Combined Gradient and Newton Projection Quadratic Programming Solver for MPC *

O. Šantin ∗ V. Havlena ∗

∗ Department of Control Engineering, Czech Technical University, Prague, CZ, (e-mail: {santio1,havlena}@fel.cvut.cz).

Abstract: The objective of this paper is to present an effective on-line solver for simple constrained quadratic programming (QP) which arises in linear model predictive control (MPC) framework. In MPC, the QP is solved at each sampling time, thus a fast solver must be used for short sampling times in real-time applications. The multi-parametric quadratic programming (mp-QP) approach (explicit solution) is impossible to use for larger systems due to the memory limitation. On the other hand, the presented approach is well suitable even for medium scale systems with short sampling time, since it is based on combination of gradient and Newton projection algorithm which is very close to optimum in a very few iterations and the computation of the Newton step is not involved at each iteration.

Keywords: Quadratic programming; Quadratic optimal regulators; Predictive control; Gradient methods; Real-time control

1. INTRODUCTION

Model predictive control (MPC) has become an accepted standard in the process industries [Bemporad et al., 2002a] and it has been successfully implemented in many commercial applications. The greatest strength of MPC is an explicit way in which constraints can be incorporated in multivariable control problem formulation. In MPC, the current control input is obtained by solving on-line at each sampling instant, a finite horizon open-loop optimal control problem (typically a constrained quadratic program (QP)), using the current state of the system as the initial state [Mayne et al., 2000].

It has been shown in Bemporad et al. [2002b], Borrelli et al. [2010], that optimal control problem can be defined as the multi-parametric quadratic programming (mp-QP) and solved off-line to obtain the explicit piecewise affine function of system state to reduce the controller to the look-up table process. The limitation of this approach is mainly the memory requirement since it grows exponentially with the number of feasible combinations of active constraints. Therefore this method works well for systems with small state and input dimensions (say no more than 5) and short horizons [Wang and Boyd, 2010].

As the computational power has increased in recent years and since there was a significant progress in optimization techniques, the on-line solution of the MPC has made way even into control applications with short sampling intervals [Richter et al., 2010]. There are generally two categories of on-line iterative solvers for MPC: Interior point (IP) and Active set method (ASM) [Richter et al., 2010]. The IP involves relatively small number of iterations, but at each iteration the linear system of equations must be solved in $O(n^3)$, thus each iteration is relatively expensive. The effective approach in IP was reported in Wang and Boyd [2010], where the structure of the MPC was exploited as well as the warm-starting (starting the algorithm from the previous solution) and the early stopping of the iterations. On the other hand, ASM is very effective in general since each iteration can be performed in $O(n^2)$ (when the updates of the factors are used), but it involves many iterations when the active set changes a lot if the warm-start is considered. An effective implementation of ASM which additionally use the piece-wise affine structure of the explicit MPC solution is discussed in Ferreau et al. [2007].

In MPC framework, the input to the controlled system is in practice constrained typically by the simple box constraints (minimum and maximum), since they represent the limits of the actuators. Thus the optimization problem which must be solved at each sample time is also constrained by the box constraints. As it will be stated next, the gradient projection approach (GP) developed in Goldstein [1964] is very efficient for this type of problem since it enables at each iteration more rapid changes in the working set then ASM.

The GP was already used for MPC: e.g. in Axehill and Hansson [2008] the combination of Newton’s method and GP was used to overcome slow convergence of GP and resulting algorithm was applied for the dual MPC to obtain the box constraints even for polyhedral form of primal constraints. In Moré [1989] the GP was combined with conjugate gradient (CG) method and algorithm that converges in less than 15 iterations (even for large scale problems like 15000 variables) was obtained.

The problem of methods mentioned above is computational difficulty of obtaining the improvement point which is needed at each iteration (Newton or CG stage respectively), since the obtaining of the improvement point leads to computation of the Newton step for QP in $O(n^4)$.

* The authors express gratitude for providing valuable comments and suggestions to Jaroslav Pekar, Honeywell Prague Laboratory. Work was supported by MSM6840770038 “Decision and control in process control III”.

