qpOASES - Online Active Set Strategy for Fast Linear MPC

Moritz Diehl, Hans Joachim Ferreau, Lieboud Vanden Broeck, Jan Swevers
Dept. ESAT and Center of Excellence for Optimization in Engineering OPTEC
K.U. Leuven, Belgium

joint work with Hans Georg Bock,
Interdisciplinary Center for Scientific Computing, Univ. Heidelberg, Germany
Linear MPC (or QP subproblems in NMPC)

For
- linear dynamic system
- linear constraints
- quadratic cost

only quadratic program (QP) needs to be solved:

\[
x_{k+1} = Ax_k + Bu_k,
\]

\[
\min_{u_0,\ldots,u_{N-1}} x_N^T P x_N + \sum_{i=0}^{N-1} (x_i^T Q x_i + u_i^T R u_i)
\]

s.t.
\[
x_{k+1} = Ax_k + Bu_k,
\]
\[
(x_0 \text{ given}),
\]
\[
c \leq C x_k \leq \overline{c},
\]
\[
d \leq D u_k \leq \overline{d},
\]
\[
c_T \leq C_T x_N,
\]
Eliminate states via “condensing“, obtain smaller scale *quadratic program (QP)* in variables \( w := (u_0^T, \ldots, u_{N-1})^T \)

\[
\begin{align*}
\text{QP}(x_0) : & \quad \min_w \frac{1}{2} w^T H w + w^T F^T x_0 \\
& \quad =: g(x_0) \\
\text{s.t.} \quad Gw & \geq b + E x_0, \\
& \quad =: b(x_0)
\end{align*}
\]

(assumption: \( H \) positive definite)

QP depends on \( x_0 \) via *affine functions* \( g(x_0) \) and \( b(x_0) \)
**Theorem**

Let $QP(x_0)$ be a strictly convex and feasible quadratic program. Then there exists a unique $w^* \in \mathbb{R}^n$ and at least one working set $\mathbb{A}$ and a vector $y^* \in \mathbb{R}^m$ which satisfy the following conditions:

\[
H w^* - G^T_{\mathbb{A}} y^*_{\mathbb{A}} = -g(x_0), \\
G_{\mathbb{A}} w^* = b_{\mathbb{A}}(x_0), \\
y^*_I = 0, \quad (I := \{1, \ldots, m\} \setminus \mathbb{A}), \\
G_I w^* \geq b_I(x_0), \\
y^*_A \geq 0.
\]
Define set of “feasible parameters“:

\[ \mathbb{P} := \{ x_0 \in \mathbb{R}^{n_x} \mid \text{QP}(x_0) \text{ is feasible} \} \]

Well known:

**THEOREM:** Set \( \mathbb{P} \) is convex, and can be partitioned into polyhedral „critical regions“ each corresponding to a different working set \( \mathbb{A} \). QP solution on each region is affine in \( x_0 \).
Sketch of Proof for Polyhedral Critical Regions

Check KKT conditions for fixed working set \( \mathbb{A} \):

1. \( g(x_0), b(x_0) \) affine: then \( w^*, y^* \) affine, because solution of linear system:

\[
H w^* - G_A^T y_A^* = -g(x_0), \\
G_A w^* = b_A(x_0), \\
y^*_\Pi = 0,
\]

2. \( w^*, y^* \) affine, therefore

\[
G_\Pi w^* \geq b_\Pi(x_0), \\
y_A^* \geq 0.
\]

are linear constraints on \( x_0 \) that define polyhedral „critical region“ in \( \mathbb{P} \).
Idea: Compute control on all critical regions in advance (Bemporad, Borrelli, Morari, 2002).

**Pro:** MPC in microsecs. possible

**Contra:** problem size limited

Example: 50 variables, lower and upper bounds:
$3^{50} = 10^{23}$ possible critical regions. Prohibitive.
Combine Explicit and Online MPC

- compute affine solution only on current critical region
- go on straight line from old to new problem data ( $\mathbb{P}$ convex! )
- **solve each QP on path exactly** (keep primal-dual feasibility)!
- need to change working set only at boundaries of critical regions
How to compute each step?

