorithm (15) converges.*

Proof Define

$$\widetilde{h}(\mathbf{t}) := \sum_{i=1}^{m} \begin{cases} 0 & \text{if } t_i \le b_i \\ w_i(t_i - b_i) + \frac{W_{ii}}{2}(t_i - b_i)^2 & \text{if } t_i > b_i \text{ and } i \text{ soft} \\ \infty & \text{if } t_i > b_i \text{ and } i \text{ hard.} \end{cases} \tag{17}$$

Calculating $\text{prox}_{\widetilde{h}}(\mathbf{t})$ by the definition of proximity operator (13), it can easily be shown that $\text{prox}_{\widetilde{h}}(\mathbf{t}) = \widetilde{\text{prox}}_{h,\mathbf{W},\mathbf{w}}(\mathbf{t})$. Because $\widetilde{h}(\mathbf{t})$ is a proper closed convex function, the algorithm (15) converges (Giselsson and Boyd, 2015). We name the limits for $\mathbf{y}^k$, $\mathbf{v}^k$ when $k \to \infty$ as $\mathbf{y}^*$, $\mathbf{v}^*$. □

5 Correctness of the proposed solution

A Lagrangian associated with (5) is defined by relaxing the inequality constraints (5b) and (5c) to obtain the expression

$$L(\mathbf{z},\mathbf{s},\mu) = \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z} + \mathbf{c}^{\mathrm{T}}\mathbf{z} + \frac{1}{2}\mathbf{s}^{\mathrm{T}}\mathbf{W}\mathbf{s} + \mathbf{w}^{\mathrm{T}}\mathbf{s} + \mu^{\mathrm{T}}\left(\mathbf{C}\mathbf{z} - \mathbf{b} - \begin{bmatrix}\mathbf{s}\\ \mathbf{0}_{m-p}\end{bmatrix}\right). \tag{18}$$

The remaining set of constraints (5d) is not relaxed since it is straightforward to keep fulfilled and can be treated separately. The vectors $\mathbf{z}$ and $\mathbf{s}$ together comprise the primal variable while $\mu \in \mathbb{R}^m$ is the Lagrange multiplier.

Lemma 2 *If* $\mathbf{y}^*$*,* $\mathbf{v}^*$ *are limits of* $\mathbf{y}^k$*,* $\mathbf{v}^k$ *in the algorithm (15) as* $k \to \infty$ *and* $\mathbf{s}^*$ *is defined to be composed of components* ${s^*}_i = \max\left(0, (\mathbf{C}\mathbf{y}^*)_i - b_i\right)$*, where* i *is a soft constraint, then* $\mathbf{z} = \mathbf{y}^*$*,* $\mathbf{s} = \mathbf{s}^*$ *is the optimum of the QP (5).*

Proof We verify that Karush-Kuhn-Tucker (KKT) conditions for optimality (Boyd and Vandenberghe, 2004; Karush, 2014; Kuhn and Tucker, 1951) for the Lagrangian (18) are fulfilled at $\mathbf{z} = \mathbf{y}^*$, $\mathbf{s} = \mathbf{s}^*$, $\mu = \mathbf{v}^*$.

We start by expressing $\mathbf{v}^*$. In the limit, (15b) becomes

$$\mathbf{v}^* = \mathbf{v}^* + \mathbf{C}\mathbf{y}^* - \widetilde{\operatorname{prox}}_{h,W,w}\left(\mathbf{v}^* + \mathbf{C}\mathbf{y}^*\right).$$

Taking the definition of $\widetilde{\operatorname{prox}}_{h,W,w}\left(\mathbf{v}^* + \mathbf{C}\mathbf{y}^*\right)$ in the account, we find that

$${v^*}_i \begin{cases} = 0 & \text{if } (\mathbf{C}\mathbf{y}^*)_i < b_i \\ \geq 0 & \text{if } (\mathbf{C}\mathbf{y}^*)_i = b_i \text{ and } i \text{ hard} \\ \geq 0 \text{ and } \leq w_i & \text{if } (\mathbf{C}\mathbf{y}^*)_i = b_i \text{ and } i \text{ soft} \\ = W_{ii}\left((\mathbf{C}\mathbf{y}^*)_i - b_i\right) + w_i & \text{if } (\mathbf{C}\mathbf{y}^*)_i > b_i. \end{cases} \tag{19}$$

The first part of the stationarity condition states that the gradient of the Lagrangian (8) in $\mathbf{z}$ is 0 at optimal $\mathbf{z}$ when μ is an optimal Lagrange multiplier. The gradient is

$$\nabla_{\mathbf{z}} L = \mathbf{H}\mathbf{z} + \mathbf{c} + \mathbf{C}^{\mathrm{T}}\mu. \tag{20}$$

From (15a) we see that $\mathbf{y}^* = -\mathbf{H}^{-1}\left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^* + \mathbf{c}\right)$. When we take the expression into account and substitute $\mu = \mathbf{v}^*$, $\mathbf{z} = \mathbf{y}^*$, it directly follows $\nabla_{\mathbf{z}} L = 0$. The condition is fulfilled.

