

# A Parallel Quadratic Programming Method for Dynamic Optimization Problems

Janick V. Frasch · Sebastian Sager ·  
Moritz Diehl

Received: date / Accepted: date

**Abstract** Quadratic programming problems (QPs) that arise from dynamic optimization problems typically exhibit a very particular structure. We address the ubiquitous case where these QPs are strictly convex and propose a dual Newton strategy that exploits the block-bandedness similarly to an interior-point method. Still, the proposed method features warmstarting capabilities of active-set methods. We give details for an efficient implementation, including tailored numerical linear algebra, step size computation, parallelization, and infeasibility handling. We prove convergence of the algorithm for the considered problem class. A numerical study based on the open-source implementation qpDUNES shows that the algorithm outperforms both well-established general purpose QP solvers as well as state-of-the-art tailored control QP solvers significantly on the considered benchmark problems.

## 1 Introduction

A large class of practical algorithms for the solution of dynamic optimization problems, as they appear for example in *Optimal Control* and *Dynamic Parameter Estimation*, is based on sequential quadratic programming (SQP) [6, 7, 10, 24]. Particularly for their online variants, *Model predictive control* (MPC) and *Moving Horizon Estimation* (MHE), a fast solution of the arising quadratic programming (QP) subproblems is crucial [10, 24, 31], exploiting all problem-inherent structure.

---

J. Frasch and S. Sager  
Institute of Mathematical Optimization, Otto von Guericke University Magdeburg, Germany  
Tel.: +49 391 67-18745  
Fax: +49 391 67-11171  
{frasch, sager}@ovgu.de

M. Diehl  
Department of Electrical Engineering, KU Leuven, Belgium  
moritz.diehl@esat.kuleuven.be

In a generic form the QP subproblems arising in this class of methods can be summarized as follows:

$$\min_{x,u} \sum_{k=0}^{N-1} \left( \frac{1}{2} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} Q_k & S_k \\ S_k^\top & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \begin{bmatrix} q_k \\ r_k \end{bmatrix}^\top \begin{bmatrix} x_k \\ u_k \end{bmatrix} \right) + \frac{1}{2} x_N^\top Q_N x_N + q_N^\top x_N \quad (1a)$$

$$\text{s.t. } x_{k+1} = A_k x_k + B_k u_k + c_k \quad \forall k \in \mathcal{S}_N \quad (1b)$$

$$\underline{d}_k \leq D_k \begin{bmatrix} x_k \\ u_k \end{bmatrix} \leq \bar{d}_k \quad \forall k \in \mathcal{S}. \quad (1c)$$

Discretized state variables are denoted by  $x_k \in \mathbb{R}^{n_x}$ , while control variables are denoted by  $u_k \in \mathbb{R}^{n_u}$  for each discretization stage  $k \in \mathcal{S} := \{0, \dots, N\}$  and  $k \in \mathcal{S}_N$ , respectively. In general we use subscripts to indicate exclusions from a set, e.g.,  $\mathcal{S}_{i,j} := \mathcal{S} \setminus \{i, j\}$ . For notational simplicity, we hide time-constant parameters in the state vector. The stages are coupled over the discretized time horizon of length  $N$  by Constraints (1b), while Constraints (1c) arise from discretized path constraints. The objective (1a) is often a quadratic approximation to the Lagrangian of the original control or estimation problem. We assume throughout this paper that (1a) is strictly convex, i.e.,  $\begin{bmatrix} Q_k & S_k \\ S_k^\top & R_k \end{bmatrix} \succ 0$ .

In a typical SQP scheme for dynamic optimization, QP (1) is solved repeatedly, each time with updated (relinearized) data, until a termination criterion is met. While the QP data changes, one often assumes that the active set, i.e., those equations in (1c) that are satisfied tightly with equality, is similar from one iteration to the next, particularly in later iterations. In the large class of MPC and MHE algorithms that feature linear time-invariant (LTI) dynamic systems, QP (1) is solved repeatedly for different initial condition (fixations of  $x_0$ ) with otherwise identical QP data.

QP (1) is very sparse for long time horizons  $N$ , a fact we strive to exploit in this paper. The block-banded structure exhibited by (1) is typically exploited well by tailored interior point (e.g., [30, 27, 11]) and fast-gradient methods (e.g., [32, 2]). However, a well-known drawback of these methods is their limited warm-starting capability to exploit the knowledge of similarity between the solutions of subsequently solved QPs.