Copyright by the International Federation of Automatic Control (IFAC)
On the other hand, in Bertsekas [1976], a super-linearly
convergent algorithm based on GP was introduced. It uses
the fact, that after a certain number of iterations of GP,
the algorithm reduces to the standard steepest descent
method on the subspace defined by the working set. Thus,
it is possible to switch to the Newton’s method once the
probable optimal working set is identified by the GP to
obtain super-linear convergence property. Therefore, the
computation of the Newton step is not necessary at each
iteration, which makes this algorithm very attractive for
fast MPC. Thus, this paper addresses modification of this
algorithm in the context of MPC.

Recently, a different way how to create very fast algorithm
for MPC was presented in Richter et al. [2010], where the
GP was used to set-up the optimal convergent gradient
methods and fast MPC algorithm for box constrained
problems was obtained. Here, the algorithm converges only
linearly, thus many iterations are needed but each iteration
is very cheap, since only the gradient evaluation must be
performed.

The paper is organized as follows: a finite-time optimal
control problem that motivated the presented approach
is formulated in Section 2. The formal statement of the
problem is given in Section 3 and is followed by a brief
description of GP in Section 4 and its modification for
simple constrained MPC in Section 5. Finally, in Section 6
the effectiveness of the proposed method is confirmed by
numerical experiments.

1.1 Notation

Through this paper, if not defined otherwise, italic let-
ters denote scalars, bold italic letters denote vectors and
matrices (e.g. \( \mathbf{g} \), \( \mathbf{A} \)), while upper case calligraphic letters
denote sets (e.g. \( \mathcal{A} \), \( \mathcal{W} \)) and complexities (e.g. \( \mathcal{O} \), \( \mathcal{M} \)). For
a matrix (vector) \( \mathbf{A} \), \( \mathbf{A}_i \) denotes i-th row(element). Given
the matrix \( \mathbf{G} \in \mathbb{R}^{n \times m} \), then for any set \( \mathcal{A} \subseteq \{1, \ldots, n\} \),
\( \mathcal{B} \subseteq \{1, \ldots, m\} \), \( \mathbf{G}_{\mathcal{A}} \) denotes the submatrix of \( \mathbf{G} \) con-
sisting of the rows indexed by \( \mathcal{A} \), while \( \mathbf{G}_{\mathcal{A}\mathcal{B}} \) denotes the submatrix of \( \mathbf{G} \) consist-
ing of the rows indexed by \( \mathcal{A} \) and columns indexed by \( \mathcal{B} \). \( \mathbf{Q} \succeq 0 \) denotes positive semidefinitely (resp. \( \mathbf{Q} \succeq 0 \) positive semidefiniteness) of a
square matrix \( \mathbf{Q} \). The superindex \((.)^{(k)}\) denotes variable
(\( k \)) at \( k \)-th iteration of the algorithm. The computational
complexities are considered in floating point operations.

2. MODEL PREDICTIVE CONTROL

Consider a discrete-time linear time-invariant system
\[ \mathbf{x}(t+1) = \mathbf{Mx}(t) + \mathbf{Nu}(t) \quad (1) \]
for all time instants \( t \geq 0 \). Let \( n_x \in \mathbb{N}, n_u \in \mathbb{N} \) are the
number of states and inputs respectively, \( \mathbf{x} \in \mathbb{R}^{n_x} \) is the
state vector, \( \mathbf{u} \in \mathbb{R}^{n_u} \) is the input vector, \( \mathbf{M} \in \mathbb{R}^{n_x \times n_x} \),
\( \mathbf{N} \in \mathbb{R}^{n_x \times n_u} \). Assume that full measurement of the state
\( \mathbf{x}(t) \) is available at time \( t \geq 0 \). Consider the problem of
regulating the system state to the origin under input
constraints. MPC can be then stated as
\[
U^* = \arg \min_{U} \frac{1}{2} \mathbf{x}_N^T \mathbf{Q}_N \mathbf{x}_N + \frac{1}{2} \sum_{k=0}^{N-1} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + u_k^T \mathbf{R} u_k
\]
s.t. \[ \mathbf{x}_{k+1} = \mathbf{Mx}_k + \mathbf{Nu}_k, \]
\[ u_k \leq u_k \leq \bar{u}_k, \]
\[ k = 0, \ldots, N-1, \quad (2) \]
where \( \mathbf{x}_0 = \mathbf{x}(0) \) is the initial state, \( N \in \mathbb{N} \) is the
prediction horizon length, \( \mathbf{Q}_N \succeq 0, \mathbf{Q} \succeq 0, \mathbf{R} \succ 0, \)
\[ \mathbf{U} = [\mathbf{u}_0^T, \ldots, \mathbf{u}_{N-1}^T]^T \in \mathbb{R}^{n_u} \] is the
optimization vector, \( n = n_u N, \mathbf{x}_t \) denotes the state at time \( t \)
if the initial state is \( \mathbf{x}_0 \) and the control sequence \( \mathbf{U} \)
is applied to the system.