The other part of the stationarity condition states that the gradient of the Lagrangian (8) in $\mathbf{s}$ is 0 for optimal $\mathbf{s}$ when μ is an optimal Lagrange multiplier. The gradient is

$$\nabla_{\mathbf{s}} L = \mathbf{W}\mathbf{s} + \mathbf{w} - \mu_{\mathbf{x}}, \tag{21}$$

where $\mu_{\mathbf{x}}$ is the vector of the components of μ corresponding to soft constraints. As the set of inequality constraints (5d) is not relaxed, the condition in this form only has to be fulfilled for the components $s_i > 0$. These correspond to the last line in the relation (19) that ensures $(\nabla_{\mathbf{s}} L)_i = 0$. For the other components, we want $(\nabla_{\mathbf{s}} L)_i \geq 0$ as this warrants $s_i = 0$ is a minimizer under the condition $s_i \geq 0$ that we decided to treat separately. For these, it thus follows from (21) that it has to be $W_{ii} s_i \geq \mu_i - w_i$. The requirement is met because they all correspond to the first and third lines in (19),

$\mu_i - w_i \leq 0$, and $W_{ii} s_i = 0$. The stationarity condition is thus met in full for $\mathbf{z} = \mathbf{y}^*$, $\mathbf{s} = \mathbf{s}^*$, $\mu = \mathbf{v}^*$.

The primal feasibility condition demands the inequalities (5b) and (5c) be fulfilled. It follows straight from the choice of $\mathbf{s}^*$ that (5b) is fulfilled for $\mathbf{z} = \mathbf{y}^*$, $\mathbf{s} = \mathbf{s}^*$. Inequality (5c) is fulfilled for $\mathbf{z} = \mathbf{y}^*$ because hard constraints always result in $\widetilde{\text{prox}}_{h,W,w}(\mathbf{v}^* + \mathbf{C}\mathbf{y}^*)$ being calculated according to one of the first two lines in (16), leading to $(\mathbf{C}\mathbf{y}^*)_i \leq b_i$.

Dual feasibility is the condition stating that the components of μ are non-negative, $\mu \succeq 0$. It follows that $v^*_i > 0$ for all cases of the relation (19) so the condition is met for $\mu = \mathbf{v}^*$.

The complementary slackness condition demands $\mu^{\mathrm{T}} \cdot (\mathbf{C}\mathbf{z} - \mathbf{b} - \mathbf{s}) = 0$. As both factors are non-negative, the i-th component of either one has to be 0 for all i. It is true for $\mathbf{z} = \mathbf{y}^*$, $\mathbf{s} = \mathbf{s}^*$, $\mu = \mathbf{v}^*$.

- For the 1st case in (19), it is $v^*_i = 0$.
- For the 2nd and 3rd case in (19), we know that $(\mathbf{C}\mathbf{y}^*)_i = b_i$ and $s^*_i = 0$ so the second factor is 0.
- In the 4th case, it is $s^*_i = (\mathbf{C}\mathbf{y}^*)_i - b_i$, so the second factor is 0.

All the KKT conditions are fulfilled for $\mathbf{z} = \mathbf{y}^*$, $\mathbf{s} = \mathbf{s}^*$, $\mu = \mathbf{v}^*$, so the limit of the iteration scheme is an optimum. □

Theorem 1 *The algorithm (15) converges to the optimum of the QP (5).*

Proof Lemma 2 tells us that the limit of the algorithm is an optimum, and lemma 1 tells us that the algorithm converges. □

6 Dual fast gradient method

The dual FGM solves the QP (4) through the iteration scheme (Giselsson and Boyd, 2014)

$$\begin{aligned} \nu^k &= \mathbf{v}^k + \beta^k \left(\mathbf{v}^k - \mathbf{v}^{k-1} \right) \\ \mathbf{y}^k &= -\mathbf{H}^{-1} \left(\mathbf{C}^{\mathrm{T}} \nu^k + \mathbf{c} \right) \\ \mathbf{v}^{k+1} &= \nu^k + \mathbf{C}\mathbf{y}^k - \text{prox}_h \left(\nu^k + \mathbf{C}\mathbf{y}^k \right), \end{aligned} \tag{22}$$

where the sequence of scalar weights β^k is chosen in a way that accelerates convergence. We will show that the modified scheme

$$\nu^k = \mathbf{v}^k + \beta^k \left(\mathbf{v}^k - \mathbf{v}^{k-1} \right) \tag{23a}$$

$$\mathbf{y}^k = -\mathbf{H}^{-1} \left(\mathbf{C}^{\mathrm{T}} \nu^k + \mathbf{c} \right) \tag{23b}$$

$$\mathbf{v}^{k+1} = \nu^k + \mathbf{C}\mathbf{y}^k - \widetilde{\text{prox}}_{h,W,w} \left(\nu^k + \mathbf{C}\mathbf{y}^k \right), \tag{23c}$$

with $\widetilde{\text{prox}}_{h,W,w} \left(\mathbf{v}^k + \mathbf{C}\mathbf{y}^k \right)$ as defined in (16) converges to a solution for the QP (5).

