A new quadratic programming strategy for efficient sparsity exploitation in SQP-based nonlinear MPC and MHE ⋆

Janick V. Frasch [*,**] **Milan Vukov** [*] **Hans Joachim Ferreau** [***] **Moritz Diehl** [*]

[*] *Department of Electrical Engineering, KU Leuven, Belgium (e-mail: janick.frasch@esat.kuleuven.be).*
[**] *Department of Mathematics, Otto-von-Guericke University Magdeburg, Germany*
[***] *ABB Corporate Research, Baden-Dättwil, Switzerland*

Abstract: A large class of algorithms for nonlinear model predictive control (MPC) and moving horizon estimation (MHE) is based on sequential quadratic programming and thus requires the solution of a sparse structured quadratic program (QP) at each sampling time. We propose a novel algorithm based on a dual two-level approach involving a nonsmooth version of Newton's method that aims at combining sparsity exploitation features of an interior point method with warm-starting capabilities of an active-set method. We address algorithmic details and present the open-source implementation qpDUNES. The effectiveness of the solver in combination with the ACADO Code Generation tool for nonlinear MPC is demonstrated based on set of benchmark problems, showing significant performance increases compared to the established condensing-based approach, particularly for problems with long prediction horizons.

Keywords: Predictive Control, Optimization algorithms, Open-Source Software

1. INTRODUCTION

Model predictive control (MPC) is an approach to obtain a feedback control law taking physical process models and problem-inherent constraints into account, cf. Rawlings and Mayne (2009). It relies on the online solution of an optimization problem in each feedback-generating iteration. For high predictive accuracy and good stability properties it is desirable to design MPC controllers that can deal with detailed process descriptions to reduce the prediction-inherent uncertainty, resulting in big dynamic systems and long prediction horizons. Particularly the long prediction horizons, however, can render the underlying optimization problem very challenging for a real-time implementation.

Most nonlinear MPC algorithms require the direct solution of an (in general) *nonlinear programming problem* (NLP) online. Approaches like the advanced step controller, see Zavala and Biegler (2009), on the one hand solve the full NLP at each sampling instant using interior point methods. Approaches like the real-time iteration (RTI) scheme on the other hand, see Diehl et al. (2002), are based on a sequential quadratic programming (SQP) algorithm and therefore need to solve a quadratic programming problem (QP) in each iteration. Particularly in combination with automatic code generation, the RTI scheme can lead to very short computation times while retaining sufficient accuracy (see, e.g., Houska et al. (2011); Ferreau et al. (2012b); Vukov et al. (2012)) and therefore seems to be a well suited approach for embedded nonlinear MPC of high-speed applications. Due to the structural equivalence, *moving horizon estimation* (MHE) can be cast into the same framework and can be solved efficiently using the RTI scheme, see Kühl et al. (2011). In the following, we subsume both problem classes, MHE and MPC, under the term MPC for clarity of the presentation.

A crucial step for long prediction horizon problems is the efficient solution of the *highly structured QPs* in each iteration, which is usally performed by tailored algorithms based on interior point methods (e.g., Mattingley and Boyd (2009); Domahidi et al. (2012)), active-set methods (e.g., Ferreau et al. (2008)), or fast-gradient methods (e.g., Richter et al. (2009); Bemporad and Patrinos (2012)). Both classes of methods, interior-point and fast-gradient methods on the one hand, and active-set methods on the other hand, have significant shortcomings in the context of MPC and MHE. While the former ones generally cannot exploit the knowledge about similarity between the solutions of two subsequent QPs in the MPC context, the latter ones typically require a so-called condensing routine to benefit from the problem-inherent sparsity, cf. Leineweber (1999). Even though recent results from Andersson et al. (2013) lead to an improved performance of

⋆ This research was supported by 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.

Preprint submitted to 19th IFAC World Congress. Received November 15, 2013.

the condensing step, its overall runtime complexity is still quadratic in the horizon length and cannot be expected to be accelerated further. The initial factorization of the condensed QP Hessian is even of cubic runtime complexity in the horizon length, cf. Kirches et al. (2012).

In this paper, we present a new idea for solving strictly convex quadratic subproblems in the context of nonlinear MPC and MHE. It is based on ideas for a band-structured QP solver that were introduced in Ferreau et al. (2012a) and extended in Frasch et al. (2013). Based on original ideas from Li and Swetits (1997) and Dai and Fletcher (2006) the stage coupling constraints connecting the MPC problem over the prediction horizon are dualized and the resulting QP is solved in a two level approach, using a non-smooth Newton method in the multipliers of the stage coupling constraints on the higher level, and a primal active-set method in the decoupled parametric QPs of each stage on the lower level. The advantage of this so-called *dual Newton strategy* is that it combines structure exploitation capabilities of interior point methods with the warm-starting capabilities of active set methods; in particular it comes with only a linear runtime complexity in the horizon length. Note that in contrast to classical active-set methods this approach permits several active-set changes in each Newton-type iteration. Still, the resulting algorithm has the flavor of an active-set method in the sense that the exact optimal solution is obtained.

In this paper, we address the core ideas of the dual Newton strategy and show how it can be applied efficiently for nonlinear MPC and MHE. Most importantly, we present qpDUNES, an open-source implementation of the *DU*al *NE*wton *S*tategy that is now available as structure-exploiting QP solver in the open-source ACADO toolkit (see Houska et al. (2011)) for nonlinear MPC and MHE of high-speed applications. We demonstrate the effectiveness of qpDUNES in comparison against a code-generated condesing/qpOASES approach first described in Houska et al. (2011); Ferreau et al. (2008) based on a selection of challenging nonlinear MPC benchmark problems.