MPC control law is then obtained according to the reced-
ing horizon principle [Mayne et al., 2000] so that at the
current time \( t \geq 0 \) the problem (2) is solved for \( \mathbf{x}_0 = \mathbf{x}(t) \),
and only the first control move of optimal trajectory \( \mathbf{U}^* \)
is applied as an input to system (1).

Remark 1. The computational complexity of the MPC
control law depends on the degrees of freedom which
depend on the number of inputs \( n_u \) multiplied with the
prediction horizon length \( N \) in standard formulation of
MPC. In order to deal with this issue, a common approach
is to use an input blocking strategy, i.e. use different
control and prediction horizons. Then the inputs are kept
constant beyond a certain point in the prediction horizon
[Cagniard et al., 2007] and the degree of freedom is
reduced to \( n_u N_c \) where \( N_c \) is control horizon.

3. STATEMENT OF THE PROBLEM

Problem (2) can be rewritten as a strictly convex quadratic
programming (QP) problem subject to simple constraints
\[
\mathbf{f}^*(\mathbf{x}) = \min_{\mathbf{U}} \frac{1}{2} \mathbf{U}^T \mathbf{H} \mathbf{U} + \mathbf{x}^T \mathbf{F} \mathbf{U} \quad (3a)
\]
s.t. \( \mathbf{U} \leq \mathbf{U} \leq \bar{\mathbf{U}}, \quad (3b) \)
where \( \mathbf{x} = \mathbf{x}_0, \mathbf{U} \in \mathbb{R}^{n_u} \) is the optimization variable,
\( \mathbf{H} \in \mathbb{R}^{n_u \times n_u}, \mathbf{H} \succ 0 \) and \( \mathbf{H}, \mathbf{F}, \mathbf{U}, \bar{\mathbf{U}} \) can be easily obtained from (2) (see Bemporad et al. [2002b] for details).
The constant term in (3) is omitted since it does not influence the
minimizer.

Remark 2. Note that the proposed algorithm can be ap-
plied for problems with additional constraints in a form
\[ \mathbf{G} \mathbf{U} \leq \mathbf{S}_e \mathbf{x} + \mathbf{w}_e \quad (4) \]
where \( \mathbf{G}_e \in \mathbb{R}^{m_e \times n_u}, \mathbf{S}_e \in \mathbb{R}^{m_e \times n_x}, \mathbf{w}_e \in \mathbb{R}^{m_e} \) defines
the general constraints. Considering additional constraints in
form (4) as a soft constraints (those which can be violated,
but any violation is penalized in the objective function, see
Mayne et al. [2000]), a problem can be rewritten in form
with simple constraints as
\[
\mathbf{f}^*(\mathbf{x}) = \min_{\mathbf{U} \in \mathbb{R}^{n_u}, \mathbf{e} \in \mathbb{R}^{m_e}, \mathbf{Q}_e \succeq 0} \frac{1}{2} \mathbf{x}^T \mathbf{F} \mathbf{U} + (1/2) \mathbf{e}^T \mathbf{Q}_e \mathbf{e} \quad (5)
\]
s.t. \[ \mathbf{U} \in \mathbb{R}^{n_u}, \mathbf{e} \succeq 0, \quad (5) \]
where \( \mathbf{e} = \mathbf{G}_e \mathbf{U} - \mathbf{S}_e \mathbf{x} - \mathbf{w}_e, \mathbf{e} \in \mathbb{R}^{m_e} \) and \( \mathbf{Q}_e \in \mathbb{R}^{m_e \times m_e}, \mathbf{Q}_e \succ 0 \) is a penalty of the violation of the
constraints (4).

Remark 3. It should be also noted that the vector \( \mathbf{x} \)
represents the parameters of the problem (e.g. references,
state of the observers, etc.) rather than only the state
vector of the system (1) in practice.