We name the limits for $\mathbf{y}^k$, $\mathbf{v}^k$, ν^k of (23) when $k \to \infty$ as $\mathbf{y}_{\mathrm{FGM}}{}^*$, $\mathbf{v}_{\mathrm{FGM}}{}^*$, $\nu_{\mathrm{FGM}}{}^*$. Noting that $\mathbf{v}_{\mathrm{FGM}}{}^* = \nu_{\mathrm{FGM}}{}^*$, equations (23b), (23c) prescribe the same properties to $\mathbf{y}_{\mathrm{FGM}}{}^*$, $\mathbf{v}_{\mathrm{FGM}}{}^*$ as equations (15a), (15b) do to $\mathbf{y}^*$ $\mathbf{v}^*$. Since $\mathbf{y}^*$ $\mathbf{v}^*$ are optimal, so are $\mathbf{y}_{\mathrm{FGM}}{}^*$, $\mathbf{v}_{\mathrm{FGM}}{}^*$.

7 Example

We demonstrate the modified dual FGM on a QP resulting from the AFTI-16 benchmark model (Giselsson, 2014a). We examine the simulated control response of the AFTI-16 system with the proposed method (23) and pay special attention to one sample operating point that has some non-zero components of $\mathbf{s}$ at the optimum.

The model is described by the matrices

$$\mathbf{A} = \begin{bmatrix} 0.9993 & -3.0083 & -0.1131 & -1.6081 \\ -0.0000 & 0.9862 & 0.0478 & 0.0000 \\ 0.0000 & 2.0833 & 1.0089 & -0.0000 \\ 0.0000 & 0.0526 & 0.0498 & 1.0000 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -0.0804 & -0.6347 \\ -0.0291 & -0.0143 \\ -0.8679 & -0.0917 \\ -0.0216 & -0.0022 \end{bmatrix}$$

when presented in the form (1). Possible system states and inputs are constrained,[1]

$$\begin{aligned} \mathscr{X}\mathbf{x}_k &\preceq \xi + \mathbf{s}^+{}_k \\ -\mathscr{X}\mathbf{x}_k &\preceq \xi + \mathbf{s}^-{}_k \\ \mathscr{U}\mathbf{u}_k &\preceq \zeta \\ -\mathscr{U}\mathbf{u}_k &\preceq \zeta, \end{aligned}$$

where

$$\mathscr{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathscr{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \xi = \begin{bmatrix} 0.5 \\ 100 \end{bmatrix}, \quad \zeta = \begin{bmatrix} 25 \\ 25 \end{bmatrix}.$$

We label

$$\mathbf{s}_k = \begin{bmatrix} \mathbf{s}^+{}_k \\ \mathbf{s}^-{}_k \end{bmatrix}.$$

The cost function is defined as

$$J = \frac{1}{2}\left(\mathbf{x}_N - \mathbf{x}_{\mathrm{ref}}\right)^{\mathrm{T}} \mathbf{Q} \left(\mathbf{x}_N - \mathbf{x}_{\mathrm{ref}}\right) + \frac{1}{2}\sum_{k=0}^{N-1} \Big(\left(\mathbf{x}_k - \mathbf{x}_{\mathrm{ref}}\right)^{\mathrm{T}} \mathbf{Q} \left(\mathbf{x}_k - \mathbf{x}_{\mathrm{ref}}\right) \\ + \left(\mathbf{u}_k - \mathbf{u}_{\mathrm{ref}}\right)^{\mathrm{T}} \mathbf{R} \left(\mathbf{u}_k - \mathbf{u}_{\mathrm{ref}}\right) + \mathbf{s}_k^{\mathrm{T}} \mathbf{W}_{QP} \mathbf{s}_k + 2\mathbf{w}_{QP}{}^{\mathrm{T}} \mathbf{s}_k \Big)$$

[1] The same system state and input components are bound from above and from below, as is often the case. Some solvers, including *QPgen* and its version augmented with the quadratic cost on constraint violation, assume upper and lower bounds on the same signals and substitute the constraints of the QP (5b) with the form $\mathbf{b}_l - \mathbf{s} \preceq \widetilde{\mathbf{C}}\mathbf{x} \preceq \mathbf{b}_u + \mathbf{s}$ and similar for (5c). The QP modified in this way is mathematically equivalent to (5) and more efficient in resource usage.

