Local Decay of Residuals in Dual Gradient Method with Soft State Constraints

Matija Perne[1], Samo Gerkšič[1], and Boštjan Pregelj[1]

Jožef Stefan Institute, Jamova cesta 39, Ljubljana, Slovenia,
{matija.perne, samo.gerksic, bostjan.pregelj}@ijs.si

Abstract. Quadratic programs resulting from a model predictive control problem in real-time control context are solved using a dual gradient method. The projection operator of the method is modified so as to implement soft state constraints with linear and quadratic cost on constraint violation without directly calculating values of slack variables. Evolution of iterates and residuals throughout iterations of the method is studied. We notice that in most iterations, the set of the constraints that are active and the ones that are violated does not change. Observing the residuals through multiple iterations in which the active and violated sets do not change leads to interesting results. When the dual residual is transformed into a certain base, its components are decaying independently of each other and at exactly predictable rates. The transformation only depends on the system matrices and on the active and violated sets. Since the matrices are independent of the system state, so is the transformation, and the decay rate of the components stays constant through multiple iterations. The predictions are confirmed by numerical simulations of MPC control, which is shown for the AFTI-16 benchmark example.

Keywords: Model Predictive Control, Gradient Method, Optimization

1 Introduction

Model predictive control (MPC) is traditionally limited to processes with relatively slow dynamics because of the computational complexity of online optimization [21]. In the last decade, a considerable advance has been made in the field of fast online optimization [3–5, 10, 15, 16, 25].

The advantages of MPC appear promising for the implementation of advanced plasma current and shape control in a tokamak fusion reactor [6]. For MPC purposes, a relatively low accuracy of the solution is sufficient, diminishing the importance of convergence rate for selection of the method. We are focusing on fast online implementations of first-order methods adapted for use with MPC [7, 9, 14, 18]. They converge sublinearly, in strongly convex case linearly [23]. In contrast, active set methods converge in a finite number of iterations [23] and interior point methods have a superlinear convergence rate [2]. However, first order methods have certain advantages. It is possible to implement them in

restricted hardware as each iteration is simple. Although they tend to require more iterations than second-order methods to achieve the desired accuracy, each iteration typically requires much less computation, possibly resulting in lower total computational time.

When moderate accuracy is required, observing the change of residual from one iteration to the next one is of interest [19]. The concept of decay of residuals is closely related to both complexity certification and to convergence rate. As defined in [23], the computational complexity certificate for an iterative solution method comes as a fixed, *a priori* computable upper iteration bound from a non-asymptotic convergence rate analysis of the method. It is a proof, valid for every possible system state included in a given set, that the calculated control input will be at most for a certain amount ϵ away from (or costlier than) the optimal control input after a certain number of iterations N of the solver. In contrast, convergence rate is a theoretical concept describing the relative decrease of the error contributed by one additional iteration in the limit when the number of iterations $N \to \infty$. Convergence rate is the last rate of the local decay of residuals, the one that is valid from some number of iterations on indefinitely. As the complexity certificate is the quality of the solution after a given number of iterations, generalized to a set of cases, all the local decay rates of the residuals throughout the iterations are of interest when obtaining it.

The MPC problem may not have a feasible solution, meaning that its constraints may be in disagreement. However, in practice we want the controller to produce a sensible output also when the constraints cannot be satisfied. In many cases this may be achieved by relaxing the state constraint, allowing it to be violated, while adding a term corresponding to the violation to the cost function, with a high violation penalty [1, 11, 13, 17, 26, 26, 27].

We examine the rates of decay of residuals observed with a dual gradient method in the presence of soft state constraints, where the violation is penalised with linear and quadratic cost. We derive an expression for the decay rates in intervals within which the sets of active and violated constraints do not change. The predicted decay rates depend on the particular set of active constraints and set of violated constraints, explaining the differences in decay rates observed in practical calculations. In contrast with [19], the cost on constraint violation has a quadratic term in addition to a linear one, and the emphasis of the work is on constraint vioaltion. The dual gradient method that is used implements both linear and quadratic cost on constraint violation without explicitly calculating the slack variables. The result is demonstrated on a MPC application to the AFTI-16 control benchmark [7, 12].

The computational problem and the studied solution algorithm are presented in Sect. 2. In Sect. 3, it is described how the residual in a given iteration is determined by the previous one. The resulting rates of decrease are presented in Sect. 4. Section 5 describes some necessary tools for performing practical calculations, and Sect. 6 presents the computational results that support the theoretical conclusions.

2 Problem Description

A MPC problem is defined by linear system dynamics, a quadratic cost function, and linear constraints. The calculation of the optimal control input for a given initial state is formulated as a quadratic program (QP). Some QPs directly resulting from the MPC problem may not be feasible. The state constraints of the MPC problem, and consequently of the QP, can be softened to ensure feasibility and thus predictability of the control input.