5568
Since the optimal input trajectory $U^*$ is affected by the parameter vector $x$, the MPC strategy can be performed either by solving (3) on-line or considering (3) as mp-QP and solving problem (3) off-line for all $x$ within a given range of values [Bemporad et al., 2002b]. The mp-QP approach is possible only for small systems due to memory limitation. On the other hand, the on-line solution of (3) involves mainly the requirements of powerful processor and, relatively to previous approach, no memory requirements. An effective on-line solution is presented in the next sections.

4. GRADIENT PROJECTION

The active set method (ASM) is one of the most common method for solving the QP (3) for a fixed $x$ [Borrelli et al., 2010]. The main idea of ASM is to identify the active set of constraints at the solution in a finite number of iterations. The ASM maintains an estimate of the active set $\mathcal{N}^{(k)}$ (called working set), which is a linearly independent set of constraints that are satisfied at the beginning of each iteration. See detailed description of ASM e.g. in Nocedal and Wright [1999].

A drawback with ASM is that the working set is changing very slowly [Axehill and Hansson, 2008]. Only one change in the working set is performed per iteration. This fact may result in slow convergence for problems which arise in optimal control, where the control variables are often at the constraints for large portion of the time interval [Bertsekas, 1979].

On the other hand, the gradient projection method (GP) is almost as simple as ASM and provides more rapid change of the working set than ASM and it also inherits many good properties as warm start possibility.

The GP was introduced in Goldstein [1964] as a generalization of the steepest descent method of optimization problems with convex constraints. It was shown, that under the non-degeneracy assumption, the algorithm of GP identifies the optimal active set of constraints in a finite number of iterations [Burke and Moré, 1988, Moré, 1989]. In general, the GP solves problem

$$\min_z f(z) \quad \text{s.t.} \quad z \in \mathcal{Z},$$

where $f(z)$ is a continuously at least once differentiable function $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{Z} \subseteq \mathbb{R}^n$ is a nonempty closed convex set.

The projection of $y$ onto $\mathcal{Z}$ is the mapping $P : \mathbb{R}^n \to \mathcal{Z}$ defined by

$$P(y) = \arg \min_z ||z - y|| \quad \text{s.t.} \quad z \in \mathcal{Z}. \quad (7)$$

The gradient projection algorithm is then defined by

$$z^{(k+1)} = P(z^{(k)} - \alpha^{(k)} \nabla f(z^{(k)})),$$  

where $\alpha^{(k)} > 0$ is the step-size, and $\nabla f(z^{(k)})$ is the gradient of $f$ at $z^{(k)}$.

One of the main drawback of GP is that the computation of the projection (7) can be expensive for general type of constraints [Moré, 1989] since it may lead to the QP in general. Thus GP is effectively limited to problems involving simple constraint sets such as positive orthant, spheres, Cartesian products of spheres or the box constraints [Bertsekas, 1976]. The box constraints are defined as a closed convex set

$$\mathcal{Z} = \{ z \in \mathbb{R}^n : l \leq z \leq u \}, \quad (9)$$

for some vectors $l \in \mathbb{R}^n$, $u \in \mathbb{R}^n$. The computation of the projection of $y$ onto such set can be done in $O(n)$ [Moré, 1989] since

$$P(y) = \text{mid}(l, u, y), \quad (10)$$

where $\text{mid}(l, u, y)$ is the vector whose $i$-th component is the median of the set $\{l_i, u_i, y_i\}$.

Another well-known drawback of GP is that it shares the slow convergence rates (linear) with the steepest descent method, since after a finite number of iterations, GP becomes a version of the steepest descent method restricted to the binding constraint manifold. Thus, the rate of convergence of GP is governed by the eigenvalue structure of the Hessian of (3) over the subspace of active constraints in optimum [Bertsekas, 1976].

To overcome this, a super-linearly convergent modifications of GP were presented in the literature: the Newton’s step can be used for an improvement of the projected point at each iteration of the algorithm (see Nocedal and Wright [1999]) or the projection of gradient can be switched to Newton method once the subspace of probable optimal binding constraints are reached by the algorithm (e.g. Bertsekas [1976]). Furthermore, the Newton’s direction can be also projected onto the feasible set [Bertsekas, 1982, Axehill and Hansson, 2008] to obtain faster rate of convergence. The former approach is briefly discussed in the next section since the latter and also the presented algorithm coming out of it as a limit approach.

4.1 Algorithm description

At the beginning of each iteration, the current point is checked for the Karush-Kuhn-Tucker (KKT) optimality conditions. After that, the GP algorithm consists of two main stages [Axehill and Hansson, 2008]: 1) finding of the Cauchy point by projection of the gradient and 2) finding of the point which improves the cost function.