2. QP SOLUTION METHOD

2.1 Quadratic subproblem description

We assume familiarity of the reader with the RTI scheme from Diehl et al. (2002). Within the RTI scheme we repeatedly need solve the following subproblem, that can be interpreted as a linear MPC problem. Here, we group the optimization variables, state increments $\Delta x_k \in \mathbb{R}^{n_x}$ and control increments $\Delta u_k \in \mathbb{R}^{n_u}$, in stage variables $z_k = \begin{bmatrix} \Delta x_k^\top & \Delta u_k^\top \end{bmatrix}^\top \in \mathbb{R}^{n_z}$ for each stage $k = 0, \dots, N-1$, and $z_N = [\Delta x_N \ 0]$ for the terminal stage. Throughout this paper we are consequently interested in repeatedly solving the following problem in an efficient manner:

$$\min_z \ \sum_{k=0}^{N} \left(\frac{1}{2} z_k^\top H_k z_k + g_k^\top z_k \right) \tag{1a}$$

$$\text{s.t. } E_{k+1} z_{k+1} = C_k z_k + c_k \qquad \forall\, k = 0, \dots, N-1 \tag{1b}$$

$$\underline{d}_k \le D_k z_k \le \overline{d}_k \qquad \forall\, k = 0, \dots, N. \tag{1c}$$

We assume positive definite second-order terms $0 \prec H_k \in \mathbb{R}^{n_z \times n_z}$ and $g_k \in \mathbb{R}^{n_z}$ for each $k \in \mathcal{S} := \{0, \dots, N\}$ throughout the paper. Two subsequent stages $k \in \mathcal{S}$ and $k+1 \in \mathcal{S}$ are coupled by first-order terms $C_k, E_{k+1} \in \mathbb{R}^{n_x \times n_z}$ and a constant term c_k. We assume that all C_k have full row rank and that all E_k have the special structure $E_k = [I \ 0]$, where $I \in \mathbb{R}^{n_x \times n_x}$ is an identity matrix and 0 is a zero matrix of appropriate dimensions. Vectors $\underline{d}_k, \overline{d}_k \in \mathbb{R}^{n_d}$, and a matrix $D_k \in \mathbb{R}^{n_d \times n_z}$ of full row rank denote stage constraints.

2.2 Dual decomposition

The main idea of our proposed QP solution algorithm is to decouple the QP stages by dualizing constraints (1b). Introducing $\lambda := \begin{bmatrix} \lambda_1^\top & \lambda_2^\top & \cdots & \lambda_N^\top \end{bmatrix}^\top \in \mathbb{R}^{N n_x}$ we can express (1a) and (1b) by the partial Lagrangian function

$$\begin{aligned} \mathcal{L}(z, \lambda) = \sum_{k=0}^{N} \Bigg(& \frac{1}{2} z_k^\top H_k z_k + g_k^\top z_k \\ & + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} z_k + \lambda_{k+1}^\top c_k \Bigg) \\ =: \sum_{k=0}^{N} & L_k(z_k, \lambda_k, \lambda_{k+1}), \end{aligned}$$

where we define zero matrices $E_0 = C_N = 0 \in \mathbb{R}^{n_x \times n_z}$ and redundant multipliers $\lambda_0 = \lambda_{N+1} := 0 \in \mathbb{R}^{n_x}$ only for notational convenience.

By elementary Lagrangian duality theory the primal QP (1) is equivalent to

$$\begin{aligned} \max_\lambda \ \min_z \ & \sum_{k=0}^{N} L_k(z_k, \lambda_k, \lambda_{k+1}) \\ \text{s.t. } & \underline{d}_k \le D_k z_k \le \overline{d}_k \qquad \forall k = 0, \dots, N \end{aligned}$$

Observe that this Problem is separable in the stage variables z_k. Problem (1) can thus equivalently be written as

$$\max_\lambda f^*(\lambda) := \max_\lambda \sum_{k=0}^{N} f_k^*(\lambda), \tag{2}$$

where

$$\begin{aligned} f_k^*(\lambda) \ := \ & \min_{z_k} \ \frac{1}{2} z_k^\top H_k z_k + m_k(\lambda)^\top z_k + \lambda_{k+1}^\top c_k \\ & \text{s.t.} \ \ \underline{d}_k \le D_k z_k \ \le \ \overline{d}_k \end{aligned} \tag{QP$_k$}$$

with $m_k(\lambda) := g_k^\top + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix}$. From this definition of the dual function f^* we can particularly see that $z(\lambda) := [z_0(\lambda), \dots, z_N(\lambda)]$ is optimal for each choice of λ.

2.3 Non-smooth Newton approach

Under the assumption that a feasible solution for (1) exists, it was shown in Ferreau et al. (2012a); Frasch et al. (2013) (based on results from Fiacco (1983); Zafiriou (1990); Berkelaar et al. (1997)) that $f^*(\lambda)$ exists on $\mathbb{R}^{N n_x}$ and further is a concave, piecewise quadratic, and once continuously differentiable function. We solve the unconstrained piecewise-quadratic program (2) by employing a non-smooth Newton method, as originally proposed in Li and Swetits (1997). The second derivative $\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda^i)$ is

unique everywhere but on a null set (the "seams" of f^*), where in general a jump occurs. Observe that $\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda^i)$ is not trivial, since all z_k implicitly depend on λ. The proposed QP solution method is given by Algorithm 1, cf. Frasch et al. (2013), where we denote the Lagrange multipliers of constraints (1c) by $\mu_k \in \mathbb{R}^{2n_d}$ for each $k \in \mathcal{S}$.

Algorithm 1: Dual Newton Strategy

Input: Initial guess λ^0, termination criteria n_{maxIt}, ϵ_λ
Output: Optimal solution (z^*, λ^*, μ^*)

1 $i := 0$
2 **while** $i < n_{\text{maxIt}}$ **do**
3 Solve all $\text{QP}_k(\lambda^i)$ to obtain $[z_k^*(\lambda^i), \mu_k^*(\lambda^i)]$
4 Compute $\left[-\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda^i)\right]$ and $\left[\frac{\partial f^*}{\partial \lambda}(\lambda^i)\right]$ analytically
5 **if** $\left\|\frac{\partial f^*}{\partial \lambda}(\lambda^i)\right\| \leq \epsilon_\lambda$ **then**
6 **return** $[z_k^*(\lambda^i), \lambda^i, \mu_k^*(\lambda^i)]$
7 Solve Newton system $\left[-\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda^i)\right] \Delta\lambda = \left[\frac{\partial f^*}{\partial \lambda}(\lambda^i)\right]$
8 Compute step size α
9 Update the current iterate $\lambda^{i+1} := \lambda^i + \alpha\Delta\lambda$
10 i := i+1

It has been shown in Frasch et al. (2013) that this algorithm converges to the (unique) optimal dual solution λ^*, that is characterized by stationary of $f^*(\lambda)$ and implicitly defines $z^* := z(\lambda^*)$ via (QP_k).