- determine change in $g(x_0)$ and $b(x_0)$, solve KKT system

\[
\begin{pmatrix}
H & G_A^T \\
G_A & 0
\end{pmatrix}
\begin{pmatrix}
\Delta w^* \\
-\Delta y^*_A
\end{pmatrix}
= \begin{pmatrix}
-\Delta g \\
\Delta b_A
\end{pmatrix}
\]

- choose steplength $\tau_{\text{max}}$ maximal such that

\[G_i^T (w^* + \tau \Delta w^*) \geq b_i(x_0) + \tau \Delta b_i\]

still holds for all inactive (primal) constraints, and

\[y^*_i + \tau \Delta y_i \geq 0\]

for all active dual variables: Set $\tau_{\text{max}} := \min \{1, \tau_{\text{prim}}^{\text{max}}, \tau_{\text{dual}}^{\text{max}}\}$

with $\tau_{\text{prim}}^{\text{max}} := \min_{\Delta w^* < \Delta b_i} \frac{b_i(x_0) - G_i^T w^*}{G_i^T \Delta w^* - \Delta b_i}$ and

$\tau_{\text{dual}}^{\text{max}} := \min_{\Delta y_i < 0} \frac{y^*_i}{\Delta y_i}$
How to change working set?

- add or remove constraints to/from working set when crossing borders of critical regions
- use null space approach, keep QR-factorization of active constraint matrix, and Cholesky factorization of projected hessian:

\[
G'_A = \begin{pmatrix} 0 & T \end{pmatrix} \begin{pmatrix} Z^T \\ Y^T \end{pmatrix},
\]

\[
Z^T H Z = R^T R,
\]

- each working set change costs only \(O(n^2)\) flops, exactly as one QP iteration in efficient QP solvers!
During homotopy, often redundant constraints become active and cause degeneracy.

Example of three active constraints in 2-D:
How to deal with degeneracy?

Without linear independence, KKT system becomes unsolvable! Addition of degenerate constraints must be avoided.

Remedy [Best 1996]: before adding extra row $G_j$ to solve auxiliary system

$$
\begin{pmatrix}
H & G_A^T \\
G_A & 0
\end{pmatrix}
\begin{pmatrix}
s \\
\xi
\end{pmatrix}
= 
\begin{pmatrix}
G_j \\
0
\end{pmatrix}
$$

if $s \equiv 0$, linear dependence is detected, and $\xi$ helps to find a constraint from $A$ that can be removed
Summary of qpOASES Algorithm

(1) Calculate $\Delta x_0$, $\Delta g$ and $\Delta b$

(2) Calculate primal and dual step directions $\Delta w^*$ and $\Delta y^*$

(3) Determine maximum homotopy step length $\tau_{\text{max}} := \min\{1, \tau_{\text{max}}^\text{prim}, \tau_{\text{max}}^\text{dual}\}$

(4) Obtain optimal solution of QP($\bar{x}_0$):
   (a) $\bar{x}_0 \leftarrow x_0 + \tau_{\text{max}} \Delta x_0$,
   (b) $\bar{w}^* \leftarrow w^* + \tau_{\text{max}} \Delta w^*$,
   (c) $\bar{y}^* \leftarrow y^* + \tau_{\text{max}} \Delta y^*$.

(5) if $\tau_{\text{max}} = 1$:
    Optimal solution of QP($x_{0}^{\text{new}}$) found.

(6) if $\tau_{\text{max}} = \tau_{\text{max}}^\text{dual}$:
    Remove a dual blocking constraint $j \left( \tau_{\text{max}}^\text{dual} = -\frac{y_j^*}{\Delta y_j} \right)$ from working set,
else if $\tau_{\text{max}} = \tau_{\text{max}}^\text{prim}$:
    Add a primal blocking constraint $j \left( \tau_{\text{max}}^\text{prim} = \frac{b_j(x_0) - G_j^T w^*}{G_j^T \Delta w^* - \Delta b_j} \right)$
    ensuring linear independence

(7) Set $x_0 \leftarrow \bar{x}_0$, $w^* \leftarrow \bar{w}^*$, $y^* \leftarrow \bar{y}^*$ and continue with step (1).
Limit number of active set changes per sampling time:

- lag behind, if too many changes necessary
- deliver solution of some problem between old and new
- make good for lag in later problems
P convex: QP on path infeasible \( \iff \) new QP infeasible