2.1 Model Predictive Control

Consider a discrete time linear system with the dynamics described as

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

where t is the time index, $\mathbf{x}$ is the system state, $\mathbf{u}$ is the system input, the matrices $\mathbf{A}$ and $\mathbf{B}$ model the dynamics. To optimally control the system over a time horizon N, a quadratic cost function J is introduced [7] as

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

The signals are constrained to polyhedra $\mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}$ where $\mathcal{X} = \left\{\mathbf{x} \in \mathbb{R}^l | \mathbf{C}_x{}'\mathbf{x} \preceq \mathbf{b}_x{}'\right\}$, $\mathcal{U} = \{\mathbf{u} \in \mathbb{R}^m | \mathbf{C}_u{}'\mathbf{u} \preceq \mathbf{b}_u{}'\}$. The question of finding the minimizer of the cost for a given value of $\mathbf{x}(0)$ is the QP [2, 19]

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

With the receding horizon implementation, $\mathbf{u}_0$ is applied as the current value of the controller output $\mathbf{u}(0)$.

2.2 Quadratic Program Formulation With Soft Constraints

Working around the infeasibility problem with the constraints-softening approach, we relax the state constraint (3c) while adding a term for its violation to the cost function. We obtain

$$\begin{aligned}
\underset{(\mathbf{x}_0,\ldots,\mathbf{x}_N,\mathbf{u}_0,\ldots,\mathbf{u}_N)}{\text{minimize}} \quad & J\left(\mathbf{x}_k, \mathbf{u}_k\right) + \frac{1}{2}\sum_{k=0}^{N}\left(\mathbf{s}_k^{\mathrm{T}}\mathbf{S}\mathbf{s}_k + \mathbf{s}_{\mathrm{lin}}^{\mathrm{T}}\mathbf{s}_k\right) && \text{(4a)}\\
\text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \ , && \text{(4b)}\\
& \mathbf{C}_x{}'\mathbf{x}_k \preceq \mathbf{b}_x{}' + \mathbf{s}_k \ , && \text{(4c)}\\
& \mathbf{C}_u{}'\mathbf{u}_k \preceq \mathbf{b}_u{}' \ , && \text{(4d)}\\
& \mathbf{x}_0 = \mathbf{x}\left(0\right) \ , && \text{(4e)}\\
& \mathbf{s}_k \succeq \mathbf{0} \ . && \text{(4f)}
\end{aligned}$$

The vectors $\mathbf{s}_k \in \mathbb{R}^p$ are called slack variables, the linear cost on them $\mathbf{s}_{\text{lin}} \in \mathbb{R}^p$ has only positive components, $\mathbf{s}_{\text{lin}} \succ \mathbf{0}$. The matrix $\mathbf{S} \in \mathbb{R}^{p \times p}$ is diagonal positive semidefinite.

We proceed by eliminating the state variables as in Ullmann and Richter [24], using (4b,4e) in (4c) to obtain the condensed form of the QP

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

The vectors $\mathbf{z} \in \mathbb{R}^{(N \times m)}$ and $\mathbf{s} \in \mathbb{R}^{(N \times p)}$ are the optimization variables constructed as

$$\mathbf{z} = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_N \end{bmatrix}, \qquad \mathbf{s} = \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{s}_1 \\ \vdots \\ \mathbf{s}_N \end{bmatrix}.$$

By construction, the vector $\mathbf{w} \in \mathbb{R}^{(N \times p)}$ has only positive components and the matrix $\mathbf{W} \in \mathbb{R}^{(N \times p) \times (N \times p)}$ is diagonal positive semidefinite. The only QP constants dependent on $\mathbf{x}(0)$ are $\mathbf{b}_x$ and $\mathbf{c}$.

2.3 Dual Proximal Gradient Method Algorithm

As described in Perne et al [19] and Perne et al [20], $\mathbf{y}^k$ of the iteration

$$\mathbf{y}^k = -\mathbf{H}^{-1}\left(\mathbf{C}^{\mathrm{T}}\mathbf{v}^k + \mathbf{c}\right) \tag{6a}$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \mathbf{C}\mathbf{y}^k - \widetilde{\text{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}\left(\mathbf{v}^k + \mathbf{C}\mathbf{y}^k\right) \tag{6b}$$

converges to the solution $\mathbf{z}^*$ of the QP (5) as $k \to \infty$. The vectors $\mathbf{y}^k \in \mathbb{R}^{(k \times m)}$ are the approximate primal solution and $\mathbf{v}^k \in \mathbb{R}^r$ is the dual variable, r is the number of all the constraints. The parameters of (5b, 5c) are used to construct $\mathbf{C}$ and $\mathbf{b}$ as

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_x \\ \mathbf{C}_u \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_u \end{bmatrix}. \tag{7}$$

The operator $\widetilde{\text{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}(\mathbf{t})$ is defined by component as

$$\widetilde{\text{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}(\mathbf{t})_i := \begin{cases} t_i & \text{if } i \text{ inactive} \\ b_i & \text{if } i \text{ active} \\ \frac{t_i + W_{jj}b_i - w_j}{W_{jj}+1} & \text{if } i \text{ violated.} \end{cases} \tag{8}$$

For soft constraints, the matrix or vector element with the index j is the one that corresponds to the constraint i. *Inactive* constraints are the ones for which $t_i \leq b_i$, for *active* ones $b_i < t_i$ and if i is soft, $t_i \leq b_i + w_j$. *Violated* are the

soft constraints for which $t_i > b_i + w_j$. The sufficient condition for convergence is that the eigenvalues of $\mathbf{M} := \mathbf{C}\mathbf{H}^{-1}\mathbf{C}^{\mathrm{T}}$ are smaller than or equal to 1, and it can be ensured by scaling the cost function (5a) which does not change the solution of the QP.