Tailored active set-methods like [12, 13] on the other hand can exploit this similarity for higher efficiency, but typically do not benefit from the problem-inherent sparsity as much as interior point methods. One popular approach to overcome this is to make use of a so-called condensing routine to express overparameterized state variables by means of the control inputs [25, 22]. In general, this condensing step needs to be performed at every sampling time, with a typically cubic runtime complexity in the horizon length. Furthermore, the initial factorization of the dense QP Hessian is also of cubic runtime complexity in the horizon length, see [23]. Alternatively, an active-set method with sparsity exploitation is suggested in [22]. Due to the fact that each active set change calls for a new sparse matrix factorization it is most useful when only few active set changes occur.

In this paper, we follow up on the idea of a novel QP algorithm that was presented for linear MPC problems in [14]. This approach aims at combining the benefits in terms of structure exploitation of interior point methods with the warm-starting capabilities of active set methods, and comes at only a linear runtime complexity in the horizon length. Based on ideas from [26] and [8] the stage coupling constraints (1b) induced by the MPC structure are dualized and the resulting QP is solved in a two level approach, using a non-smooth/semismooth 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. Note that in contrast to classical active-set methods, this approach permits several active-set changes at the cost of one block-banded matrix factorization. We refer to this procedure as a *dual Newton strategy* in the following and give more details on the method in Section 2.

While in [14] only a limited prototypic Matlab implementation was described, this work extends the algorithm to a more general problem class and gives details on an efficient implementation of the method. We further provide a theoretical foundation to the algorithm and proof convergence. We discuss parallelization aspects of the method and present a novel algorithm for the solution of the structured Newton system, that results in an parallel runtime complexity of the dual Newton strategy of  $\mathcal{O}(\log N)$  per iteration. Most importantly, we present qpDUNES, an open-source, plain C implementation of the *DUAL NEWTON STRATEGY*. This software comes with interfaces for C/C++ and Matlab. We compare runtimes of qpDUNES to state of the art structure-exploiting QP solvers for dynamic optimization problems based on three challenging control benchmark problems.

## 2 Method Description

For clarity of presentation we group the optimization variables of MPC problems, system states  $x_k \in \mathbb{R}^{n_x}$  and control inputs  $u_k \in \mathbb{R}^{n_u}$ , in stage variables  $z_k = [x_k^\top u_k^\top]^\top \in \mathbb{R}^{n_z}$  for each stage  $k \in \mathcal{S}_N$ , and  $z_N = [x_N 0]^\top \in \mathbb{R}^{n_z}$  for the terminal stage. We are therefore interested in solving the following type of problem:

$$\min_z \sum_{k=0}^N \left( \frac{1}{2} z_k^\top H_k z_k + g_k^\top z_k \right) \quad (\text{P1})$$

$$\text{s.t. } E_{k+1} z_{k+1} = C_k z_k + c_k \quad \forall k \in \mathcal{S}_N \quad (\text{P2})$$

$$\underline{d}_k \leq D_k z_k \leq \bar{d}_k \quad \forall k \in \mathcal{S}. \quad (\text{P3})$$

The cost function on each stage consists of a positive definite second-order term  $0 \prec H_k \in \mathbb{R}^{n_z \times n_z}$  and a first-order term  $g_k \in \mathbb{R}^{n_z}$  for each  $k \in \mathcal{S}$ . 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, i.e.,  $\text{rk}(C_k) = n_x$ ,  $\forall k \in \mathcal{S}_N$ , 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  $\underline{d}_k, \bar{d}_k \in \mathbb{R}^{n_d}$ , and a matrix  $D_k \in \mathbb{R}^{n_d \times n_z}$  of full row rank denote affine stage constraints.

If not stated differently, we assume in the following that a feasible solution  $z^* := [z_0^{*\top}, \dots, z_N^{*\top}]^\top$  of (P) exists that fulfills the linear independence constraint qualification (LICQ). Section 5 will discuss the consequences resulting from infeasibility of (P) and state how it can be detected.

## 2.1 Dual decomposition

We decouple the QP stages by dualizing constraints (P2). Introducing

$$\lambda := [\lambda_1^\top \ \lambda_2^\top \ \dots \ \lambda_N^\top]^\top \in \mathbb{R}^{Nn_x} \quad (3)$$

we can express (P1) and (P2) by the partial Lagrangian function

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

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 in this context (note that they are not among the optimization variables of the dual problem defined below).