where $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{W}_{QP}$ are symmetric positive semidefinite cost matrices

$$\begin{aligned}\mathbf{Q} &= \mathrm{diag}(10^{-4}, 10^{2}, 10^{-3}, 10^{2}),\\ \mathbf{R} &= \mathrm{diag}(10^{-2}, 10^{-2}),\\ \mathbf{W}_{QP} &= \mathrm{diag}(10^{3}, 10^{3}, 10^{3}, 10^{3}).\end{aligned} \tag{24}$$

Constant vectors $\mathbf{x}_{\mathrm{ref}}$, $\mathbf{u}_{\mathrm{ref}}$ are setpoints and the linear cost on the slack variables is

$$\mathbf{w}_{QP} = \begin{bmatrix}1300\\1300\\1300\\1300\end{bmatrix}.$$

The prediction horizon is chosen to be $N = 10$.

We derive a condensed QP of the form (5) from the MPC problem the same way as it is done in (Ullmann and Richter, 2012). First we choose the optimization vectors

$$\mathbf{z} = \begin{bmatrix}\mathbf{u}_0\\ \vdots\\ \mathbf{u}_{N-1}\end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix}\mathbf{s}_0\\ \vdots\\ \mathbf{s}_{N-1}\end{bmatrix}.$$

We define

$$\chi = \begin{bmatrix}\mathbf{x}_1\\ \vdots\\ \mathbf{x}_N\end{bmatrix}$$

and express its dependence on $\mathbf{x}_0$ and $\mathbf{z}$ as

$$\chi = \mathscr{A}\mathbf{x}_0 + \mathscr{B}\mathbf{z}$$

From the dynamics (1), we get the expressions for $\mathscr{A}$ and $\mathscr{B}$ that are

$$\mathscr{A} = \begin{bmatrix}\mathbf{A}\\ \mathbf{A}^2\\ \vdots\\ \mathbf{A}^N\end{bmatrix}, \quad \mathscr{B} = \begin{bmatrix}\mathbf{B} & & \\ \mathbf{AB} & \mathbf{B} & \\ \vdots & \ddots & \ddots \\ \mathbf{A}^{N-1}\mathbf{B} & \dots\ \mathbf{AB} & \mathbf{B}\end{bmatrix}.$$

We define

$$\mathscr{Q} = \begin{bmatrix}\mathbf{Q} & & \\ & \ddots & \\ & & \mathbf{Q}\end{bmatrix}, \quad \mathscr{R} = \begin{bmatrix}\mathbf{R} & & \\ & \ddots & \\ & & \mathbf{R}\end{bmatrix}, \quad \mathscr{C} = \begin{bmatrix}\mathscr{X} & & \\ & \ddots & \\ & & \mathscr{X}\\ -\mathscr{X} & & \\ & \ddots & \\ & & -\mathscr{X}\end{bmatrix}$$

and express the remaining elements of the QP (5)

$$\mathbf{H} = \mathscr{B}^{\mathrm{T}}\mathscr{Q}\mathscr{B} + \mathscr{R}$$

$$\mathbf{c} = \mathscr{B}^{\mathrm{T}}\mathscr{Q}\left(\mathscr{A}\mathbf{x}_0 + \begin{bmatrix}\mathbf{x}_{\mathrm{ref}}\\ \vdots \\ \mathbf{x}_{\mathrm{ref}}\end{bmatrix}\right) + \mathscr{R}\begin{bmatrix}\mathbf{u}_{\mathrm{ref}}\\ \vdots \\ \mathbf{u}_{\mathrm{ref}}\end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix}\mathbf{W}_{QP} & & \\ & \ddots & \\ & & \mathbf{W}_{QP}\end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix}\mathbf{w}_{QP}\\ \vdots \\ \mathbf{w}_{QP}\end{bmatrix}$$

$$\mathbf{C}_{\mathbf{x}} = \mathscr{C}\mathscr{B}$$

$$\mathbf{b}_{\mathbf{x}} = \begin{bmatrix}\begin{bmatrix}\xi\\ \vdots \\ \xi\end{bmatrix} - \mathscr{C}\mathscr{A}\mathbf{x}_0 \\ \begin{bmatrix}\xi\\ \vdots \\ \xi\end{bmatrix} + \mathscr{C}\mathscr{A}\mathbf{x}_0\end{bmatrix}$$

$$\mathbf{C}_{\mathbf{u}} = \begin{bmatrix}\mathscr{U} & & \\ & \ddots & \\ & & \mathscr{U} \\ -\mathscr{U} & & \\ & \ddots & \\ & & -\mathscr{U}\end{bmatrix}$$

$$\mathbf{b}_{\mathbf{u}} = \begin{bmatrix}\zeta\\ \vdots \\ \zeta\end{bmatrix}.$$