3 Behaviour of Residuals of Dual Gradient Method

Consider three iterations, k, $k+1$, $k+2$, for which the sets of the constraints that are active and of constraints that are violated remain constant. Let us define dual residuals to be

$$\boldsymbol{\Delta}^k = \mathbf{v}^{k+1} - \mathbf{v}^k \tag{9a}$$

$$\boldsymbol{\Delta}^{k+1} = \mathbf{v}^{k+2} - \mathbf{v}^{k+1} \tag{9b}$$

like in Perne et al [19] and analyse the relationship between $\boldsymbol{\Delta}^k$ and $\boldsymbol{\Delta}^{k+1}$.

We see from (9b) and from (6b) with the index k advanced for 1 that

$$\boldsymbol{\Delta}^{k+1} = \mathbf{C}\mathbf{y}^{k+1} - \widetilde{\mathrm{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}\left(\mathbf{v}^{k+1} + \mathbf{C}\mathbf{y}^{k+1}\right) \tag{10}$$

and applying (6a) we obtain

$$\boldsymbol{\Delta}^{k+1} = -\mathbf{M}\mathbf{v}^{k+1} - \mathbf{C}\mathbf{H}^{-1}\mathbf{c} - \widetilde{\mathrm{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}\left(\mathbf{v}^{k+1} - \mathbf{M}\mathbf{v}^{k+1} - \mathbf{C}\mathbf{H}^{-1}\mathbf{c}\right) . \tag{11}$$

Substituting $\mathbf{v}^{k+1}$ from (9a), equation (11) further expands into

$$\begin{aligned}\boldsymbol{\Delta}^{k+1} = &- \mathbf{M}\left(\mathbf{v}^k + \boldsymbol{\Delta}^k\right) - \mathbf{C}\mathbf{H}^{-1}\mathbf{c} \\ &- \widetilde{\mathrm{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}\left(\mathbf{v}^k + \boldsymbol{\Delta}^k - \mathbf{M}\left(\mathbf{v}^k + \boldsymbol{\Delta}^k\right) - \mathbf{C}\mathbf{H}^{-1}\mathbf{c}\right) .\end{aligned} \tag{12}$$

Using (11) again, this time with index k in place of $k+1$, it follows from (12) that

$$\boldsymbol{\Delta}^{k+1} = \boldsymbol{\Delta}^k - \mathbf{M}\boldsymbol{\Delta}^k - \boldsymbol{\Delta}_{\mathrm{prox}} . \tag{13}$$

In (13), we substituted

$$\begin{aligned}\boldsymbol{\Delta}_{\mathrm{prox}} := &\widetilde{\mathrm{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}\left(\mathbf{v}^k + \boldsymbol{\Delta}^k - \mathbf{M}\left(\mathbf{v}^k + \boldsymbol{\Delta}^k\right) - \mathbf{C}\mathbf{H}^{-1}\mathbf{c}\right) \\ &- \widetilde{\mathrm{prox}}_{\mathbf{b},\mathbf{W},\mathbf{w}}\left(\mathbf{v}^k - \mathbf{M}\mathbf{v}^k - \mathbf{C}\mathbf{H}^{-1}\mathbf{c}\right) .\end{aligned} \tag{14}$$

Taking into account that the sets of the active and of the violated constraints do not change in the studied iterations, it can be shown from (8) that

$$\left(\boldsymbol{\Delta}_{\mathrm{prox}}\right)_i = \begin{cases} \left(\boldsymbol{\Delta}^k - \mathbf{M}\boldsymbol{\Delta}^k\right)_i & \text{if } i \text{ inactive} \\ 0 & \text{if } i \text{ active} \\ \frac{1}{W_{jj}+1}\left(\boldsymbol{\Delta}^k - \mathbf{M}\boldsymbol{\Delta}^k\right)_i & \text{if } i \text{ violated} . \end{cases} \tag{15}$$

The vector $\boldsymbol{\Delta}_{\mathrm{prox}}$ is thus a linear function of $\boldsymbol{\Delta}^k$ and (13) can be written in the form

$$\boldsymbol{\Delta}^{k+1} = \mathbf{P}\boldsymbol{\Delta}^k . \tag{16}$$

We can see how the matrix $\mathbf{P}$ is constructed from (13) and (15):

- If i is inactive, the i-th line of $\mathbf{P}$ only contains zeros.
- If i is active, the i-th line of $\mathbf{P}$ is equal to the i-th line of $(\mathbf{I}-\mathbf{M})$.
- If i is violated, the i-th line of $\mathbf{P}$ is equal to the i-th line of $(\mathbf{I}-\mathbf{M})$ multiplied by $W_{jj}/\left(W_{jj}+1\right)$.

The symbol $\mathbf{I}$ denotes an identity matrix. To investigate the local decay of residuals, we demonstrate that the eigenvalues of $\mathbf{P}$ are in the interval $[0,1]$.