In the first stage, a line search optimization along the negative gradient direction is performed for obtaining $a^{(k)}$. Whenever a previously non-active constraints is encountered, the search direction is bent-off in a way that the constraints remain satisfied.

Note, that when the search line is bent-off, the optimization problem to find the global minimizer along the piecewise linear path is, in general, no longer a convex optimization problem even if the minimized function is convex [Axehill, 2008]. Thus, the finding of the global minimizer along the projected path is not used in GP. Instead, either the generalized Armijo procedure for GP [Bertsekas, 1976] or the exact line search is used to find the first local minimizer [Axehill, 2008], called the Cauchy point $z_c$. An efficient exact line search procedure, also used in this work, can be found in Nocedal and Wright [1999].

In the second stage, an improvement step of the Cauchy point is computed for better convergence rate [Axehill and Hansson, 2008] by solving smaller problem
Algorithm 1 Gradient projection algorithm [Nocedal and Wright, 1999]
1: Compute a feasible starting point $z^{(0)}$.
2: for $k = 0$ to $k_{\text{max}}$ do
3: if $z^{(k)}$ satisfies the KKT conditions for (6) then
4: Stop with $z^* = z^{(k)}$.
5: end if
6: Starting from $z^{(k)}$ project the gradient to find the Cauchy point $z^+_C$.
7: Find an approximate feasible solution $z_+^{(k)}$ of (11) such that $f(z_+^{(k)}) \leq f(z^+_C)$.
8: $z^{(k)} = z_+^{(k)}$
9: end for

$$z_+^{(k)} = \arg \min_{z \in \mathcal{Z}} f(z^{(k)}) \quad (11a)$$
$$\text{s.t. } z_i^{(k)} = z_i^{C_i}, \quad i \in \mathcal{W}^{(k)} \quad (11b)$$
$$z_i^{(k)} \in \mathcal{Z}, \quad i \notin \mathcal{W}^{(k)} \quad (11c)$$

where $\mathcal{W}^{(k)}$ denotes the working set in the Cauchy point $z^+_C$ at $k$-th iteration of the algorithm. Since the Cauchy point itself guarantees the global convergence of the algorithm it is possible to solve (11) only approximately [Nocedal and Wright, 1999]. It is only necessary that the approximate solution $z_+^{(k)}$ is feasible and not worse then $z^{(k)}$ with respect to (6). A common approach for large-scale problems is to disregard the constraints (11c) and run the conjugate gradient method on the subspace defined by the active constraints corresponding to the Cauchy point, see Moré [1989]. An alternative approach was presented in Axehill and Hansson [2008], where the constraints (11c) were also disregarded and problem was solved using a Riccati recursion to obtain the Newton’s direction which can be projected onto the inequality constraints of (6) as was discussed in Bertsekas [1976]. Algorithm 1 summarizes the super-linearly convergent modification of GP algorithm.

5. GRADIENT PROJECTION TAILORED FOR MPC

In this section Algorithm 1 will be tailored for QP (3) which arises from MPC. Since the parameter vector $x$ is usually obtained by noisy measurements (or obtained from inaccurate observer) and in case of strongly disturbed measurements it may not be even reasonable to solve the corresponding QP exactly, but rather use the early termination of the iteration of the optimization algorithm after a certain number of iteration [Ferreau et al., 2007].

**Remark 4.** Note, that the early termination of the GP algorithm is possible since it is a primary feasible algorithm, therefore each iteration produces a feasible point with respect to primal constraints.

Moreover, in Ferreau et al. [2007] it was shown, that increasing the control horizon from two to five had almost no influence on the tracking performance for particular practical problem. Although this is not true in general, many practical applications show that it is not reasonable to choose large control horizon. Since it may lead to drastic increase of the computational burden which has to be done at each sampling time and it gives no performance improvement. Of course the simulations and experiments are needed to certify this fact for particular problem. This observation is mainly due to insufficient model accuracy which arises in many practical applications where highly non-linear system is controlled.