3. ALGORITHMIC DETAILS OF THE DUAL NEWTON STRATEGY

Due to its temporal coupling, Problem (2) possesses a specific structure that can be exploited for an efficient solution. We analyze details for the individual steps of Algorithm 1 in the following.

3.1 Structure-exploiting solution of the Newton System

The dual gradient $\frac{\partial f^*}{\partial \lambda}(\lambda) \in \mathbb{R}^{Nn_x}$ (as a column vector) is easily seen to only depend on two neighboring stages in each block λ_k. It holds

$$\frac{\partial f^*}{\partial \lambda}(\lambda) := \begin{bmatrix} \frac{\partial f_0^*}{\partial \lambda_1} + \frac{\partial f_1^*}{\partial \lambda_1} \\ \frac{\partial f_1^*}{\partial \lambda_2} + \frac{\partial f_2^*}{\partial \lambda_2} \\ \vdots \\ \frac{\partial f_{N-1}^*}{\partial \lambda_N} + \frac{\partial f_N^*}{\partial \lambda_N} \end{bmatrix}(\lambda).$$

The dual Hessian $\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda) \in \mathbb{R}^{Nn_x \times Nn_x}$ possesses a block tri-diagonal structure, as only neighboring multipliers λ_k, λ_{k+1} can have a joint contribution to f^*. We have

$$\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda) := \begin{bmatrix} \frac{\partial^2 f^*}{\partial \lambda_1^2} & \frac{\partial^2 f^*}{\partial \lambda_1 \lambda_2} & & \\ \frac{\partial^2 f^*}{\partial \lambda_2 \lambda_1} & \frac{\partial^2 f^*}{\partial \lambda_2^2} & \ddots & \\ & \ddots & \ddots & \frac{\partial^2 f^*}{\partial \lambda_{N-1} \lambda_N} \\ & & \frac{\partial^2 f^*}{\partial \lambda_N \lambda_{N-1}} & \frac{\partial^2 f^*}{\partial \lambda_N^2} \end{bmatrix}(\lambda).$$

3.2 Dual function evaluation

Each stage problem (QP_k) has a fixed second order term H_k and a parametric first-order term $p_k(\lambda)$. An efficient method to solve such parametric problems repeatedly for changing parameter values λ is the so-called online active-set strategy (see Ferreau et al. (2008)); a well-established implementation of this algorithm that we propose to employ here is the open-source QP solver qpOASES, cf. Ferreau et al. (2008).

In the special case where H_k is a diagonal matrix and $D_k = I$ is an identity matrix (i.e., only bounds on states and controls of the MPC problem exist) the optimal solution z_k^* can conveniently computed by component-wise "clipping" of the unconstrained solution, here denoted by z_k, as it was presented in Ferreau et al. (2012a):

$$z_k^* = \max(\underline{d}_k, \min(z_k, \overline{d}_k)) \tag{3}$$

3.3 Explicit derivative computation

Due to the strict positive definiteness of H_k the derivative of each dual function summand f_k^* with respect to the multiplier contribution $\begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}$ exists for all $k \in \mathcal{S}$, and is given by (see Bertsekas and Tsitsiklis (1989), App. C for a formal derivation)

$$\begin{bmatrix} \frac{\partial f_k^*}{\partial \lambda_k}^\top & \frac{\partial f_k^*}{\partial \lambda_{k+1}}^\top \end{bmatrix} = z_k^{*\top} \begin{bmatrix} -E_k^\top & C_k^\top \end{bmatrix} + \begin{bmatrix} 0 & c_k \end{bmatrix}. \tag{4}$$

Observe that Equation (4) links the equivalence of primal feasibility and stationarity of the dual function f^*, justifying the termination criterion in Step 5 of Algorithm 1.

Differentiating (4) once more with respect to λ the non-zero dual Hessian blocks read (cf. Frasch et al. (2013))

$$\begin{aligned} \frac{\partial^2 f^*}{\partial \lambda_k \lambda_{k+1}} &= \frac{\partial}{\partial \lambda_k}\left(\frac{\partial f_k^*}{\partial \lambda_{k+1}} + \frac{\partial f_{k+1}^*}{\partial \lambda_{k+1}}\right) \\ &= \frac{\partial z_k^*}{\partial \lambda_k} C_k^\top + \underbrace{\frac{\partial z_{k+1}^*}{\partial \lambda_k}}_{=0} E_{k+1}^\top \\ &= E_k P_k^* C_k^\top \end{aligned}$$

and

$$\begin{aligned} \frac{\partial^2 f^*}{\partial \lambda_k \lambda_k} &= \frac{\partial}{\partial \lambda_k}\left(\frac{\partial f_{k-1}^*}{\partial \lambda_k} + \frac{\partial f_k^*}{\partial \lambda_k}\right) \\ &= \frac{\partial z_{k-1}^*}{\partial \lambda_k} C_{k-1}^\top - \frac{\partial z_k^*}{\partial \lambda_k} E_k^\top \\ &= -C_{k-1} P_{k-1}^* C_{k-1}^\top - E_k P_k^* E_k^\top. \end{aligned}$$

Here $P_k^* := Z_k^*(Z_k^{*\top} H_k Z_k^*)^{-1} Z_k^{*\top} \in \mathbb{R}^{n_z \times n_z}$ is an elimination matrix for the nullspace of the optimal active set of QP_k, with a nullspace basis matrix $Z_k^* \in \mathbb{R}^{n_z \times (n_z - n_{\text{act}})}$ suitably chosen to project the Hessian matrix H_k onto the active set of the (unique) optimal solution $z_k^*(\lambda)$ of QP_k, where we use n_{act} to denote the number of active constraints. When employing a null-space QP solver like qpOASES, Z_k^* and a Cholesky factor R for $R^\top R = Z_k^{*\top} H_k Z_k^*$ are directly available, making the computation

of P_k^* rather cheap. In the case where Equation (3) is used to solve the stage subproblems, P_k^* boils down to a diagonal matrix with entries either from H_k^{-1} or 0, depending on whether the corresponding variable bound is inactive or active, cf. Ferreau et al. (2012a).