In Fig. 1 we see that the simulated control response of the AFTI-16 system with the proposed method (23) in 10^5 iterations closely resembles the response obtained with the reference approach with slack variables (7) using the *MATLAB Optimization Toolbox* function *quadprog* solver with default parameters. 100 samples are simulated starting from $\mathbf{x}_0 = {\mathbf{x}_0}^{\mathrm{c}}$; the setpoints in the first 50 samples are $\mathbf{x}_{\mathrm{ref}} = {\mathbf{x}_{\mathrm{ref}}}^{\mathrm{c1}}$, $\mathbf{u}_{\mathrm{ref}} = {\mathbf{u}_{\mathrm{ref}}}^{\mathrm{c}}$, and in the following 50 samples $\mathbf{x}_{\mathrm{ref}} = {\mathbf{x}_{\mathrm{ref}}}^{\mathrm{c2}}$, $\mathbf{u}_{\mathrm{ref}} = {\mathbf{u}_{\mathrm{ref}}}^{\mathrm{c}}$, respectively. The numerical values of the parameters are

$${\mathbf{x}_0}^{\mathrm{c}} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \quad {\mathbf{x}_{\mathrm{ref}}}^{\mathrm{c1}} = \begin{bmatrix}0\\0\\0\\10\end{bmatrix}, \quad {\mathbf{x}_{\mathrm{ref}}}^{\mathrm{c2}} = \begin{bmatrix}0\\0\\0\\0\end{bmatrix}, \quad {\mathbf{u}_{\mathrm{ref}}}^{\mathrm{c}} = \begin{bmatrix}0\\0\end{bmatrix}.$$

In this and all the other dual FGM calculations, restarting is used. As soon as the scalar product $\left(v^k - \mathbf{v}^{k+1}\right) \cdot \left(\mathbf{v}^{k+1} - \mathbf{v}^k\right)$ is positive, the acceleration step (23a) would oppose the gradient. This is prevented by setting $\mathbf{v}^{k+1} = \mathbf{v}^k$ instead of following (23c) in a single iteration cycle.

We inspect a sample point along the simulated state trajectory at

$$\mathbf{x}_{\text{ref}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 10 \end{bmatrix}, \quad \mathbf{x}_0 = \begin{bmatrix} -13.8575 \\ 0.37 \\ 19.405 \\ 0.485 \end{bmatrix}, \tag{25}$$

where a soft constraint is violated. The solution of the QP for this sample point, obtained with either the proposed algorithm (23) or the reference approach, is

$$\mathbf{z}^* = \begin{bmatrix} 11.2934 \\ 25.0000 \\ 3.96299 \\ 25.0000 \\ -5.51605 \\ 25.0000 \\ -0.25038 \\ 25.0000 \\ -1.83887 \\ 25.0000 \\ -1.17691 \\ 25.0000 \\ -1.45277 \\ 25.0000 \\ -1.33781 \\ 25.0000 \\ -1.38572 \\ 25.0000 \\ -1.36575 \\ 25.0000 \end{bmatrix}.$$

The solution obtained after 10^4 iterations of the algorithm (23) has the value of the quadratic norm equal to 80.2259. The quadratic norm of the difference between it and the reference approach solution is equal to 1.52484×10^{-9}. This shows that the algorithm with the modified proximity operator is giving correct results.

The constraint violation is verified to be strong enough that the quadratic cost on the constraint violation has an influence. The quadratic norm of the slack vector in the reference solution is 0.1081. The quadratic norm of the difference between the solution obtained with the scheme (23) and the one obtained with the same method but with $\mathbf{W} = \mathbf{0}$ in (5a) is 25.0892. The original hard-constrained QP is feasible at this system state, and the norm of the difference between our solution and hard-constrained QP solution is 10.529.