The matrix $\mathbf{M}$ is positive semidefinite [19] and we ensured that its eigenvalues are in the interval $[0,1]$. By construction, the matrix $(\mathbf{I}-\mathbf{M})$ is positive semidefinite as well and its eigenvalues are also in the interval $[0,1]$. It is evident from construction that $\mathbf{P}$ is a product of a a positive semidefinite diagonal matrix that can be labelled $\mathbf{F}$ and the matrix $(\mathbf{I}-\mathbf{M})$. The diagonal elements of $\mathbf{F}$ are in the interval $[0,1]$ by construction of $\mathbf{P}$. The matrix $\mathbf{P}$ is thus a product of two matrices with eigenvalues in the interval $[0,1]$ so its eigenvalues are in the interval $[0,1]$.

4 Rates of Decrease of Residuals

Equation (16) can be diagonalized using eignedecomposition into

$$\mathbf{d}^{k+1}=\mathbf{D}\mathbf{d}^{k}\ . \tag{17}$$

Here, $\mathbf{D}=\mathbf{Q}^{-1}\mathbf{P}\mathbf{Q}$, $\mathbf{d}^{k,k+1}=\mathbf{Q}^{-1}\boldsymbol{\Delta}^{k,k+1}$, columns of $\mathbf{Q}$ are eignevectors of $\mathbf{P}$, $\mathbf{D}$ is diagonal with eigenvalues λ_i of $\mathbf{P}$ on the diagonal.

In each iteration, the i-th component of $\mathbf{d}^k$ gets multiplied by λ_i. If $0<\lambda_i<1$, the component is decreasing toward 0. As long as all eigenvalues of $\mathbf{P}$ are below 1, the residual is decreasing toward $\mathbf{0}$.

The case when 1 is an eigenvalue of $\mathbf{P}$ deserves further attention. Consider a vector $\boldsymbol{\Delta}$ for which it is $\mathbf{P}\boldsymbol{\Delta}=\boldsymbol{\Delta}$. It follows that

$$\mathbf{F}\left(\mathbf{I}-\mathbf{M}\right)\boldsymbol{\Delta}=\boldsymbol{\Delta}\ . \tag{18}$$

Since eigenvalues of both $\mathbf{F}$ and $(\mathbf{I}-\mathbf{M})$ are in the interval $[0,1]$, this can only be the case if

$$\left(\mathbf{I}-\mathbf{M}\right)\boldsymbol{\Delta}=\boldsymbol{\Delta}\ ,\qquad \mathbf{F}\boldsymbol{\Delta}=\boldsymbol{\Delta}\ . \tag{19}$$

From the left equation it follows that $\mathbf{M}\boldsymbol{\Delta}=\mathbf{0}$, and the right equation tells us that the non-zero components of $\boldsymbol{\Delta}$ all correspond to active constraints (not to violated ones). It is shown in Perne et al [19] that these eigenvectors do not influence the primal solution and they are equal to $\mathbf{0}$ for feasible active sets.

Only eigenvalues of $\mathbf{P}$ that are $\lambda_i<1$ thus influence the decay of residuals influencing the primal solution. The smaller the λ_i, the faster the decay of the corresponding component of $\mathbf{d}^k$ and the corresponding linear combination of components of $\boldsymbol{\Delta}^k$. The slowest component of the residual to decay corresponds to the biggest $\lambda_i<1$; the components of the residual proportional to smaller λ_i's have faster dynamics. If the $\mathbf{P}$ being studied is the final one and the active and violated sets do not change in subsequent iterations, the biggest $\lambda_i<1$ determines the convergence rate.

5 Practical Extensions

In this section, we introduce preconditioning and a way of treating upper and lower bounds on the same signals at the same time. If done properly, preconditioning improves the convergence rate. On the other hand, treating upper and lower bounds together has no influence on the behaviour of the iteration procedure but it reduces the computational complexity. Both improvements can be used together.

5.1 Preconditioning

Consider the following quadratic program:

$$\begin{aligned} \underset{\mathbf{z},\mathbf{s}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z} + \mathbf{c}^{\mathrm{T}}\mathbf{z} + \frac{1}{2}\mathbf{s}^{\mathrm{T}}{\mathbf{E}_x}^{-2}\mathbf{W}\mathbf{s} + {\mathbf{E}_x}^{-1}\mathbf{w}^{\mathrm{T}}\mathbf{s} \\ \text{subject to} \quad & \mathbf{E}_x\mathbf{C}_x\mathbf{z} \preceq \mathbf{E}_x\mathbf{b}_x + \mathbf{s} \;, \\ & \mathbf{E}_u\mathbf{C}_u\mathbf{z} \preceq \mathbf{E}_u\mathbf{b}_u \;, \\ & \mathbf{s} \succeq \mathbf{0} \;. \end{aligned} \tag{20}$$

where $\mathbf{E}_x$ and $\mathbf{E}_u$ are diagonal positive definite matrices and $\mathbf{E} = \begin{bmatrix} \mathbf{E}_x & 0 \\ 0 & \mathbf{E}_u \end{bmatrix}$. The quadratic program (20) is equivalent to (5). It has the same form as well, so it can be solved through the same algorithm (6), applied as

$$\mathbf{y}^k = -\mathbf{H}^{-1}\left(\mathbf{C}^{\mathrm{T}}\mathbf{E}\mathbf{v}^k + \mathbf{c}\right) \tag{21a}$$

