A Condensing Algorithm for Nonlinear MPC with a Quadratic Runtime in Horizon Length ⋆

Joel A. E. Andersson, Janick V. Frasch, Milan Vukov, Moritz Diehl

Electrical Engineering Department (ESAT) and Optimization in Engineering Center (OPTEC), KU Leuven, Belgium

Abstract

A large number of practical algorithms for Optimal Control Problems (OCP) relies on a so-called *condensing* procedure to exploit the given structure in the quadratic programming (QP) subproblems. While the established structure-exploiting condensing algorithm is of cubic complexity in the horizon length, in this technical note we propose a novel algorithm that is only of quadratic complexity in horizon length. We present numerical results confirming that the proposed algorithm is faster in practice and discuss implications for related QP solution algorithms that rely on the elimination of the state sequence.

Key words:
Optimal control, model-predictive control, structured quadratic programming, condensing algorithm, fast-gradient methods

1 Introduction

In this paper we consider the solution of the following linear time-varying optimal control problem (LTV-OCP) with quadratic objective, in n_x states and n_u controls on a time horizon of length N:

$$\begin{aligned}
\underset{\substack{u_0,\ldots,u_{N-1}\\ x_1,\ldots,x_N}}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \left(\frac{1}{2} u_k^\intercal R_k u_k + x_k^\intercal S_k u_k + u_k^\intercal r_k \right) \\
& + \sum_{k=1}^{N} \left(\frac{1}{2} x_k^\intercal Q_k x_k + x_k^\intercal q_k \right) && \text{(1a)} \\
\text{subject to} \quad & x_{k+1} = A_k x_k + B_k u_k + c_k, \\
& \qquad k = 0, \ldots, N-1, && \text{(1b)} \\
& \underline{u}_k \leq u_k \leq \overline{u}_k, \quad k = 0, \ldots, N-1, && \text{(1c)} \\
& \underline{x}_k \leq x_k \leq \overline{x}_k, \quad k = 1, \ldots, N. && \text{(1d)}
\end{aligned}$$

For the ease of presentation, we assume a fixed initial value x_0 and only bound constraints, while the algorithms presented in this paper can be extended in a straightforward way to more general LTV-OCPs. Such LTV-OCP with quadratic objective are a structured form of quadratic programs (QP) and originate from a wide class of problem formulations in optimal control, notably in solution algorithms for linear-quadratic model predictive control (MPC) problems [9], linear moving horizon estimation (MHE) problems [8], as well as their nonlinear counterparts and general nonlinear optimal control problems (OCPs) and dynamic parameter estimation problems solved by sequential quadratic programming (SQP) based algorithms [1, 3, 4].

Interior-point (IP) algorithms can exploit the block-banded structure of LTV QPs well internally [9], but have well-known difficulties in exploiting the similarity between subsequently solved QPs in an SQP or online solution context. Active-set (AS) methods on the other hand can exploit the similarity between subsequently solved QPs very well, but typically do not benefit from the known problem structure to the same extent as IP methods. One way to exploit structure in AS methods is to make use of a so-called *condensing* preprocessing step [3] to eliminate the over-parameterized state sequence from the QP optimization variables. This can lead to drastically reduced problem sizes if the number of states is relatively large compared to the number of controls, which is the case for large classes of control and estimation problems.

Recent comparisons indicate that efficient condensing-based AS methods outperform efficient IP methods on time horizons of short and medium length, but are being outperformed on longer time horizons [10]. This is largely due to the expensive condensing step, which needs to be performed in every iteration for LTV systems originating from the linearization of nonlinear MPC problems. Up to now, the condensing step was believed to be of cubic complexity in the horizon length of the optimization problem [2, 6, 7], as described in Section 2.

⋆ Corresponding author: Joel Andersson, joel.andersson@esat.kuleuven.be

The contribution of this paper is a novel algorithm that is equivalent to the established condensing algorithm but with quadratic rather than cubic complexity in the horizon length.