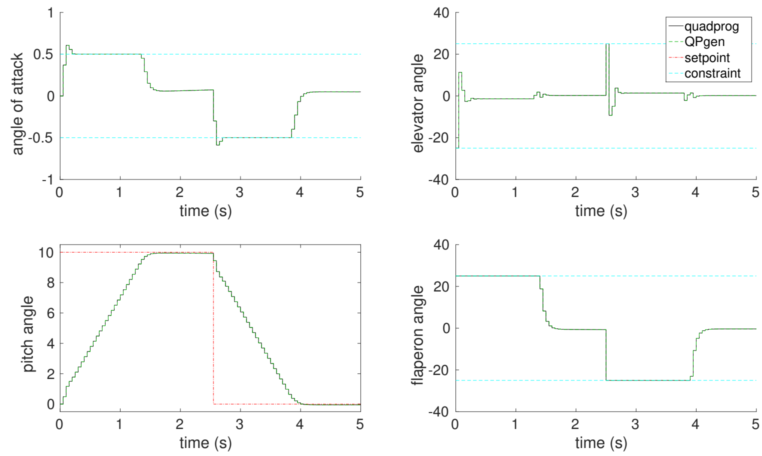

Fig. 1 Comparison of the control response in AFTI-16 MPC simulation using either *MATLAB Optimization Toolbox* function *quadprog* (solid black line) or *QPgen* (dashed green line) to solve the QPs. Two components of the system state and two components of the system input are shown; it can be seen that *quadprog* and *QPgen* results overlap. The setpoints for the components are 0 except where shown in the graphs. It can be seen that the signal x_2 violates the soft constraint in certain samples. In samples 3, 4, 5 (between 0.1 and 0.25 s from the beginning of the simulation), it violates the upper constraint, while in samples 53, 54 (between 2.6 and 2.7 s from the beginning), it violates the lower constraint

The performance of the algorithm (23) is compared to solving the augmented problem with slack variables constructed as in (7) using algorithm (22) and the same restarting scheme. The solution accuracy after various numbers of iterations is compared with the reference *quadprog* solution and the results in terms of the quadratic norm of the relative error are given in Fig. 2. The quadratic norm of the relative error is obtained by dividing each component of the error vector by the full span between the maximum and the minimum value, which in our case is 50 for all of the components, and calculating the quadratic norm of the vector. We see that the results of the proposed algorithm (23) are consistently better than the results of the standard FGM algorithm (22). In addition, the system matrices are of different sizes in the two algorithms, resulting in different computational complexities of a single iteration. For our benchmark problem, the dimensions of the matrix $\mathbf{C}$ are 40 by 20 in the algorithm (23) and 80 by 60 (though sparsely populated) after augmentation by procedure (7) for solving with the algorithm (22).

7.1 Practical implications

On a computer running MATLAB R2015a, OS Ubuntu 16.04 LTS with Linux kernel 4.4.0-83-generic, 15.1 GiB of RAM and Intel Core i7-3770K CPU @ 3.50GHz, the proposed algorithm (23) with 10^4 iterations takes 24.9 ms on average in *QPgen*-generated code for the sample point (25). For comparison, the standard FGM algorithm (22) with slack variables requires 35.1 ms under the same conditions.

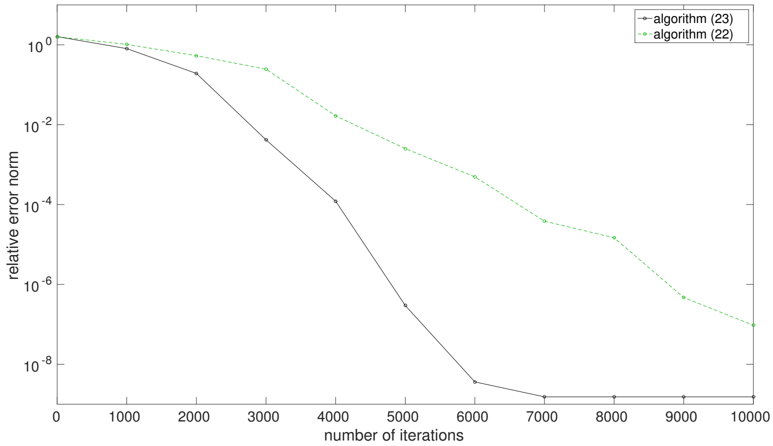

Fig. 2 Relative error norm of the difference between the solution after a certain number of iterations and the reference *quadprog* solution in the sample point described in (25)

It is known that for the poorly conditioned AFTI-16 benchmark problem, the convergence can be improved dramatically by preconditioning (Giselsson, 2014a).[2] Performance of the proposed algorithm (23) with the default *QPgen* preconditioning method is presented in Fig. 3 together with algorithm (22) with preconditioning. We investigate the convergence of the preconditioned algorithms in all the 100 sample points of the simulation, one of which is the sample point (25). We find that 95 iterations of the algorithm (23) with preconditioning are needed to get the relative error norm less than 10^{-4} in all points of the simulation. The chosen relative error is small enough for reasonable control response; when using this QP solver, the system output relative error norm, compared to the full-precision simulation using *quadprog*, stays within 1.3×10^{-6}. The performance of the algorithm (22) with preconditioning is worse; the result after 95 iterations has relative error norm of up to 5.6×10^{-4}.

We also measure the time required by the preconditioned algorithms on the same computer. Solving the QP and obtaining the control input using 95 iterations of (23) with preconditioning requires 0.50 ms in the slowest one of the 100 samples. For the same number of iterations of (22) with preconditioning, 0.55 ms are needed. For the purpose, the algorithms are superior to the interior point algorithm of *quadprog*: when *quadprog* tolerance is increased enough that the relative error norm of its solution increases to 1.01×10^{-4} in the sample point (25), it requires 6.0 ms for the computation. On another similar computer, *IBM ILOG CPLEX* QP solver required around 60% of the time needed by *quadprog* on average, which is still much slower than the algorithm (23) with preconditioning.