Considering points above, the modification of the algorithm of Bertsekas [1976] is proposed in this paper. The differences between the algorithms are that the algorithm presented in this paper uses the exact line search (see Nocedal and Wright [1999]) to find the first local minimizer during the computing of the Cauchy point rather than the usage of generalized Armijo step-size rule. Next, since the Newton step is always descent direction for convex QP, a necessary test to ensure that the Newton step leads to decreasing of the cost function can also be left out. The presented algorithm also uses the early termination of the iteration, as it was suggested in Ferreau et al. [2007]. The warm-start of the presented algorithm is also possible.

**5.1 Algorithm description**

At the beginning of the presented algorithm (see Algorithm 2), the feasibility of the starting point is repaired mainly due to possibility of changes of the input bounds (3b) in conjunction with warm-start. Note, that this can be done in $O(n)$ by (10).

Then, the projected trial step along the negative gradient from the current point $U^{(k)}$ is done with sufficiently small fixed step length $c > 0$. It can be done by (8) and (10) as $U^{(k)}_T[c]$ with a little abuse of notation. After that, two situations are possible: a) the current working set $\mathcal{W}^{(k)}$ is the same as active constraints of the projected point $U^{(k)}_T[c]$, b) the sets are different.

a) **Active sets are the same**, thus the optimal working set is probably found and it is possible to find the minimizer of (3) on the subspace defined by the currently active constraints, thus solve the problem

$$U^{(k)}_+ = \arg \min_{U^{(k)}} (1/2) U^{(k)T} H U^{(k)} + x^T F U^{(k)} \quad (12a)$$
$$\text{s.t. } U^{(k)} = U^{(k)}_T[c], \quad i \in \mathcal{W}^{(k)}. \quad (12b)$$

The minimizer of (12) can be written as $U^{(k)}_T = U^{(k)} + p^{(k)}$, where the search direction $p^{(k)}$ is the Newton step. Since the problem (3) has special form of constraints, the computation of the Newton step can be efficiently solved by the Null space method (NS) (see e.g. Nocedal and Wright [1999]), as follows: let $m \in \mathbb{N}$ be the number of active constraints at the current iteration and $r = n - m$. Then consider a matrix $G \in \mathbb{R}^{r \times r}$, $G > 0$, and a vector $g \in \mathbb{R}^r$ defined as follows [Bertsekas, 1976]

$$G = H_{\mathcal{I} \setminus \mathcal{W}^{(k)}} H_{\mathcal{I} \setminus \mathcal{W}^{(k)}}$$
$$g = \nabla f(U^{(k)})_{\mathcal{I} \setminus \mathcal{W}^{(k)}}, \quad (13)$$

where $\mathcal{I} = \{1, \ldots, n\}$ and $\nabla f(U^{(k)}) = H U^{(k)} + F^T x$. Then the Newton step is

$$p^{(k)} = \begin{cases} p^{(k)}_{\mathcal{I} \setminus \mathcal{W}^{(k)}} = - G^{-1} g \\ p^{(k)}_{\mathcal{W}^{(k)}} = 0. \end{cases} \quad (14)$$

**Remark 5.** Note, that (14) can be efficiently solved by Cholesky factorization with $O((1/3)r^3)$ and forward and backward substitution with $O(2r^2)$. 

5570
If the minimizer of (12) violates one or more currently inactive constraints, the working set cannot be optimal, and the Newton step (instead of gradient) can be used for projection onto the constraints (3b) as

\[ U_{C}^{(k+1)}[\alpha^{(k)}]_N = P(U^{(k)} + \alpha^{(k)}p^{(k)}) \]

(15)

to obtain the Cauchy point, the first local minimizer along the projected path by the exact line search procedure (see Nocedal and Wright [1999] for details).

On the other hand, if, the minimizer of (12) stays inside the subspace defined by \( W^{(k)} \), the optimizer of (3) was possibly found and algorithm continues to the next iteration, where the found point will probably satisfy the KKT conditions.

**b) Active sets are different**, thus the current working set is not optimal and the Cauchy point is searched using the exact line search along the negative gradient as it is described in Nocedal and Wright [1999].