3.4 Solution of the Newton system

The dual function f^* is concave, but not strictly concave everywhere, so $-\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda^i)$ is positive semi-definite. In case $-\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda^i)$ is rank-deficient, we propose to add a small regularization term $\epsilon \Delta\lambda$ for $\epsilon > 0$ in Step 7 of Algorithm 1. The resulting, possibly regularized, system system $\mathcal{M}(\lambda^i)\,\Delta\lambda = \frac{\partial f^*}{\partial \lambda}(\lambda^i)$ therefore has unique solution, and a Cholesky factorization $\mathcal{M}(\lambda^i) = LL^\top$ always exists. The fact that L possesses the same structural zero-blocks as $\mathcal{M}(\lambda^i)$ below the diagonal can be exploited in the factorization algorithm by skipping all redundant blocks left and below the subdiagonal block $-\frac{\partial^2 f^*}{\partial \lambda_{i+1} \lambda_i}$ of each block column i. Utilizing this modified Cholesky decomposition the solution of the (possibly regularized) Newton system requires only $O(N n_x^3)$ floating point operations (FLOPs) in Algorithm 1.

3.5 Choice of the Newton step size

The convergence of Algorithm 1 requires a globalization strategy (Step 8 in Algorithm 1) that finds an (approximate) solution to

$$\arg\max_{\alpha \in (0,1]} f^*(\lambda^i + \alpha \Delta\lambda), \tag{5}$$

where the search direction $\Delta\lambda$ is computed as suggested in Section 3.4. Note that due to the piecewise quadratic nature of $f^*(\lambda)$ the exact solution of (5) is computationally tractable. Still, we propose to employ a line search strategy for increased computational efficiency. It is important to observe in this context that based on the fact that Newton's method minimizes a local quadratic model of f^* we are able to state a lower bound for (5) based on the first primal active-set change in search direction. Readers may convince themselves of this by recalling that first-order kinks f^* are caused by primal active-set changes. A formal proof can be found in Frasch et al. (2013). Step sizes α at which an active set changes occurs are obtained for free when employing an online active set strategy to solve each (QP_k), cf. Ferreau et al. (2008). In cases where the solution to (QP_k) can be computed by Equation (3), points of active-set changes can still be obtained by a ratio test of the unconstrained and the clipped solution in each component.

Piecing everything together, Figure 1 illustrates the steps of Algorithm 1 in the dual space. Each cell corresponds to a constant primal active set and therefore to a region in which the dual function is quadratic (quadratic region/critical region). These cells indeed are polytopic as it can be observed from characterizations, e.g., in Ferreau et al. (2008). From an initial guess λ^0 the first full Newton step of Algorithm 1 would lead to λ^1_{FS}. Employing a line search, the step $\lambda^1_{\bar{\alpha}}$ is taken. Here, $\lambda^1_{\alpha_{\min}}$ denotes the step of minimum size given by the first active-set change in search direction. Analogously the second step causes

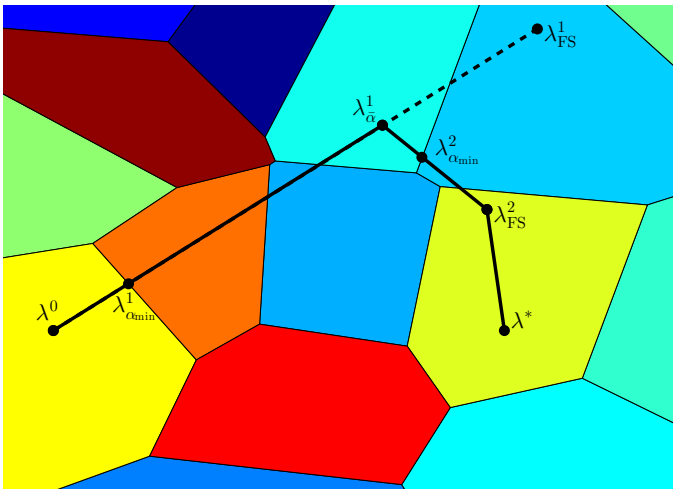

Fig. 1. Steps of Algorithm 1 in the dual space.

several primal active set changes, and attains the maximal value in search direction at the full step. By the Newton property we finally have a 1-step convergence to λ^* from within the correct quadratic region, cf. Frasch et al. (2013).

3.6 Warm-starting for series of QPs

A key advantage of the dual Newton strategy in the context of nonlinear MPC and MHE are its warm-starting capabilities. While interior point methods typically cannot be warmstarted efficiently, and the active set of a *condensed* QP can, even in the nominal case, change quite significantly from one sampling time to the next due to shifted state constraints, the optimization variables λ can be shifted alongside with the sampling time in Algorithm 1. It is notable that Newton's method guarantees a 1-step convergence in this context if the shifted λ-guess is in the correct quadratic region, i.e., if the optimal primal active set is consistent with the shifted one, even if the QP data changes (e.g., through re-linearization of the nonlinear problem in the RTI framework).

In detail, we suggest to perform the following simple shift from the optimal dual vector of the QP at sampling time s, λ^*, to an initial guess λ^0 for the following QP at sampling time $s+1$:

$$\begin{aligned} \lambda^0_k &:= \lambda^*_{k+1} \qquad && \forall\, k = 1, \ldots, N-1 \\ \lambda^0_N &:= \lambda^*_N \end{aligned}$$

In the nominal case, this shift ensures that λ^0 already lies in the correct quadratic region (and thus 1-step convergence if the primal terminal stage variables z_N lie in a stable active set (e.g., given by a steady state).