[2] The comparison above is made without preconditioning because the results of the two approaches are no longer directly comparable when preconditioning is used.

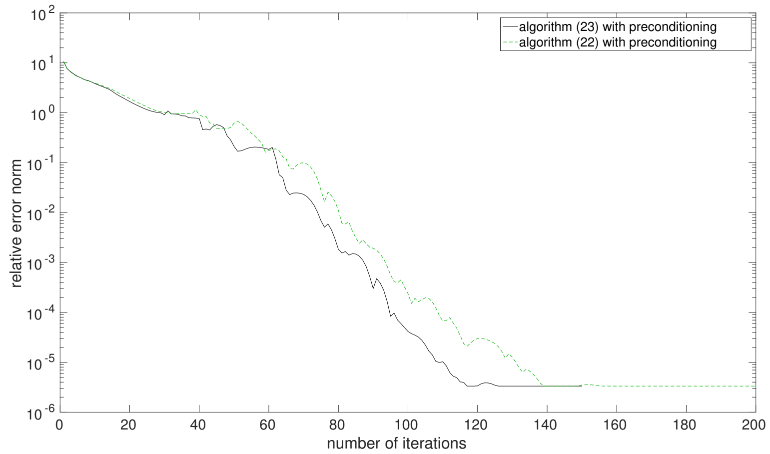

Fig. 3 Maximum relative error norm of the difference between the solution after a certain number of iterations and the reference *quadprog* solution over all 100 sample points of the MPC simulation

8 Conclusion

We derive a method for efficient handling of linear and quadratic cost on inequality constraint violation in a QP. It can be applied to the dual GM and dual FGM for the condensed form of the MPC-derived quadratic program as used in (Giselsson, 2014b). The convergence of the algorithm is proved and Karush-Kuhn-Tucker conditions are used to prove the optimality of the solution.

The method for efficient handling of soft constraints results both in smaller system matrices compared to QP augmented with soft constraints and faster convergence of the algorithm. Fewer iterations are needed to reach the required accuracy and each iteration requires fewer arithmetic operations, both leading to faster computation. This is an important advantage in construction of a MPC controller for a multivariable system with fast dynamics. The method can be combined with techniques for complexity reduction of the MPC problem and for preconditioning of the QP for faster convergence. It enables use of proven MPC features with fast processes.

References

Afonso RJM, Galvo RKH (2012) Infeasibility handling in constrained MPC. In: Frontiers of Model Predictive Control, InTech, pp 47–64, DOI 10.5772/38437

Bemporad A, Morari M, Dua V, Pistikopoulos EN (2002) The explicit linear quadratic regulator for constrained systems. Automatica 38(1):3 – 20, DOI https://doi.org/10.1016/S0005-1098(01)00174-1, URL `http://www.sciencedirect.com/science/article/pii/S0005109801001741`

Borrelli F, Bemporad A, Morari M (2015) Predictive Control for linear and hybrid systems. Cambridge

Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge University Press, New York, NY, USA, URL https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf

Domahidi A, Zgraggen AU, Zeilinger MN, Morari M, Jones C (2012) Efficient interior point methods for multistage problems arising in receding horizon control. In: Proceedings of the 51st IEEE Conference on Decision and Control, pp 668–674, DOI 10.1109/CDC.2012.6426855

Dorn WS (1960) Duality in quadratic programming. Quart Appl Math 18:155–162, DOI 10.1090/qam/112751

Ferreau HJ, Bock HG, Diehl M (2008) An online active set strategy to overcome the limitations of explicit MPC. Int J Robust Nonlin 18(8):816–830, DOI 10.1002/rnc.1251

Ferreau HJ, Kirches C, Potschka A, Bock HG, Diehl M (2014) qpOASES: a parametric active-set algorithm for quadratic programming. Math Program Comput 6(4):327–363, DOI 10.1007/s12532-014-0071-1

Gerkšič S, De Tommasi G (2016) ITER plasma current and shape control using MPC. In: 2016 IEEE Conference on Control Applications (CCA), pp 599–604, DOI 10.1109/CCA.2016.7587895

Gerkšič S, Pregelj B, Perne M (2018) FPGA acceleration of model predictive control for ITER plasma current and shape control. In: 21st IEEE Real Time Conference

Giselsson P (2014a) Improved fast dual gradient methods for embedded model predictive control. IFAC Proceedings Volumes 47(3):2303 – 2309, 19th IFAC World Congress

Giselsson P (2014b) QPgen. URL http://www.control.lth.se/QPgen/index.html, v0.0.2