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \mathbf{E}\mathbf{C}\mathbf{y}^k - \widetilde{\mathrm{prox}}_{\mathbf{E}\mathbf{b},{\mathbf{E}_x}^{-2}\mathbf{W},{\mathbf{E}_x}^{-1}\mathbf{w}}\left(\mathbf{v}^k + \mathbf{E}\mathbf{C}\mathbf{y}^k\right) \;. \tag{21b}$$

While $\mathbf{y}^k$ converges to the same solution as in (6), provided that eigenvalues of $\mathbf{E}\mathbf{C}\mathbf{H}^{-1}\mathbf{C}^{\mathrm{T}}\mathbf{E}$ are smaller than or equal to 1, the iteration steps and the convergence rate are not the same. The better the choice of $\mathbf{E}$, the smaller the eigenvalues of the encountered sets $\mathbf{P}$, the faster the decay of residuals and the convergence rate.

5.2 Upper and Lower Boundaries

The same signals are often bound from above and from below in MPC, leading to the same linear functionals of the optimization variable in QP having both upper and lower bounds as well [19]. If the QP is given in the form of (5), $\mathbf{C}_x$ and $\mathbf{C}_u$ can thus be written as

$$\mathbf{C}_x = \begin{bmatrix} \mathbf{C}_{x1} \\ -\mathbf{C}_{x1} \end{bmatrix} \;, \qquad \mathbf{C}_u = \begin{bmatrix} \mathbf{C}_{u1} \\ -\mathbf{C}_{u1} \end{bmatrix} \;. \tag{22}$$

The QP (5) can be reformulated as

$$\begin{aligned} \underset{\mathbf{z},\mathbf{s}}{\text{minimize}} \quad & \frac{1}{2}\mathbf{z}^{\mathrm{T}}\mathbf{H}\mathbf{z} + \mathbf{c}^{\mathrm{T}}\mathbf{z} + \frac{1}{2}\mathbf{s}_1^{\mathrm{T}}\mathbf{W}_1\mathbf{s}_1 + \mathbf{w}_1^{\mathrm{T}}\mathbf{s}_1 && \text{(23a)} \\ \text{subject to} \quad & \mathbf{b}_{x1} - \mathbf{s}_1 \preceq \mathbf{C}_{x1}\mathbf{z} \preceq \mathbf{b}_{x2} + \mathbf{s}_1 \ , && \text{(23b)} \\ & \mathbf{b}_{u1} \preceq \mathbf{C}_{u1}\mathbf{z} \preceq \mathbf{b}_{u2} \ , && \text{(23c)} \\ & \mathbf{s} \succeq \mathbf{0} \ . && \text{(23d)} \end{aligned}$$

A solver for this QP can be implemented more efficiently because some problem dimensions are smaller. The implementation is

$$\begin{aligned} \mathbf{y}^k &= -\mathbf{H}^{-1}\left(\mathbf{C}_1{}^{\mathrm{T}}\mathbf{v}^k + \mathbf{c}\right) && \text{(24a)} \\ \mathbf{v}^{k+1} &= \mathbf{v}^k + \mathbf{C}_1\mathbf{y}^k - \widetilde{\mathrm{prox}}_{\mathbf{b}_2,\mathbf{W}_1,\mathbf{w}_1}\left(\mathbf{v}^k + \mathbf{C}_1\mathbf{y}^k\right) \\ &\quad - \widetilde{\mathrm{prox}}_{-\mathbf{b}_1,\mathbf{W}_1,\mathbf{w}_1}\left(\mathbf{v}^k - \mathbf{C}_1\mathbf{y}^k\right) \ , && \text{(24b)} \end{aligned}$$

where $\mathbf{C}_1 = \begin{bmatrix} \mathbf{C}_{x1} \\ \mathbf{C}_{u1} \end{bmatrix}$ and the dimension of $\mathbf{v}$ is half what it is in (6). In the definition of $\mathbf{M}$, the matrix $\mathbf{C}$ is replaced with $\mathbf{C}_1$. If preconditioning is used together with combining upper and lower boundaries, it is done with matrix $\mathbf{E}$ of half the size as well.