4. OPEN-SOURCE SOFTWARE IMPLEMENTATION

4.1 The structure-exploiting QP solver qpDUNES

The dual Newton strategy has been implemented in the open-source software package qpDUNES, which is available for download at qpDUNES (2013). It is a plain, self-contained C code written according to the C90 standard to enlarge compatibility with embedded hardware platforms. It comes with its own linear algebra module and efficient

data storage formats to better exploit the problem intrinsic structures. Memory allocation is performed on a global scale to enable reusability of memory blocks and to enable switching between dynamic memory allocation for maximum flexibility and static memory allocation for increased performance and deployment on embedded hardware. A code generation routine for the linear algebra modules tailored to the structure and dimensions of a specific problem instance for even higher efficiency is currently under development. Application of such code generation techniques has lead to significant performance increases in related areas like interior point solvers, cf. Mattingley and Boyd (2009); Domahidi et al. (2012), and NMPC Controllers, cf. Houska et al. (2011).

4.2 The ACADO Code Generation tool

The ACADO Toolkit (available at ACADO (2009)) is an open source software for the modeling, simulation and control of nonlinear dynamic processes. It is particularly suited to set up nonlinear MPC and MHE problems. It was recently extended by the ACADO Code Generation tool, see Houska et al. (2011), which allows to export a lean RTI scheme tailored for a specific problem's requirements. It is essentially based on Bock's multiple-shooting discretization, see Bock and Plitt (1984), and a Gauss-Newton method for the solution of the obtained nonlinear program (NLP). For the evaluation of the possibly nonlinear dynamic system both constant step-size explicit and implicit Runge-Kutta ingrators are available, cf. Quirynen et al. (2012). In its default configuration, the solution of the structured quadratic subproblems is based on an optimized condensing routine, cf. Andersson et al. (2013) for reduction of the problem size. The dense QPs are then solved using the online QP solver qpOASES Ferreau et al. (2008). This configuration has already been successfully applied to a variety of applications, e.g., Vukov et al. (2012); Ferreau et al. (2012b); Houska et al. (2011). In the latest release, an interface for a direct solution of the sparse quadratic subproblems via qpDUNES has been added, that alternatively allows to avoid the condensing step and to exploit the problem-inherent sparsity structure directly as described above.

5. NUMERICAL TESTS

We provide a comparison of those two QP solution strategies within the ACADO Code Generation tool in the following. All simulations were performed on a 3.4GHz Intel i7 based desktop computer, running the 64-bit version of Ubuntu Linux 13.04. All codes are compiled with Clang 3.2.1, using the flag `-O3` and execution times are measured with Linux function `clock_gettime()`.

The first classes of benchmark problems deals with stabilizing a strongly deflected chain of M masses connected by springs after an initial perturbation, as described in Wirsching et al. (2006) and also used for benchmarking in Ferreau et al. (2008) and Vukov et al. (2013). One end of the chain is attached to a fixed point, while the velocity of the other end can be controlled. Each mass is described by its position and velocity coordinates, resulting in a total of $n_x = 6M + 3$ states and $n_u = 3$ control inputs. We chose a sampling time of $T_s = 200\,\text{ms}$ and varying prediction horizon lengths $N \in [10, \dots, 100]$. For integration of the nonlinear system dynamics an implicit Gauss-Legendre integrator of order four was used, with two integration step per discretization interval. for a more challenging scenario.

In the RTI scheme each iteration is split in a preparation and a feedback phase, which we consider separately (a basic familiarity of the reader with the RTI scheme again is assumed here; otherwise we kindly refer to Diehl et al. (2002)). In the condensing-based approach, the preparation phase consists of the linearization of the NLP and the condensing routine that yields the reduced-size QP, both of which are of significant computational effort. The time spent in the feedback phase is dominated by the solution of the condensed QP by qpOASES. In the sparse (qpDUNES-based) approach, no condensing routine is needed, so the preparation phase is dominated only by the effort for the linearization of the NLP. Almost all time of the feedback phase is then spent in the solution of the sparse QP.

We present average computation times for one RTI in Figure 2 and highlight the time spent in the feedback phase (note that the preparation phase always has to be shorter in the sparse QP strategy). It can be seen that for a moderate number of states n_x, the sparse approach is already competitive on short horizon lengths, while it clearly outperforms the condensing-based approach both in terms of feedback time and in terms of total iteration time on longer horizon lengths due to its lower per-iteration computational complexity. The test case for $M = 5$ shows the benefits of the condensing-based approach for systems with many states that lie in a drastically reduced size of the problem that remains to be solved in the feedback phase; still, regarding the overall iteration times, the sparse approach performs reasonably well, even though it is not targeted for this class of problems.

A similar pattern can be observed when regarding maximum computation times over all simulation steps in Figure 3. For a small to medium number of states, the sparse approach dominates already on relatively short horizon lengths, both in the feedback phase and (thus even more) in the total iteration time. Obviously the computational efforts for integration and condensing are problem-data independent and thus the relative gap between both approaches decreases a bit for long horizons in the consideration of maximum computation times, as observed in the test case of $M = 3$. For a rather big number of states (big at least for fast scale applications), particularly seen in the test cases $M = 5$, the feedback time (i.e., the pure QP solution time) of the condensed approach dominates the QP solution time of the sparse approach for all considered horizon lengths. This effect is even more visible in the maximum computation time plot, since again, the time spent in condensing plays less of a roll here.

Particularly in the test case $M = 5$ it could be observed that the sparse approach suffers more than the condensed approach from the bad conditioning of the problems, resulting from the objective function pressing strongly against the constraints. Improving the stability of factorization and smarter regularization approaches are subject of ongoing research. In general it should be noted that the relatively large gap between average and maximum computation times of the dual Newton method can partly