2 QP condensing

The constraints (1b) can be written in matrix form:

$$A\,x = B\,u + c \Leftrightarrow \begin{bmatrix} I & & & \\ -A_1 & I & & \\ & \ddots & \ddots & \\ & & -A_{N-1} & I \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} B_0 & & & \\ & B_1 & & \\ & & \ddots & \\ & & & B_{N-1} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} A_0\,x_0 + c_0 \\ c_1 \\ \vdots \\ c_{N-1} \end{bmatrix}, \quad (2)$$

giving the following equivalent form of (1):

$$\begin{aligned} \underset{x,u}{\text{minimize}} \quad & \frac{1}{2}\,u^\intercal R\,u + x^\intercal S\,u + u^\intercal r + \frac{1}{2}\,x^\intercal Q\,x + x^\intercal q \\ \text{subject to} \quad & A\,x = B\,u + c \\ & \underline{u} \le u \le \overline{u}, \quad \underline{x} \le x \le \overline{x}, \end{aligned} \quad (3)$$

where, in addition to the quantities in (2), we define the vectors $r := [r_0^\intercal, \ldots, r_{N-1}^\intercal]^\intercal$, $q := [q_1^\intercal, \ldots, q_N^\intercal]^\intercal$, $\underline{x} := [\underline{x}_1^\intercal, \ldots, \underline{x}_N^\intercal]^\intercal$, $\overline{x} := [\overline{x}_1^\intercal, \ldots, \overline{x}_N^\intercal]^\intercal$ as well as the matrices $R := \operatorname{diag}(R_0, \ldots, R_{N-1})$, $S := \operatorname{diag}(S_0, \ldots, S_{N-1})$ and $Q := \operatorname{diag}(Q_0, \ldots, Q_N)$.

Rather than solving the full space QP (1) directly, we seek to solve an equivalent so-called *condensed* QP:

$$\begin{aligned} \underset{u}{\text{minimize}} \quad & \frac{1}{2}\,u^\intercal\,H\,u + u^\intercal\,h \\ \text{subject to} \quad & \underline{u} \le u \le \overline{u}, \quad \underline{x} - g \le G\,u \le \overline{x} - g, \end{aligned} \quad (4)$$

which has $n_u\,N$ instead of $(n_u + n_x)\,N$ free variables.

Since A is nonsingular, we can use (2) to explicitly eliminate x from (3), giving the following expressions for g, G, h and H in (4):

$$g := A^{-1}\,c, \quad (5a)$$
$$G := A^{-1}\,B, \quad (5b)$$
$$h := r + G^\intercal\,(q + Q\,g) + S^\intercal\,g, \quad (5c)$$
$$H := R + G^\intercal\,Q\,G + S^\intercal\,G + G^\intercal\,S, \quad (5d)$$

where $g =: [g_1^\intercal, \ldots, g_N^\intercal]^\intercal$, $h =: [h_0^\intercal, \ldots, h_{N-1}^\intercal]^\intercal$,

$$\underbrace{\begin{bmatrix} G_{1,0} & & \\ \vdots & \ddots & \\ G_{N,0} & \cdots & G_{N,N-1} \end{bmatrix}}_{=:G} \text{ and } \underbrace{\begin{bmatrix} H_{0,0} & \cdots & H_{0,N-1} \\ \vdots & \ddots & \vdots \\ H_{N-1,0} & \cdots & H_{N-1,N-1} \end{bmatrix}}_{=:H}.$$

For future reference, we define the Lagrangian of (1),

$$\begin{aligned} \mathcal{L}(u, x, \mu, \lambda, \nu) := & \sum_{k=0}^{N-1} \left(\frac{1}{2}\,u_k^\intercal\,R_k\,u_k + x_k^\intercal\,S_k\,u_k + u_k^\intercal\,r_k \right) \\ & + \sum_{k=1}^{N} \left(\frac{1}{2}\,x_k^\intercal\,Q_k\,x_k + x_k^\intercal\,q_k \right) + \sum_{k=0}^{N-1} \mu_k^\intercal u_k + \sum_{k=1}^{N} \lambda_k^\intercal x_k \\ & + \sum_{k=0}^{N-1} \nu_{k+1}^\intercal\,(A_k\,x_k + B_k\,u_k + c_k - x_{k+1}), \end{aligned} \quad (6)$$

introducing the multipliers ν, μ, and λ for the constraints (1b), (1c) and (1d). For μ and λ, we use the same multipliers for both the upper and lower variable bounds.

3 The classical condensing algorithm

The classical algorithm for a structure-exploiting computation of (5) can be derived as follows, cf. [**?** 6, 7].