6 Example

A discrete-time form of the AFTI-16 benchmark model as in [7] has the system matrices

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

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

in (1). The constraints are:

$$\begin{aligned} \mathcal{X} &= \left\{\mathbf{x} \in \mathbb{R}^4; -0.5 - s_1 \le x_2 \le 0.5 + s_1, \ -100 - s_2 \le x_4 \le 100 + s_2\right\} \\ \mathcal{U} &= \left\{\mathbf{u} \in \mathbb{R}^2; -25 \le u_1 \le 25, \ -25 \le u_2 \le 25\right\} \ , \end{aligned} \qquad \text{(25)}$$

where s_1, s_2 stand for slack variables corresponding to soft constrained components of the system state. The cost matrices are

$$\begin{aligned} \mathbf{Q} &= \mathrm{diag}(10^{-4}, 10^2, 10^{-3}, 10^2) \ , \\ \mathbf{R} &= \mathrm{diag}(10^{-2}, 10^{-2}) \ . \end{aligned} \qquad \text{(26)}$$

Following the procedure from Ullmann and Richter [24] implemented in *QPgen* [8, 22] modified as in [20], we obtain QPs in the form (5) and then (23) for $N = 10$. Cost on slack has to be chosen: we choose $\mathbf{W}_1 = 1000 \times \mathbf{I} \in \mathbb{R}^{20\times 20}$, $\mathbf{w}_1$ is a vector of length 20 with all components equal to 1300. The linear cost vector $\mathbf{c}$ reflects the chosen reference vector $\mathbf{x}_{\text{ref}} = [0, 0, 0, 10]^{\text{T}}$ and the current state. We simulate 10 samples and the initial condition is $\mathbf{x}_0 = [0, 0, 0, 0]^{\text{T}}$. The matrices $\mathbf{C} \in \mathbb{R}^{40\times 20}$ and $\mathbf{H} \in \mathbb{R}^{20\times 20}$ are constant. The preconditioning diagonal matrix $\mathbf{E}$ is chosen so as to minimize the condition number of the non-singular part of $\mathbf{M}$ while setting the highest eigenvalue of $\mathbf{M}$ to 1. *QPgen* finds $\mathbf{E} = \text{diag}(10.4796, 3.5413, 9.9973, 10.0080, 9.9987, 10.0005, 10.0000, 10.0037, 9.9990, 9.9997, 10.0001, 10.0033, 9.9989, 9.9979, 10.0003, 10.0036, 9.9999, 9.9965, 9.9972, 10.0034, 0.2058, 0.0918, 0.1003, 0.1000, 0.1007, 0.1001, 0.1005, 0.1000, 0.1004, 0.1001, 0.1007, 0.1000, 0.1004, 0.1001, 0.1009, 0.0999, 0.1004, 0.1000, 0.1013, 0.1000)$. The model is initially simulated in closed loop with 10^6 iterations in every sample. The system state $\mathbf{x}$ is recorded at every sample and used as the input initial state for observing convergence of the resulting QPs. Algorithm behaviour is analysed in all samples for 1 to 2000 iterations.

Figure 1 shows the active and violated constraints through the iterations. The white columns delimit samples.

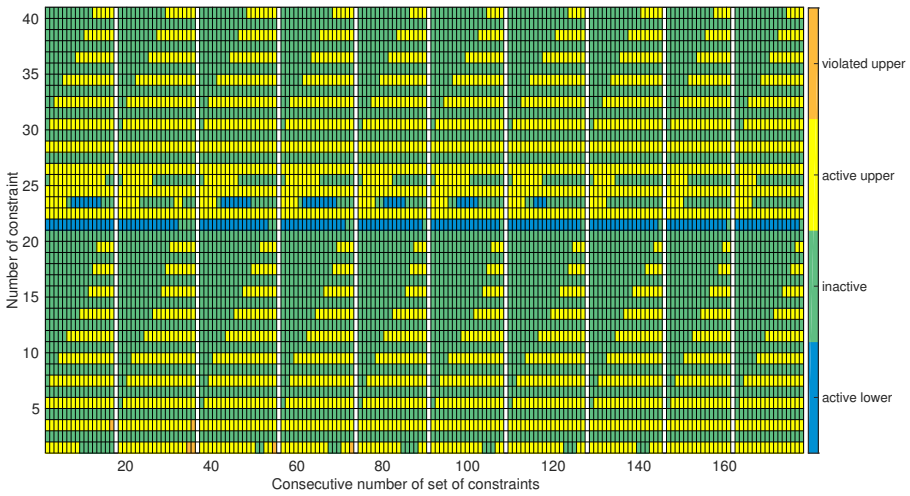

Fig. 1. Activity of constraints for the first 10 samples. The constraints are listed from the bottom to the top: the first 20 correspond to the soft constraints on $\mathbf{x}$ from 1st to N-th time step within the prediction horizon, the following 20 are from constraints on $\mathbf{u}$ for 10 steps and are hard. Samples are listed from left to right, the white columns separate blocks corresponding to different samples. For a given sample, the first list of active and violated constraints is in the leftmost column of the block and every change in the activity or violation during iterations results in a new column

In Fig. 2, convergence through the active set changes in sample 2 is shown graphically.