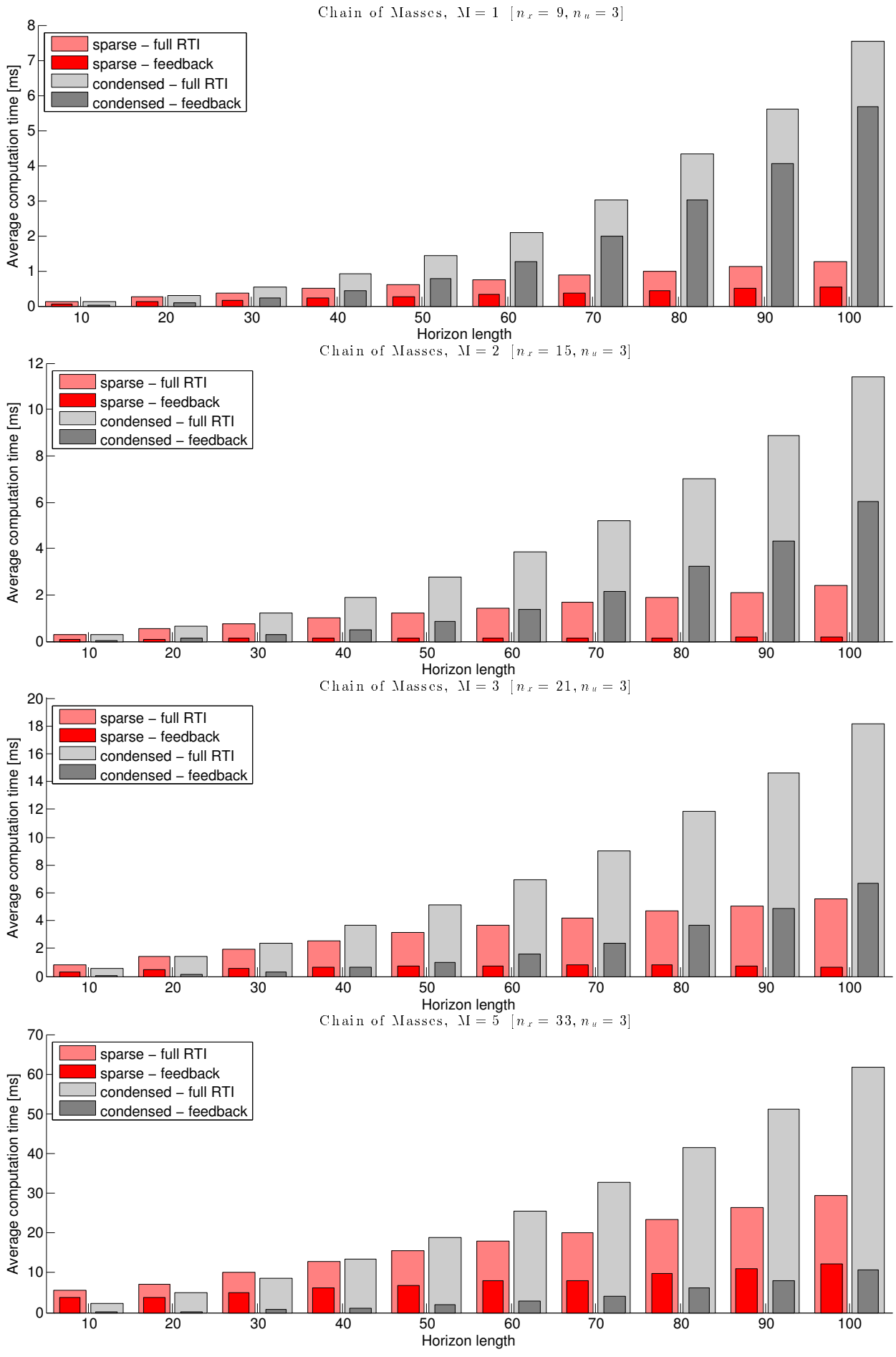

Fig. 2. Average computation time benchmark for four chain-of-masses test cases

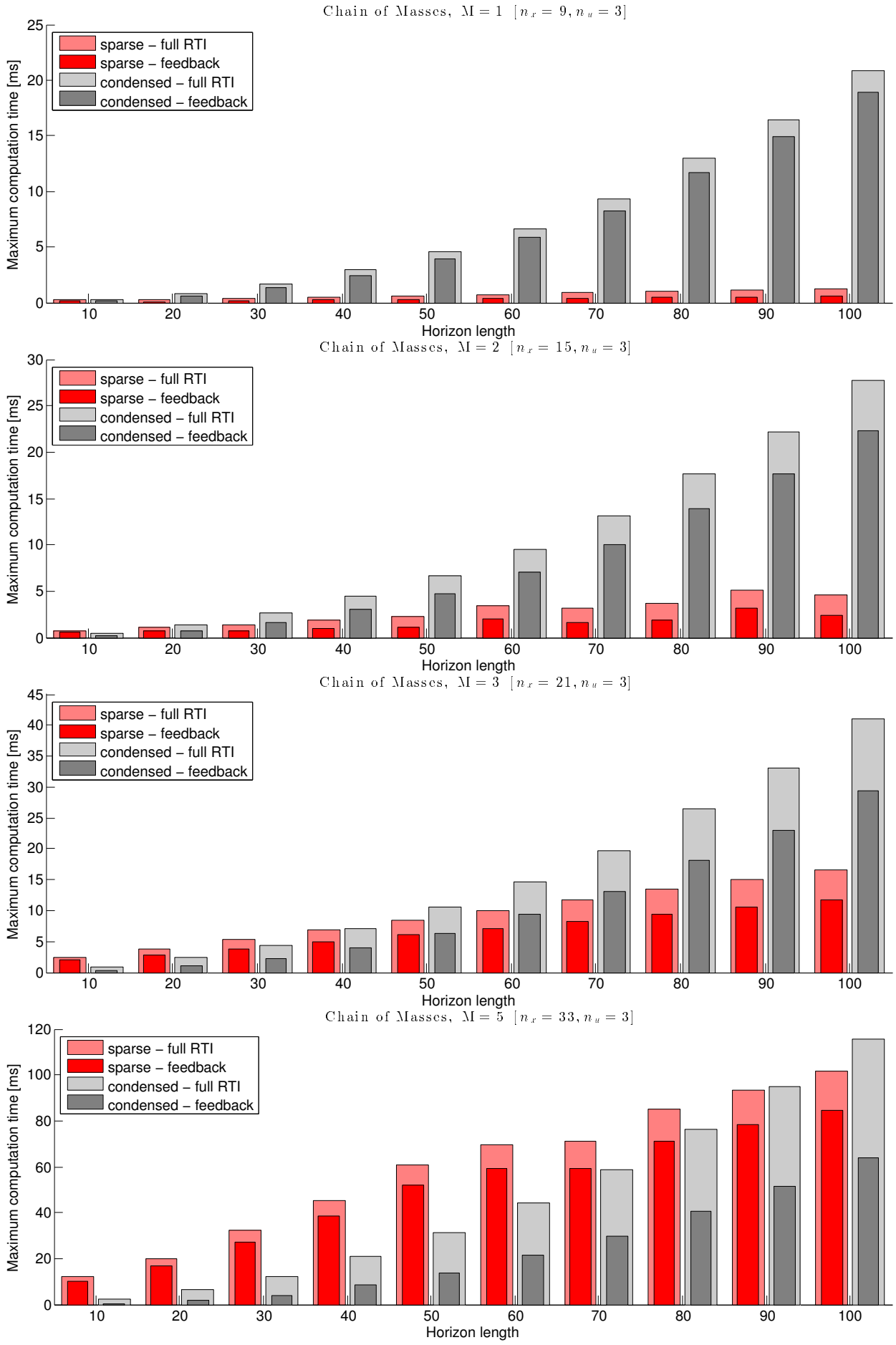

Fig. 3. Maximum computation time benchmark for four chain-of-masses test cases

be explained by the wrong initalization of the QP solver after the chain is deflected. Excluding the first iteration from considerations already leads to significantly shorter maximum computation times, e.g., about 20 ms for the sparse QP solution in the $M = 5$ case on a horizon length of 100 steps. The other part of the explanation obviously is the active set method nature of the dual Newton strategy.

While the first benchmark problem is purely academic (but well scalable), we use a real-world motivated second benchmark problem. Nonlinear MPC is used to prevent the occurence of surge in centrifugal compressors (see Cortinovis et al. (2012) for details). Centrifugal compressors are widely used in gas extraction plants or gas pipelines to extract and transport natural gas from the source to the consumer. As compressing is an energy-intensive process, it is important to operate compressors efficiently in order to save resources. This means to operate them at working points that are close to surge, an instable system state that can cause severe damage to the compressor and piping system. We describe the compressor by a nonlinear ODE model similar to the one presented in Cortinovis et al. (2012). It comprises $n_x = 6$ differential states and $n_u = 2$ control inputs: the opening of the recycle valve as well as the torque of the compressor's drive (which are both subject to physical limitations). Our nonlinear MPC aims at tracking a given operating point in the event of a simulated sudden closure of the compressor's outlet valve.

The anti-surge controller is running at a sampling time of 25 ms on prediction horizon of length N as given in Figure 4. Again, the sparse approach performs competitively on all considered horizon lengths, tieing for horizon lengths around $N = 30$. For longer prediction horizons significant computational savings can be achieved applying the sparse approach. Due to the practical relevance we also include maximum computation times in the lower part of Figure 4. Since both, the sparse and the condensed approach, rely on active-set based QP solution methods, the absolute computation times increase. The problem data independent effort for integration and condensing causes the relative performance gap between the sparse and the condensed approach on long horizons to diminish slightly, while the absolute gap even increases.

6. CONCLUSIONS AND FUTURE WORK