By Lagrangian duality the solution of (P) can therefore be computed as

$$\begin{aligned} \max_{\lambda} \min_z \sum_{k=0}^N \left( \frac{1}{2} z_k^\top H_k z_k + g_k^\top z_k + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} z_k + \lambda_{k+1}^\top c_k \right) \\ \text{s.t. } \underline{d}_k \leq D_k z_k \leq \bar{d}_k \quad \forall k = 0, \dots, N. \end{aligned}$$

As this Problem is separable in the stage variables  $z_k$ , minimization and summation can be interchanged, and a solution to (P) is obtained by solving

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

where

$$\begin{aligned} f_k^*(\lambda) &:= \min_{z_k} \frac{1}{2} z_k^\top H_k z_k + \left( g_k^\top + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} \right) z_k + \lambda_{k+1}^\top c_k \\ \text{s.t. } \underline{d}_k &\leq D_k z_k \leq \bar{d}_k. \end{aligned} \quad (\text{QP}_k)$$

We refer to  $(\text{QP}_k)$  as stage QP. Note that each  $(\text{QP}_k)$  depends on at most two block components of the vector of dual optimization variables  $\lambda$  defined in (3).

*Remark 1* Since  $\lambda$  only enters in the objective of each  $(\text{QP}_k)$ , feasibility of  $(\text{QP}_k)$ , and thus existence of  $f_k^*(\lambda)$  is independent of the choice of  $\lambda \in \mathbb{R}^{Nn_x}$ . In particular, since the constraints of  $(\text{QP}_k)$  are a subset of (P3), feasibility of (P) implies feasibility of  $(\text{QP}_k)$ .

*Remark 2* Each  $f_k^*(\lambda)$  implicitly defines a  $z_k^*(\lambda)$ , the solution of  $(\text{QP}_k)$ .

## 2.2 Characterization of the dual function

It was shown in [14] (based on results from [15, 38, 3]) that  $f^*(\lambda)$  is concave, piecewise quadratic, and once continuously differentiable. We establish the relevant findings in the following.

**Definition 1** For a stage  $k \in \mathcal{S}$ , the optimal *active set* at  $\lambda$  is given by

$$\mathcal{A}_k^*(z_k^*(\lambda)) := \{1 \leq i \leq n_d \mid D_k^i \cdot z_k^*(\lambda) = \underline{d}_k^i \vee D_k^i \cdot z_k^*(\lambda) = \bar{d}_k^i\},$$

i.e., the set of row indices of the constraints of  $(\text{QP}_k)$  that are fulfilled with equality.

Definition 1 naturally extends to a definition of the active set in the full space of primal variables by  $\mathcal{A}^*(z^*(\lambda)) := \mathcal{A}_0^*(z_0^*(\lambda)) \times \dots \times \mathcal{A}_N^*(z_N^*(\lambda))$ . The finite number of disjoint active sets further induces a subdivision of the dual  $\lambda$  space:

**Definition 2** Each active set defines a *region*  $A \subseteq \mathbb{R}^{Nn_x}$  in the dual  $\lambda$  space. For a representative  $\lambda^{(j)} \in \mathbb{R}^{Nn_x}$  we have

$$A^{(j)} := \{\lambda \in \mathbb{R}^{Nn_x} \mid \mathcal{A}^*(z^*(\lambda)) = \mathcal{A}^*(z^*(\lambda^{(j)}))\}.$$

By choosing representatives of pairwise distinct regions we can define an arbitrary, but fixed order that allows us to uniquely identify each region  $A^{(j)}$ .

The name *region* is anticipatory, but it will become clear from Corollary 2 that the sets  $A^{(j)}$  indeed are connected. The number of regions  $n_r$  clearly is finite, as there is only a finite number of distinct active sets. From Remark 1 we can conclude that each  $\lambda \in \mathbb{R}^{Nn_x}$  is contained in a region  $A^{(j)}$ , and thus

$$\bigcup_{1 \leq j \leq n_r} A^{(j)} = \mathbb{R}^{Nn_x}.$$

*Remark 3* Two distinct regions  $A^{(j_1)}$  and  $A^{(j_2)}$  do not need to be disjoint. Values of  $\lambda$  that lead to weakly active stage constraints are contained in two or more regions. These values of  $\lambda$  form the seams of the regions.