Since A in (2) lower triangular, g can be calculated from (5a) using a blockwise forward substitution:

Algorithm 1. Calculating g from (5a)
1: $g_0 \leftarrow x_0$
2: **for** $k = 0, \ldots, N-1$ **do**
3: $\quad g_{k+1} \leftarrow A_k\,g_k + c_k$
4: **end for**

Similarly, G from (5b) is computed using blockwise forward substitutions exploiting the sparsity structure:

Algorithm 2. Calculating G from (5b)
1: **for** $i = 0, \ldots, N-1$ **do**
2: $\quad G_{i+1,i} \leftarrow B_i$
3: $\quad$ **for** $k = i+1, \ldots, N-1$ **do**
4: $\quad\quad G_{k+1,i} \leftarrow A_k\,G_{k,i}$
5: $\quad$ **end for**
6: **end for**

Subsequently, h is obtained by performing the matrix-vector multiplications given in (5c):

Algorithm 3. Calculating h from (5c), $\mathcal{O}(N^2)$ complexity
1: **for** $i = 0, \ldots, N-1$ **do**
2: $\quad h_i \leftarrow r_i + S_i^\intercal\,g_{i+1}$
3: $\quad$ **for** $k = i+1, \ldots, N$ **do**
4: $\quad\quad h_i \leftarrow h_i + G_{k,i}^\intercal\,(q_k + Q_k\,g_k)$
5: $\quad$ **end for**
6: **end for**

Finally, H is calculated by carrying out the blockwise matrix operations in (5d), exploiting the special structure of R, Q, S and G. Due to symmetry of H, only the upper triangular part needs to be calculated. To limit the floating point operations, we precalculate $W =: Q\,G$:

Algorithm 4. Calculating H from (5d), $\mathcal{O}(N^3)$ complexity

for $i = 0, \dots, N-1$ **do**
  for $k = 1, \dots, N$ **do**
    $W_{i,k} \leftarrow Q_k\, G_{k,j}$
  end for
  for $j = 0, \dots, i-1$ **do**
    $H_{i,j} \leftarrow S_i^\intercal\, G_{i+1,j}$
    for $k = i+1, \dots, N$ **do**
      $H_{i,j} \leftarrow H_{i,j} + G_{k,i}^\intercal\, W_{k,j}$
    end for
  end for
  $H_{i,i} \leftarrow R_i$
  for $k = i+1, \dots, N$ **do**
    $H_{i,i} \leftarrow H_{i,i} + G_{k,i}^\intercal\, W_{k,i}$
  end for
end for

Algorithms 1, 2, 3 and 4 together allow to formulate the condensed QP (4), which can be solved using a dense QP solver yielding the primal solution u and multipliers for the inequality constraints λ.

To recover x and ν, Constraints (1b) can be used in combination with $\nabla_{x_k}\mathcal{L} = 0$, $k = N, \dots, 1$, which are necessary conditions of optimality:

Algorithm 5. Recovering x and ν in (1)

for $k = 0, \dots, N-1$ **do**
  $x_{k+1} \leftarrow A_k\, x_k + B_k\, u_k + c_k$
end for
$\nu_N \leftarrow \lambda_N + Q_N\, x_N$
for $k = N-1, \dots, 1$ **do**
  $\nu_k \leftarrow \lambda_k + Q_k\, x_k + A_k^\intercal\, \nu_{k+1} + S_k\, u_k + q_k$
end for

Complexity analysis

The computational complexity of the established condensing algorithm can be assessed by counting the number of floating point operations (FLOPs) in the corresponding algorithms. We assume *multiply-accumulate* (MAC) to be a single cycle operation. A summary of the complexity can be found in Table 1. The entries of the table are consistent with the complexity observed, e.g., by Kirches et al. [6]. When special structures occur, like Q diagonal or S zero, the figures of the table may become lower.