We presented a new quadratic programming strategy for an efficient solution of sparse quadratic subproblems in and nonlinear MPC and MHE of long horizon problems. Numerical results on several test cases showed the competitiveness and and the potential of this approach in comparison with existing, already well-tuned methods. Currently ongoing research on the method side addresses parallelization aspects and code-generation of the linear algebra routines.

REFERENCES

ACADO Toolkit (2009). `http://www.acadotoolkit.org`

qpDUNES (2013). `http://mathopt.de/qpdunes`

Andersson, J.A.E., Frasch, J.V., Vukov, M., and Diehl, M. (2013). A Condensing Algorithm for Nonlinear MPC with a Quadratic Runtime in Horizon Length. *Automatica*. (under review).

Bemporad, A. and Patrinos, P. (2012). Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. In *4th IFAC Nonlinear Model Predictive Control Conference*, 14–20.

Berkelaar, A., Roos, K., and Terkaly, T. (1997). *Recent Advances in Sensitivity Analysis and Parametric Programming*, chapter 6: The Optimal Set and Optimal Partition Approach to Linear and Quadratic Programming. Kluwer Publishers.

Bertsekas, D. and Tsitsiklis, J.N. (1989). *Parallel and distributed computation: Numerical methods*. Prentice Hall.

Bock, H. and Plitt, K. (1984). A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC World Congress Budapest*, 242–247. Pergamon Press.

Cortinovis, A., Pareschi, D., Mercangoez, M., and Besselmann, T. (2012). Model predictive anti-surge control of centrifugal compressors with variable-speed drives. In *Proceedings of the 2012 IFAC Workshop on Automatic Control in Offshore Oil and Gas Production, Trondheim, Norway*, 251–256.

Dai, Y.H. and Fletcher, R. (2006). New algorithms for singly linearly constrained quadratic programs subject to low and upper bounds. *Mathematical Programming*, 106(3), 403–421.

Diehl, M., Bock, H., Schlöder, J., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4), 577–585.

Domahidi, A., Zgraggen, A., Zeilinger, M., Morari, M., and Jones, C. (2012). Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control. In *51st IEEE Conference on Decision and Control*, 668 – 674.

Ferreau, H.J., Bock, H.G., and 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.

Ferreau, H., Kozma, A., and Diehl, M. (2012a). A parallel active-set strategy to solve sparse parametric quadratic programs arising in MPC. In *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*.

Ferreau, H., Kraus, T., Vukov, M., Saeys, W., and Diehl, M. (2012b). High-speed moving horizon estimation based on automatic code generation. In *51th IEEE Conference on Decision and Control*.

Fiacco, A. (1983). *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press.

Frasch, J.V., Sager, S., and Diehl, M. (2013). A Parallel Quadratic Programming Method for Dynamic Optimization Problems. *Mathematical Programming Computations*. Available at `http://www.optimization-online.org/DB_HTML/2013/11/4114.html`. (under review).