Next, we substantiate Remark 2 by characterizing the nature of the dependency of  $z_k^*$  on the dual variables  $\lambda$  in the stage problems (QP<sub>k</sub>).

**Lemma 1** *Let (QP<sub>k</sub>) be feasible. Then, the optimal solution of (QP<sub>k</sub>),  $z_k^*(\lambda)$ , is a piecewise affine and continuous function in  $\lambda$ . In particular, the dependency is affine on each region  $A^{(j)}$ ,  $1 \leq j \leq n_r$*

*Proof* (cf. [14], Lem. 2; [38]) For stage Lagrange multipliers  $\mu_k \in \mathbb{R}^{2n_d}$  the solution of (P) is given by (see, e.g., [16])

$$\begin{bmatrix} H_k & -\underline{D}_k^{*\top} & \overline{D}_k^{*\top} \\ -\underline{D}_k^* & & \\ \overline{D}_k^* & & \end{bmatrix} \begin{bmatrix} z_k^* \\ \mu_k^* \end{bmatrix} = \begin{bmatrix} -\left( g_k^\top + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} \right) \\ \underline{d}_k^* \\ -\overline{d}_k^* \end{bmatrix}, \quad (4)$$

where  $\underline{D}_k^*$ ,  $\underline{d}_k^*$  and  $\overline{D}_k^*$ ,  $\overline{d}_k^*$  consist of the rows of  $D_k$ ,  $\underline{d}_k$ , and  $\overline{d}_k$  that correspond to the constraints that are active (i.e., fulfilled with equality) at the lower or, respectively, the upper bound in the solution  $z_k^*$ , and  $\mu_k^*$  is the vector of consistent dimension that contains the corresponding multipliers. The remaining stage multiplier entries are 0 in the solution of (P). As  $\lambda$  enters affinely only on the right-hand side, it is clear that for identical active sets it holds that  $z_k^*$  depends affinely on  $\lambda$ . Continuity has been shown in [15].  $\square$

**Corollary 2** *Each region  $A^{(j)}$ ,  $1 \leq j \leq n_r$ , of the dual space is convex and polyhedral.*

*Proof* Each stage problem (QP<sub>k</sub>) is constrained by affine constraints. For a given representative  $\lambda^{(j)}$  the set  $\mathcal{F}_k := \{z_k \in \mathbb{R}^{n_z} \mid \mathcal{A}_k(z_k) = \mathcal{A}_k^*(z_k^*(\lambda^{(j)}))\}$  is therefore convex and polyhedral. From Lemma 1 we have that  $z_k^*$  is affine in  $\lambda$  for a certain (fixed) active set. A region  $A^{(j)}$  is therefore the intersection of  $N+1$  (i.e., a finite number) preimages of convex sets, and therefore convex.  $\square$

Lemma 1 is the basis for the following exhaustive characterization of the dual function  $f^*(\lambda)$ , which we take from [14] without proof.

**Lemma 3** ([14], Lem. 3) *If all stage QPs (QP<sub>k</sub>) are feasible, then the dual function  $f^*(\lambda)$  exists and is described by a concave, continuously differentiable, and piecewise quadratic spline in  $\lambda$  space.*

*Remark 4* In particular  $f^*(\lambda)$  is quadratic on each region  $A^{(j)}$ .

### 2.3 Solution by a (non-smooth) Newton method

As (D) is an unconstrained problem and  $f^*(\lambda)$  is a piecewise-quadratic spline we employ a non-smooth Newton method, as originally proposed in [26], and also used in [14]. We iterate

$$\lambda^{i+1} := \lambda^i + \alpha \Delta \lambda \quad (5)$$

**Algorithm 1:** Dual Newton Strategy

---

**Input:** Initial guess  $\lambda^0$ , termination criteria  $n_{\max\text{It}}$ ,  $\epsilon_\lambda$   
**Output:** Optimal solution  $(z^*, \lambda^*, \mu^*)$

- 1 **for**  $i = 0 : (n_{\max\text{It}} - 1)$  **do**
- 2     Solve all  $\text{QP}_k(\lambda^i)$  to obtain  $[z_k^*(\lambda^i), \mu_k^*(\lambda^i)]$
- 3     Set up gradient  $\mathcal{G}(\lambda^i)$
- 4     **if**  $\|\mathcal{G}(\lambda^i)\| \leq \epsilon_\lambda$  **then**
- 5         **return**  $[z_k^*(\lambda^i), \lambda^i, \mu_k^*(\lambda^i)]$
- 6     Set up Newton matrix  $\mathcal{M}(\lambda^i)$
- 7     Solve Newton system (6)
- 8     Compute appropriate step size  $\alpha$
- 9     Update current iterate  $\lambda^{i+1} := \lambda^i + \alpha\Delta\lambda$