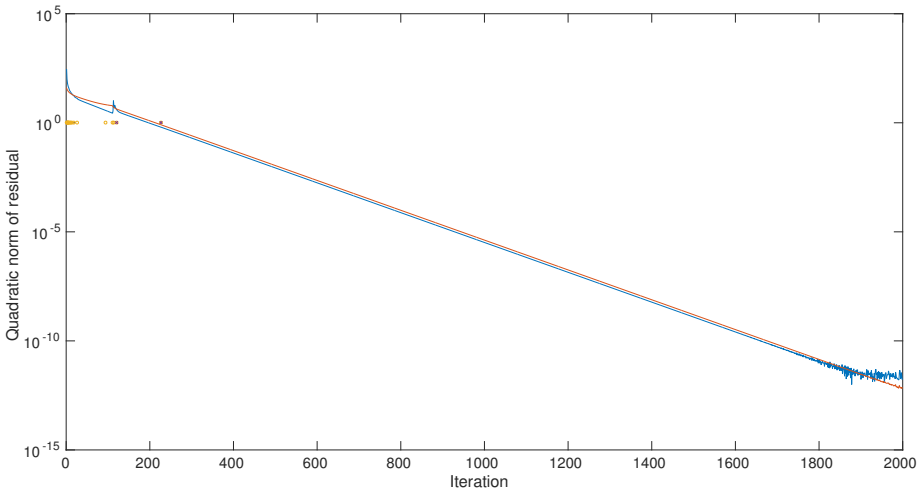

Fig. 2. Convergence of the quadratic norm of the primal (blue) and the dual (red) solution in sample 2 as a function of the iteration number. The yellow circles mark iterations in which the set of active constraints changes, the purple x's mark changes in the set of violated constraints

We see from Fig. 2 that the last two changes in the sets of active and violated constraints in sample 2 happen in iterations 121 and 227. In Fig. 1, we see that some constraints are violated both from iteration 121 on and from 227 on. These two intervals are thus chosen for further study. The dual residual is decomposed into components corresponding to eigenvectors of the matrix $\mathbf{P}$ corresponding to both sets of active and violated constraints and plotted in Figs. 3 and 4. Each component seems to decay exponentially in the relevant interval, just as predicted by (17). We can also check the decay rates. The highest three eigenvalues of $\mathbf{P}$ for iterations 121 to 226 are $\lambda_1 = 0.9846$, $\lambda_2 = 0.8898$, $\lambda_3 = 0.8673$. The ratio ${d_i}^{226}/{d_i}^{121}$ is expected to be equal to ${\lambda_i}^{226-121}$ and in all three cases ${\lambda_i}^{226-121}/\left({d_i}^{226}/{d_i}^{121}\right) = 1$ is accurate to 6 decimal places. Similarly, for iterations from 227 on, the highest three eigenvalues of $\mathbf{P}$ are $\lambda_1 = 0.9844$, $\lambda_2 = 0.8887$, $\lambda_3 = 0.8646$. Equivalent calculations show that the ratio of λ_i to the ratio of the components d_i in different iterations is 1 to 3 decimal places. The initial iteration in observing the decrease of d_i is 227 in all three cases and for the final iteration we take the last one where the residual is above 10^{-10} to avoid numerical errors. The final iteration is number 1675, 373, and 347 respectively.

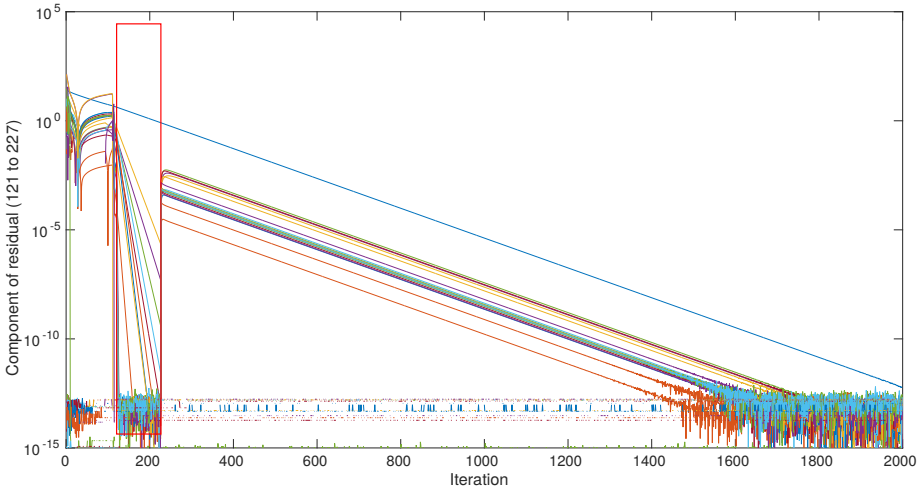

Fig. 3. The components of the dual residual $\mathbf{d}^k$ between iterations 121 and 226 (red frame) for sample 2 as a function of the iteration number