4 A condensing algorithm with N^2 complexity

An alternative algorithm for h can be derived by exploiting the special structure of G. From (5):

$$\begin{aligned} h &= r + G^\intercal\,(q + Q\,g) + S^\intercal\, g \\ &= r + B^\intercal\, A^{-\intercal}\,(q + Q\,g) + S^\intercal\, g, \end{aligned} \tag{7}$$

Table 1
Complexity analysis, $\mathcal{O}(N^3)$ condensing

Step	Cost [FLOPs], leading term(s)
G (Algorithm 2)	$\frac{1}{2}\,N^2\,n_x^2\,n_u$
H (Algorithm 4)	$\frac{1}{6}\,N^3\,n_x\,n_u^2 + N^2\,n_x^2\,n_u$
g (Algorithm 1)	$N\,n_x^2$
h (Algorithm 3)	$\frac{1}{2}\,N^2\,n_x^2\,n_u + \frac{1}{2}\,N^2\,n_x\,n_u^2$
u, λ (Solving QP)	$\frac{1}{3}\,N^3\,n_u^3$
s,ν (Algorithm 5)	$3\,N\,n_x^2 + 2\,N\,n_x\,n_u$

which can be calculated cheaply using matrix-vector multiplications and a backward substitution for $A^\intercal$:

Algorithm 6. Calculating h from (5c), $\mathcal{O}(N)$ complexity

$w_N \leftarrow q_N + Q_N\, g_N$
for $k = N-1, \dots, 1$ **do**
  $h_k \leftarrow r_k + S_k^\intercal\, g_k + B_k^\intercal\, w_{k+1}$
  $w_k \leftarrow q_k + Q_k\, g_k + A_k^\intercal\, w_{k+1}$
end for
$h_0 \leftarrow r_0 + S_0^\intercal\, x_0 + B_0^\intercal\, w_1$

Similarly, from (5d), we can write H as:

$$\begin{aligned} H &= R + G^\intercal\, Q\, G + S^\intercal\, G + G^\intercal\, S \\ &= R + B^\intercal\, A^{-\intercal}\,(Q\,G) + S^\intercal\, G + (S^\intercal\, G)^\intercal, \end{aligned} \tag{8}$$

which using blockwise backward substitution to calculate $W := A^{-\intercal}\,(Q\,G)$ can be calculated as follows:

Algorithm 7. Calculating H from (5d), $\mathcal{O}(N^2)$ complexity

for $i = 0, \dots, N-1$ **do**
  $W_{N,i} \leftarrow Q_N\, G_{N,i}$
  for $k = N-1, \dots, i+1$ **do**
    $H_{k,i} \leftarrow S_k^\intercal\, G_{k,i} + B_k^\intercal\, W_{k+1,i}$
    $W_{k,i} \leftarrow Q_k\, G_{k,i} + A_k^\intercal\, W_{k+1,i}$
  end for
  $H_{i,i} \leftarrow R_i + B_i^\intercal\, W_{i+1,i}$
end for

The equivalence between Algorithms 3 and 6 and the equivalence between Algorithms 4 and 7 are given from the manner in which the algorithms were derived.

Complexity analysis

For the proposed algorithm, the complexity in terms of floating point operations, again to the highest order term, is shown in Table 2. We see that the complexity decreases from cubic to quadratic in the calculation of H and from quadratic to linear in the calculation of h. The complexities in n_x and n_u are unchanged.

5 Numerical results

To assess the proposed condensing algorithm, we consider the benchmark from nonlinear MPC presented in [11]. A chain of masses connected by springs with one end

Table 2
Complexity analysis, $\mathcal{O}(N^2)$ condensing

Step	Cost [FLOPs], leading terms
H (Algorithm 7)	$N^2\,n_x^2\,n_u + N^2\,n_x\,n_u^2$
h (Algorithm 6)	$2\,N\,n_x^2 + 2\,N\,n_x\,n_u$

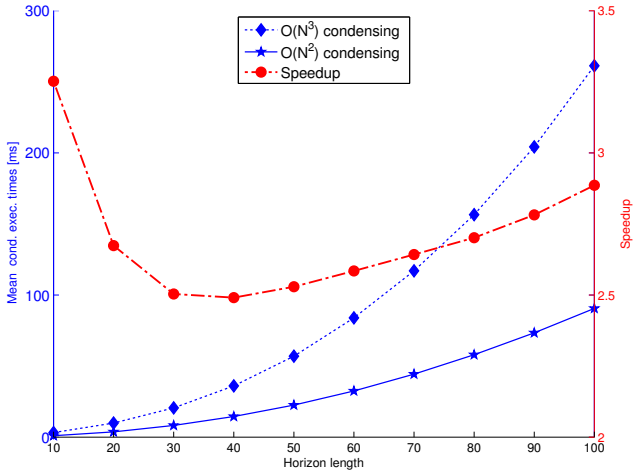

Fig. 1. Execution times vs. horizon length for benchmark [11]