---

(the Newton iterates  $\lambda^i$  are not to be confused with the region representatives  $\lambda^{(j)}$  from Section 2.2) for an initial guess  $\lambda^0$  and a suitably chosen step size  $\alpha$  until  $f^*(\lambda^i)$  is stationary. The step direction  $\Delta\lambda$  is computed from

$$\mathcal{M}(\lambda^i) \Delta\lambda = \mathcal{G}(\lambda^i), \quad (6)$$

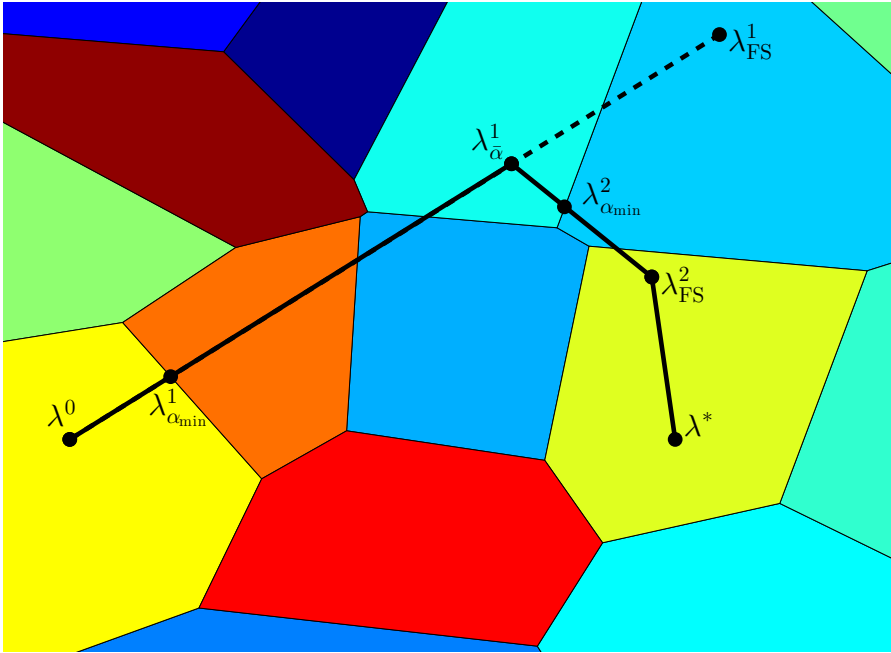
where  $\mathcal{M}(\lambda^i) := -\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda^i)$  and  $\mathcal{G}(\lambda^i) := \frac{\partial f^*}{\partial \lambda}(\lambda^i)$ . By Remarks 4 and 3,  $\mathcal{M}(\lambda)$  is unique everywhere but on the seams of  $f^*$  (a null set), where an arbitrary, but fixed second derivative from the finite number of possible choices is used, ensuring well-definedness of  $\mathcal{M}(\lambda)$ .

Clearly, stationarity of  $f^*(\lambda^i)$  is equivalent to optimality of (D). Observe that by definition of (D),  $z^*(\lambda^i)$  is always optimal and (P3) are always fulfilled in the spirit of an active set method. Lemma 4 will show that feasibility of (P2) is identical with stationarity of  $\mathcal{G}(\lambda^i)$ .

The complete QP solution method is given in Algorithm 1, where we denote the Lagrange multipliers of the stage constraints (P3) by  $\mu_k \in \mathbb{R}^{2n_d}$  for each stage  $k \in \mathcal{S}$ . Note that the parametric solution of the stage problems  $\text{QP}_k$  for the current iterate  $\lambda^i$  in Step 2, as well as the setup of  $\mathcal{G}(\lambda^i)$  and  $\mathcal{M}(\lambda^i)$  in Steps 3 and 6 permits an independent, concurrent execution on all stages (see also Section 6). A convergence proof for Algorithm 1 is given in Section 4.