Giselsson P, Boyd S (2014) Preconditioning in fast dual gradient methods. In: 53rd IEEE Conference on Decision and Control, CDC 2014, Los Angeles, CA, USA, December 15-17, 2014, pp 5040–5045, DOI 10.1109/CDC.2014.7040176

Giselsson P, Boyd S (2015) Metric selection in fast dual forward-backward splitting. Automatica 62:1–10, DOI 10.1016/j.automatica.2015.09.010

Hartley EN, Jerez JL, Suardi A, Maciejowski JM, Kerrigan EC, Constantinides GA (2014) Predictive control using an FPGA with application to aircraft control. IEEE Trans Control Syst Technol 22(3):1006–1017, DOI 10.1109/TCST.2013.2271791

Hovd M, Stoican F (2014) On the design of exact penalty functions for MPC using mixed integer programming. Comput Chem Eng 70:104–113, DOI http://dx.doi.org/10.1016/j.compchemeng.2013.07.001, URL http://www.sciencedirect.com/science/article/pii/S0098135413002214, Manfred Morari Special Issue

Jerez JL, Goulart PJ, Richter S, Constantinides GA, Kerrigan EC, Morari M (2014) Embedded online optimization for model predictive control at megahertz rates. IEEE Trans Automat Contr 59(12):3238–3251, DOI 10.1109/TAC.2014.2351991

Karush W (2014) Minima of functions of several variables with inequalities as side conditions. In: Giorgi G, Kjeldsen T (eds) Traces and Emergence of Nonlinear Programming, Birkhäuser, Basel, DOI 10.1007/978-3-0348-0439-4_10, reproduction of Master Thesis, Department of Mathematics, University of Chicago, Chicago 1939

Kerrigan EC, Maciejowski JM (2000) Soft constraints and exact penalty functions in model predictive control. In: Proc. UKACC International Conference (Control), URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.228.1327&rep=rep1&type=pdf

Kouzoupis D (2014) Complexity of First-Order Methods for Fast Embedded Model Predictive Control (Master Thesis). Eidgenössische Technische Hochschule, Zürich, DOI 10.3929/ethz-a-010255501

Kuhn HW, Tucker AW (1951) Nonlinear programming. In: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, pp 481–492, URL https://projecteuclid.org/euclid.bsmsp/1200500249

Mattingley J, Boyd S (2012) CVXGEN: a code generator for embedded convex optimization. Optim Eng 13(1):1–27, DOI 10.1007/s11081-011-9176-9

Mattingley J, Wang Y, Boyd S (2011) Receding horizon control, automatic generation of high-speed solvers. IEEE Control Syst Mag 31:52–65, DOI 10.1109/MCS.2011.940571

Nesterov Y (2003) Introductory Lectures on Convex Optimization: A Basic Course. Applied Optimization, Springer US

de Oliveira NMC, Biegler LT (1994) Constraint handling and stability properties of model-predictive control. AIChE J 40(7):1138–1155, URL http://repository.cmu.edu/cgi/viewcontent.cgi?article=1196&context=cheme

Patrinos P, Bemporad A (2014) An accelerated dual gradient-projection algorithm for embedded linear model predictive control. IEEE Trans Automat Contr 59(1):18–33, DOI 10.1109/TAC.2013.2275667

Patrinos P, Guiggiani A, Bemporad A (2015) A dual gradient-projection algorithm for model predictive control in fixed-point arithmetic. Automatica 55(C):226–235, DOI 10.1016/j.automatica.2015.03.002

Qin S, Badgwell TA (2003) A survey of industrial model predictive control technology. Control Eng Pract 11(7):733–764, DOI 10.1016/S0967-0661(02)00186-7, URL http://www.sciencedirect.com/science/article/pii/S0967066102001867

Richter S (2012) Computational Complexity Certification of Gradient Methods for Real-Time Model Predictive Control (Dissertation). Eidgenössische Technische Hochschule, Zürich, DOI 10.3929/ethz-a-007587480

Ullmann F, Richter S (2012) FiOrdOs – MPC Example. Automatic Control Laboratory, ETH Zurich, Zurich, URL http://fiordos.ethz.ch/dokuwiki/doku.php?id=mpcexample

Wang Y, Boyd S (2010) Fast model predictive control using online optimization. IEEE Trans Control Syst Technol 18:267–278, DOI 10.1109/TCST.2009.2017934

Zafiriou E, Chiou HW (1993) Output constraint softening for SISO model predictive control. In: Proceedings of the American Control Conference, San Francisco, pp 372–376, URL http://hdl.handle.net/1903/5360

Zheng A, Morari M (1995) Stability of model predictive control with mixed constraints. IEEE Trans Automat Contr 40(10):1818–1823, DOI 10.1109/9.467664