Stop at last feasible QP, wait for better posed problems
P convex: QP on path infeasible $\iff$ new QP infeasible

Stop at last feasible QP, wait for better posed problems

Fortunately: new QP feasible $\iff$ full path is feasible, and strategy works again
qpOASES: Open Code by Hans Joachim Ferreau

qpOASES: open source C++ code by Hans Joachim Ferreau

http://www.kuleuven.be/optec/software/qpOASES
Application to Chain of Masses

- 10 balls connected by springs, No. 1 fixed
- 3-D velocity of ball No. 10 controlled: \( \dot{x}_{N+1} = u(t) \)
- 2nd order ODE for other balls:
  \[
  \ddot{x}_i = \frac{1}{m} \left( F_{i,i+1} - F_{i-1,i} \right) + g, \quad i = 1, \ldots, N
  \]
- Force according to Hooke’s law strongly nonlinear:
  \[
  F_{i,i+1} = D \left( 1 - \frac{L}{\|x_{i+1} - x_i\|} \right) (x_{i+1} - x_i),
  \]
- Together: 57 nonlinear ODEs, chaotic system
After disturbance, chain crashes into wall
MPC controller shall avoid crashing into wall

- linearize system at steady state
- choose 200 ms sampling time
- predict 80 samples: \(3 \times 80 = 240\) degrees of freedom
- bounds (up/lo): \(2 \times 240 = 480\)
- state constraints that avoid hitting the wall: \(9 \times 80 = 720\)

Note: large QP with \(~1\) MB data
MPC respects bounds and state constraint
Compare four QP strategies

- Standard solver (QPSOL), cold start
- QPSOL, warm start
- Online Active Set Strategy (qpOASES), full convergence
- qpOASES, real-time variant with at most 10 QP Iterations

Note: Explicit MPC cannot be applied due to problem size
Number of QP Iterations (Working Set Changes)
Number of QP Iterations (Working Set Changes)
Number of QP Iterations (Working Set Changes)
Number of QP Iterations (Working Set Changes)

- QPSOL (cold)
- QPSOL (warm)
- OASES (full)
- OASES (real)

![Graph showing iterations over time for different algorithms.](image-url)
Performance of Online Active Set Strategy

- Number of QP working set changes 3-5 times lower than for QPSOL with warm starts
- Can limit maximum number without much suboptimality

Additionally:
- QPSOL often even needs Phase1 LP Iterations
- qpOASES needs no new matrix factorizations
- CPU times compare even more favourably...
CPU Time Comparison: qpOASES Factor 10 Faster
Time Optimal MPC: a 100 Hz Application

- Quarter car: oscillating spring damper system
- MPC Aim: settle at any new setpoint in *in minimal time*
- Two level algorithm: MIQP
  - 6 online data
  - 40 variables + one integer
  - 242 constraints (in-&output)
- use qpOASES on dSPACE
- CPU time: <10 ms

Lieboud Van den Broeck in front of quarter car experiment
Setpoint change without control: oscillations
With LQR control: inequalities violated
With Time Optimal MPC
Time Optimal MPC: a 60 Hz Application

- Overhead crane
- MPC Aim: settle at any new setpoint in \textit{in minimal time}
- Two level algorithm: MIQP
  - 6 online data
  - 40 variables + one integer
  - 242 constraints (in-& output)
- use qpOASES on dSPACE
- CPU time: <10 ms

\textit{Lieboud Van den Broeck}
qpOASES: open code, but direct industrial funding

Hoerbiger: MPC of Large Bore Gas Engines for US Pipeline Compressors

IPCOS: Efficient QP for Process Control
qpOASES running on Industrial Control Hardware (20 ms)

Project manager (Dec. 2008): “...we had NO problem at all with the qpOASES code. Your Software has throughout the whole project shown reliable and robust performance.”
Conclusions

- Linear and Nonlinear MPC need reliable QP solution
- Explicit MPC prohibitive for nontrivial problem dimensions
- Online Active Set Strategy (qpOASES) is one order of magnitude faster than conventional QP with warmstarts
- Linear MPC in kHz range realizable even for larger QPs
- Time Optimal MPC interesting alternative to tracking