#### 2.4 Characterization of the dual Newton iterations

A full QP solution by Algorithm 1 can be visualized as in Figure 1. Each cell corresponds to a region  $A^{(j)}$  in  $\lambda$ -space, for which the primal active set is constant. Starting from an initial guess  $\lambda^0$ , a Newton step direction is computed from equation (6) that leads to  $\lambda_{\text{FS}}^1$ . Using a globalization strategy (see Section 3.6), a suitable step size  $\lambda_\alpha^1$  is found. In contrast to classical active-set methods, *multiple* active set changes are possible *in one iteration*. For future reference we also indicate a minimum guaranteed step  $\lambda_{\alpha_{\min}}^1$ . In the second iteration,



**Fig. 1** Steps of the dual Newton strategy in the  $\lambda$ -space.

$\lambda_{\text{FS}}^2$  already provides sufficient progress, thus no globalization is applied. In the following iteration  $\lambda^*$  is found. We will prove in Lemma 8 that a one-step terminal convergence is guaranteed, once the correct region is identified.

### 3 Algorithmic details of the dual Newton strategy

The dynamic optimization origin induces a specific structure in Problem (D), that we strive to exploit in the following.

#### 3.1 Solution of decoupled parametric stage QPs

On each stage  $k \in \mathcal{S}$  we have to repeatedly solve a QP of size  $(n_z, n_d)$  that only changes in the first-order term (and the negligible constant term) with the current guess of  $\lambda$ . We have:

$$\begin{aligned} \text{QP}_k &= \min_{z_k} \frac{1}{2} z_k^\top H_k z_k + m_k(\lambda)^\top z_k + p_k(\lambda) \\ &\text{s.t. } \underline{d}_k \leq D_k z_k \leq \bar{d}_k, \end{aligned}$$



with  $m_k(\lambda)^\top := g_k^\top - \lambda_k E_k + \lambda_{k+1} C_k$ ,  $p_k(\lambda) := \lambda_{k+1}^\top c_k$ , and, in general,  $H_k$  and  $D_k$  dense. Such QPs can be solved efficiently (see [12, 5]), for example by employing the Online Active Set Strategy, which is implemented in the open-source QP solver qpOASES [13].

In the special, yet practically relevant, case where  $H_k$  is a diagonal matrix and  $D_k$  is an identity matrix (i.e., only bounds on states and controls exist) the optimal solution  $z_k^*$  can conveniently be computed by component-wise “clipping” of the unconstrained solution as it was presented in [14]:

$$z_k^* = \max(\underline{d}_k, \min(H_k^{-1} m_k, \bar{d}_k)). \quad (7)$$

### 3.2 Structure of the Newton System

The right hand side vector  $\mathcal{G} : \mathbb{R}^{Nn_x} \rightarrow \mathbb{R}^{Nn_x}$  of Newton system (6) is easily seen to only depend on two neighboring stages in each block  $\lambda_k$ . It holds that

$$\mathcal{G}(\lambda) := \left( \frac{\partial f^*}{\partial \lambda}(\lambda) \right)^\top = \begin{bmatrix} \frac{\partial f_0^*}{\partial \lambda_1}^\top + \frac{\partial f_1^*}{\partial \lambda_1}^\top \\ \frac{\partial f_1^*}{\partial \lambda_2}^\top + \frac{\partial f_2^*}{\partial \lambda_2}^\top \\ \vdots \\ \frac{\partial f_{N-1}^*}{\partial \lambda_N}^\top + \frac{\partial f_N^*}{\partial \lambda_N}^\top \end{bmatrix}(\lambda). \quad (8)$$

The left-hand side Newton matrix  $\mathcal{M} : \mathbb{R}^{Nn_x} \rightarrow \mathbb{R}^{Nn_x \times Nn_x}$  has a block tri-diagonal structure, as only neighboring multipliers  $\lambda_k, \lambda_{k+1}$  can have a joint contribution to  $f^*$ . At a fixed  $\lambda$  it holds

$$\mathcal{M}(\lambda) := -\frac{\partial^2 f^*}{\partial \lambda^2}(\lambda) = \begin{bmatrix} W_1 & U_1 & & & \\ U_1^\top & W_2 & \ddots & & \\ & \ddots & \ddots & U_{N-1} & \\ & & & U_{N-1}^\top & W_N \end{bmatrix}(\lambda), \quad (9)$$

where the diagonal and off-diagonal block components are given by

$$W_k(\lambda) := -\frac{\partial^2 f^*}{\partial \lambda_k^2}(\lambda) \quad \text{and} \quad U_k(\lambda) := -\frac{\partial^2 f^*}{\partial \lambda_k \partial \lambda_{k+1}}(\lambda). \quad (10)$$