fixed are steered to an equilibrium position with a wall constituting a hard constraint on the state trajectory. With 9 masses, we obtain a system with $n_x = 57$ states and $n_u = 3$ controls. We present the execution time of Algorithms 1, 2, 5, 6 and 7 for control horizons up to 100 in Figure 1. The timings are compared against the standard $\mathcal{O}(N^3)$ condensing algorithm, i.e. Algorithm 1, 2, 3, 4 and 5. We have also included the runtime ratio between the old and new algorithms. Both algorithms have been implemented in ANSI-C as part of the ACADO Code Generation tool [5]. The results, which have been produced on a 3GHz Intel Q9650 based desktop computer running Linux, confirm significant savings of 60 % or more (i.e. a speedup of 2.5 or greater) for all considered horizon lengths.

6 Conclusions and Outlook

For the benchmark considered, the new condensing algorithm proposed in this paper outperformed the classical one for all considered control horizons with a factor of at least 2.5. This suggests that the proposed condensing algorithm is not only superior for long control horizons, but for short to medium ones as well. We see that the curve for the speed-up factor in Figure 1 becomes asymptotically linear for growing horizons, confirming the one order of magnitude improvement.

When regarding the complete algorithm to solve the full space QP, there is one remaining step that is of cubic complexity: the solution of the condensed QP. This cost arises from the Cholesky factorization of the Hessian before the first active set change, while all active set changes have a cost of $\mathcal{O}(N^2\,n_u^2)$. Though this cost is independent of the state dimension n_x and in most practical instances still lower than the new condensing algorithm of quadratic complexity, it would be desirable to find an active algorithm for the condensed QP with quadratic complexity, which is an interesting topic for future research.

Acknowledgements

This work was supported by the Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC, GOA/10/09 MaNet and GOA/10/11 Global real- time optimal control of autonomous robots and mechatronic systems. Flemish Government: IOF / KP / SCORES4CHEM, FWO: PhD/postdoc grants and projects: G.0320.08 (convex MPC), G.0377.09 (Mechatronics MPC); IWT: PhD Grants, projects: SBO LeCoPro; Belgian Federal Science Policy Office: IUAP P7 (DYSCO, Dynamical systems, control and optimization, 2012-2017); EU: FP7- EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), Eurostars SMART, ACCM.

References

[1] L.T. Biegler. Advances in nonlinear programming concepts for process control. *J. Process Control*, 8(5):301–311, 1998.

[2] L.T. Biegler, J. Nocedal, and C. Schmid. A reduced Hessian method for large-scale constrained optimization. *SIAM J. Optimization*, 5:314–347, 1995.

[3] H.G. Bock. Recent advances in parameter identification techniques for ODE. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*. Birkhäuser, Boston, 1983.

[4] M. Diehl et al. Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *J. Process Control*, 12(4):577–585, 2002.

[5] B. Houska, H.J. Ferreau, and M. Diehl. An Auto-Generated Real-Time Iteration Algorithm for Nonlinear MPC in the Microsecond Range. *Automatica*, 47(10):2279–2285, 2011.

[6] C. Kirches et al. Efficient Direct Multiple Shooting for Nonlinear Model Predictive Control on Long Horizons. *J. Process Control*, 22(3):540–550, 2012.

[7] D.B. Leineweber. *Efficient Reduced SQP Methods for the Optimization of Chemical Processes Described by Large Sparse DAE Models*, volume 613 of *Fortschritt-Berichte VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 1999.

[8] C.V. Rao, J.B. Rawlings, and J.H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(2):1619–1628, 2001.

[9] C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of Interior-Point Methods to Model Predictive Control. *J. Optimization Theory and Applications*, 99:723–757, 1998.

[10] Vukov et al. Auto-generated Algorithms for Nonlinear Model Predicitive Control on Long and on Short Horizons. In *Proc. 52nd Conference on Decision and Control (CDC)*, 2013.

[11] L. Wirsching, H. G. Bock, and M. Diehl. Fast NMPC of a chain of masses connected by springs. In *Proc. IEEE International Conference on Control Applications, Munich*, pages 591–596, 2006.