**Algorithm 2** GP algorithm tailored for MPC

1: Repair feasibility of \( U^{(0)} \), set \( c \in (0, 1) \), \( k_{\text{max}} \in \mathbb{N} \).
2: for \( k = 0 \) to \( k_{\text{max}} \) do
3: if \( U^{(k)} \) satisfies the KKT conditions then
4: Stop with \( U^* = U^{(k)} \)
5: end if
6: Set \( A^{(k)} = \text{active set of } U_{T}^{(k)}[c] \).
7: if \( W^{(k)} = A^{(k)} \) then
8: Find the Newton step \( p^{(k)} \) on the subspace of currently active constraints by (14).
9: if \( U^{(k)} + p^{(k)} \) leads to constraints violation then
10: Starting from \( U^{(k)} \) project \( p^{(k)} \) to find the Cauchy point \( U_{C}^{(k)} \) by (15).
11: \( U^{(k+1)} = U_{C}^{(k)} \)
12: else
13: Set \( U^{(k+1)} = U^{(k)} + p^{(k)} \).
14: end if
15: end if
16: Starting from \( U^{(k)} \) project gradient to find the Cauchy point \( U_{C}^{(k)} \) by (8).
17: \( U^{(k+1)} = U_{C}^{(k)} \)
18: end if
19: end for

**Remark 6.** It should be noted, that Algorithm 2 does not involve the computation of Newton step with \( O(r^3) \) at each iteration unlike Algorithm 1.

**Remark 7.** The constant \( c \) can be seen as a tuning parameter of the algorithm, since the smaller value it has, the Newton step (14) will be computed more often and algorithm will converges in a smaller (but more expensive) number of iterations. It can be also interpreted as a variable which enables a smooth transition between Algorithm 1 (for \( c = 0 \)) and original GP algorithm presented in Goldstein [1964] (for \( c = \infty \)).

6. NUMERICAL EXPERIMENTS

In this section, Algorithm 2 is applied to randomly generated GP problems in form (3) and also for simulation of the control of the turbo diesel engine (see Ferreau et al. [2007] for description). All computational performance tests have been tested on standard PC with CPU E8500@3.16 GHz. The algorithm was implemented in standard C and compiled to the C-MEX file by Microsoft Visual C++ compiler with -O2 options. The C-MEX file was called from MATLAB 7.8.0 environment. The computation time was measured using the `QueryPerformanceCounter` function which gives a \( \mu \text{s} \) accuracy. All tests were performed with \( c = 0.01 \).

In randomly generated GP problems, double precision floating point implementation of Algorithm 2 was used and the dimension of the problem \( n \) was chosen from 1 to 200. For each \( n \), there were generated 7000 problems. Lower and upper bounds were generated so that the problems were feasible. The Hessians of the problems were generated as positive definite dense matrices. The warm starting has not been considered and the algorithm has started from the center of the feasible set.

The average number of iterations to solve the problem to the minimizer versus number of variables \( n \) and a number of active constraints in optimum is shown in Fig. 1. This shows, that in less then 60 iterations of the algorithm the minimizer was found in average.

From Fig. 2, it is clear, that the problems with \( n \leq 200 \) might be solved in less than 61ms (for \( n \leq 100 \) less than 7ms) on standard CPU. The performance of the algorithm was compared with the qpOASES, the online active set strategy also used in Ferreau et al. [2007]. In Fig. 3 it can be seen that qpOASES involves much more computational time than Algorithm 2 if many active constraints are in optimum. Thus, our approach is more preferable for MPC, where this is true very often.

The high fidelity local linear model of the turbo diesel engine was identified in neighborhood of the working point by linearization. An exhaust gas recirculation valve (EGR) and wastegate (WGT) were used as actuators. As in Ferreau et al. [2007] the mass air flow (MAF) and the intake manifold absolute pressure (MAP) were tracked to the references. The sampling period was \( T_s = 50 \text{ms} \), prediction horizon was chosen 7s and control horizon 0.15s (i.e. \( N_c = 3 \)). Thus, the resulting problem had dimension \( n = 6 \). During the simulations, the states of the engine were estimated by the Kalman filter which was as Algorithm 2 implemented in C-MEX file in single floating point implementation. It was chosen only 7 iterations for the early stopping of the algorithm and warm start from the previous solution was used. The performance of the resulting controller (see Fig. 4) was compared with the optimal...
have shown, that algorithm converges from the feasible point in less than 61ms on standard CPU for problems with dimension up to 200. And also that it is possible to use it for systems with short sampling times (less than 25µs). The optimal choice of constant c is still an open question and it is a goal of the future work.

REFERENCES


7. CONCLUSION

In this paper, we have presented algorithm for box constrained QP, that uses projection of the gradient and of the Newton step. It was shown, that this algorithm is very suitable for MPC where often many constraints are active in optimum, because the more constraints are active, the more is the decrease in computation time. Numerical tests