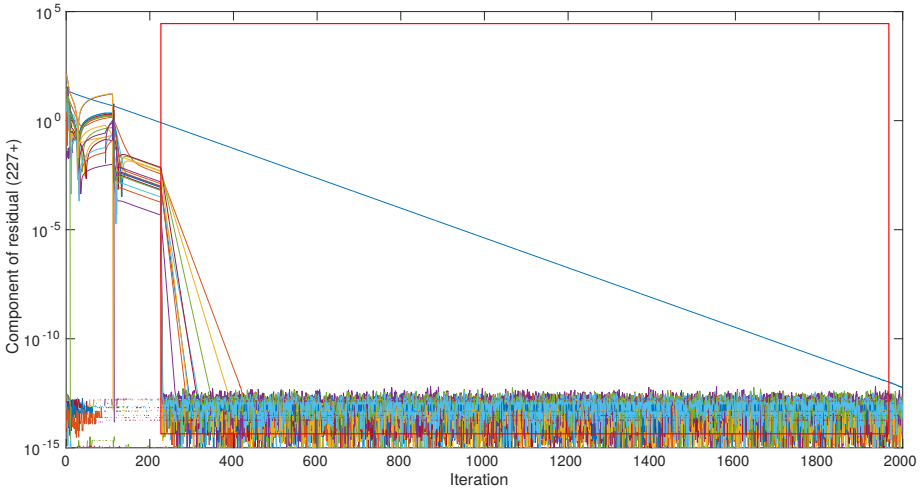

Fig. 4. The components of the dual residual $\mathbf{d}^k$ from iteration 227 on (red frame) for sample 2 as a function of the iteration number

7 Conclusions

The local decay of residuals when solving a QP a gradient method with linear and quadratic cost on violation of soft constraints is predicted. It is determined only by the system matrices and the current sets of active and violated constraints. It is independent from the vectors defining the QP, or in the case of MPC case, it is not directly dependent on the system state, the reference, and the limit values. The predictions are confirmed by a numerical example, explaining varying rates of convergence through samples and iterations.

The local decay of residuals is determined by the eigenvalue of the matrix $\mathbf{P}$ that is the highest among the ones that are smaller than 1. For a given QP, the matrix $\mathbf{P}$ is determined by the set of active constraints and the set of violated constraints, so the set of possible $\mathbf{P}$ is limited. Thus there exists an upper bound smaller than 1 for the eigenvalues of $\mathbf{P}$ smaller than 1. The better one is able to select the choices of possible matrices $\mathbf{P}$, the stricter upper bound could be obtained.

Acknowledgements 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.

Bibliography

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

[2] Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge University Press, New York, NY, USA

[3] 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

[4] Ferreau HJ, Bock HG, Diehl M (2008) An online active set strategy to overcome the limitations of explicit MPC. International Journal of Robust and Nonlinear Control 18(8):816–830, DOI 10.1002/rnc.1251, URL http://dx.doi.org/10.1002/rnc.1251

[5] 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, URL http://dx.doi.org/10.1007/s12532-014-0071-1

[6] Gerkšič S, Tommasi GD (2014) Improving magnetic plasma control for ITER. Fusion Engineering and Design 89(910):2477 – 2488, DOI http://dx.doi.org/10.1016/j.fusengdes.2013.12.034, URL http://www.sciencedirect.com/science/article/pii/S0920379613007485, proceedings of the 11th International Symposium on Fusion Nuclear Technology-11 (ISFNT-11) Barcelona, Spain, 15-20 September, 2013

[7] Giselsson P (2013) Improving Fast Dual Ascent for MPC - Part II: The Embedded Case. ArXiv e-prints 1312.3013

[8] 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, URL http://dx.doi.org/10.1109/CDC.2014.7040176

[9] 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, URL http://dx.doi.org/10.1016/j.automatica.2015.09.010

[10] 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 Systems Technology 22(3)

[11] Hovd M, Stoican F (2014) On the design of exact penalty functions for MPC using mixed integer programming. Computers & Chemical Engineering 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

[12] Kapasouris P, Athans M, Stein G (1990) Design of feedback control systems for unstable plants with saturating actuators. In: Proceedings of the IFAC Symposium on Nonlinear Control System Design, Pergamon Press, p 302 307

[13] 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

[14] Kouzoupis D (2014) Complexity of First-Order Methods for Fast Embedded Model Predictive Control (Master Thesis). Eidgenssische Technische Hochschule, Zrich

[15] Mattingley J, Boyd S (2012) CVXGEN: a code generator for embedded convex optimization. Optimization and Engineering 13(1):1–27, DOI 10.1007/s11081-011-9176-9, URL http://dx.doi.org/10.1007/s11081-011-9176-9

[16] Mattingley J, Wang Y, Boyd S (2011) Receding horizon control, automatic generation of high-speed solvers. IEEE Control Syst Mag 31:52 –65

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

[18] 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, URL http://dx.doi.org/10.1016/j.automatica.2015.03.002

[19] Perne M, Gerkšič S, Pregelj B (2017) Local decay of residuals in dual gradient method applied to MPC studied using active set approach. In: Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics, pp 54–63, DOI 10.5220/0006416500540063

[20] Perne M, Gerkšič S, Pregelj B (2017) Soft inequality constraints in gradient method and fast gradient method for quadratic programming, manuscript in preparation

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

[22] QPgen (2014) QPgen. URL www.control.lth.se/QPgen/, accessed 24 January 2017

[23] Richter S (2012) Computational Complexity Certification of Gradient Methods for Real-Time Model Predictive Control (Dissertation). Eidgenssische Technische Hochschule, Zrich

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

[25] Wang Y, Boyd S (2010) Fast model predictive control using online optimization. IEEE Trans Control Syst Techn 18:267–278

[26] 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

[27] Zheng A, Morari M (1995) Stability of model predictive control with mixed constraints. IEEE Transactions on Automatic Control 40(10):1818–1823, DOI 10.1109/9.467664