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

Kirches, C., Wirsching, L., Bock, H., and Schlöder, J. (2012). Efficient Direct Multiple Shooting for Nonlinear Model Predictive Control on Long Horizons. *Journal of*

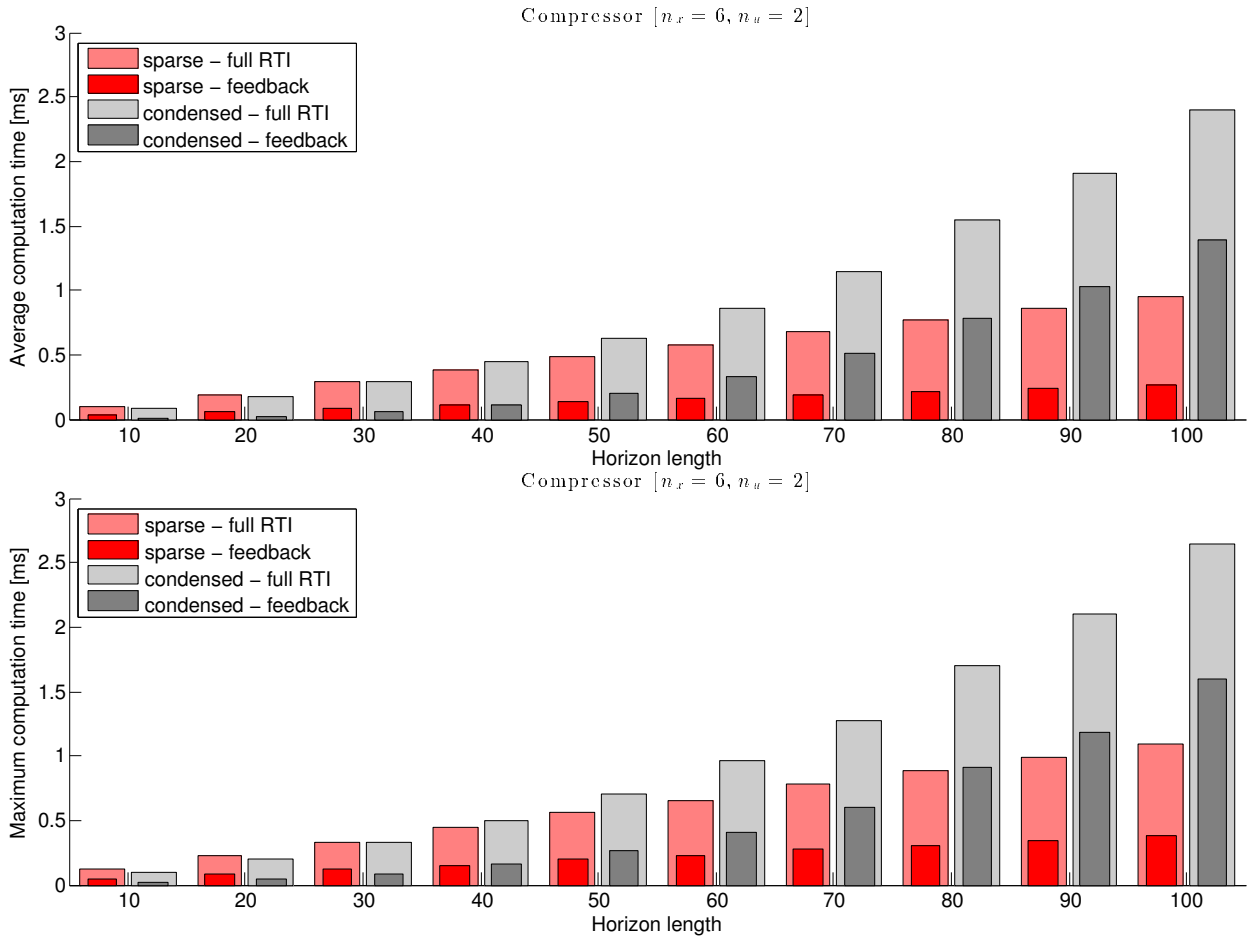

Fig. 4. Average and maximum computation time benchmark for the compressor test case

Process Control, 22(3), 540–550.

Kühl, P., Diehl, M., Kraus, T., Schlöder, J.P., and Bock, H.G. (2011). A real-time algorithm for moving horizon state and parameter estimation. *Computers & Chemical Engineering*, 35(1), 71–83.

Leineweber, D. (1999). *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.

Li, W. and Swetits, J. (1997). A new algorithm for solving strictly convex quadratic programs. *SIAM Journal of Optimization*, 7(3), 595–619.

Mattingley, J. and Boyd, S. (2009). *Convex Optimization in Signal Processing and Communications*, chapter Automatic Code Generation for Real-Time Convex Optimization. Cambridge University Press.

Quirynen, R., Vukov, M., and Diehl, M. (2012). Auto Generation of Implicit Integrators for Embedded NMPC with Microsecond Sampling Times. In M. Lazar and F. Allgöwer (eds.), *Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference*.

Rawlings, J. and Mayne, D. (2009). *Model Predictive Control: Theory and Design*. Nob Hill.

Richter, S., Jones, C., and Morari, M. (2009). Real-time input-constrained MPC using fast gradient methods. In *48th IEEE Conference on Decision and Control*.

Vukov, M., Domahidi, A., Ferreau, H.J., Morari, M., and Diehl, M. (2013). Auto-generated Algorithms for Nonlinear Model Predicitive Control on Long and on Short Horizons. In *52nd Conference on Decision and Control*.

Vukov, M., Loock, W.V., Houska, B., Ferreau, H., Swevers, J., and Diehl, M. (2012). Experimental Validation of Nonlinear MPC on an Overhead Crane using Automatic Code Generation. *The 2012 American Control Conference*.

Wirsching, L., Bock, H.G., and Diehl, M. (2006). Fast NMPC of a chain of masses connected by springs. In *Proceedings of the IEEE International Conference on Control Applications*, 591–596.

Zafiriou, E. (1990). Robust model predictive Control of processes with hard constraints. *Computers & Chemical Engineering*, 14(4–5), 359–371.

Zavala, V.M. and Biegler, L. (2009). The Advanced Step NMPC Controller: Optimality, Stability and Robustness. *Automatica*, 45, 86–93.