### 3.3 Gradient and Hessian computation

**Lemma 4** (cf. [4, App. C]) *Let all (QP<sub>k</sub>) be feasible. Then the derivative of  $f_k^*$  with respect to the dual variables  $\lambda$  exists and is given by*

$$\begin{bmatrix} \frac{\partial f_k^*}{\partial \lambda_k} & \frac{\partial f_k^*}{\partial \lambda_{k+1}} \end{bmatrix} = z_k^{*\top} \begin{bmatrix} -E_k \\ C_k \end{bmatrix}^\top + \begin{bmatrix} 0 \\ c_k \end{bmatrix}^\top. \quad (11)$$

*Proof* The derivative  $\frac{\partial f_k^*}{\partial \lambda}$  exists by Lemma 3. We derive a closed form by regarding the stage QP Lagrangian

$$\mathcal{L}_k(z_k, \mu_k; \lambda) := \frac{1}{2} z_k^\top H_k z_k + m_k(\lambda)^\top z_k + p_k(\lambda) + \mu_k^\top \begin{bmatrix} D_k z_k - \underline{d}_k \\ \underline{d}_k - D_k z_k \end{bmatrix}.$$

Since  $H_k \succ 0$  and  $(\text{QP}_k)$  is feasible by assumption, it holds that  $(\text{QP}_k)$  has a (finite) optimal primal and dual solution  $(z_k^*, \mu_k^*)$ , and, by Danskin's Theorem [9], we can interchange optimization and derivation in the sense that

$$\frac{\partial f_k^*}{\partial \lambda} = \frac{\partial}{\partial \lambda} \mathcal{L}_k(z_k^*, \mu_k^*; \lambda)$$

holds. We then have

$$\begin{aligned} \frac{\partial f_k^*}{\partial \lambda} &= \frac{\partial \mathcal{L}_k(z_k^*, \mu_k^*; \lambda)}{\partial \lambda} + \frac{\partial \mathcal{L}_k(z_k^*, \mu_k^*; \lambda)}{\partial z_k^*} \cdot \frac{\partial z_k^*}{\partial \lambda} + \frac{\partial \mathcal{L}_k(z_k^*, \mu_k^*; \lambda)}{\partial \mu_k^*} \cdot \frac{\partial \mu_k^*}{\partial \lambda} \\ &= \frac{\partial \mathcal{L}_k(z_k^*, \mu_k^*; \lambda)}{\partial \lambda} + \frac{\partial \mathcal{L}_k(z_k^*, \mu_k^*; \lambda)}{\partial z_k^*} \cdot \frac{\partial z_k^*}{\partial \lambda} + \begin{bmatrix} D_k z_k^* - \underline{d}_k \\ \underline{d}_k - D_k z_k^* \end{bmatrix}^\top \cdot \frac{\partial \mu_k^*}{\partial \lambda} \\ &= \left( z_k^{*\top} \begin{bmatrix} -E_k \\ C_k \end{bmatrix}^\top + \begin{bmatrix} 0 \\ c_k \end{bmatrix}^\top \right) + 0 \cdot \frac{\partial z_k^*}{\partial \lambda} + 0, \end{aligned}$$

where the second and the third term vanish due to the stationarity and due to the complementarity requirement of the optimal stage solution  $(z_k^*, \mu_k^*)$ .  $\square$

*Remark 5* We can see from Lemma 4 that  $\|\mathcal{G}(\lambda)\|$  is indeed a measure for both, stationarity of  $f^*(\lambda)$ , and infeasibility of (P2), as claimed in Section 2.3.

The second derivative of  $f^*$  can be computed as follows:

**Lemma 5** *Let  $Z_k^* \in \mathbb{R}^{n_z \times (n_z - n_{\text{act}})}$ ,  $k \in \mathcal{S}$  (where  $n_{\text{act}}$  denotes the number of active constraints) be a basis matrix for the nullspace of  $\mathcal{A}_k^*(z_k^*(\lambda))$ , the optimal active set of  $(\text{QP}_k)$  at  $\lambda$ , and let  $P_k^* := Z_k^* (Z_k^{*\top} H_k Z_k^*)^{-1} Z_k^{*\top} \in \mathbb{R}^{n_z \times n_z}$  denote the elimination matrix for this nullspace. Then  $\mathcal{M}(\lambda)$  is given by*

$$\mathcal{M}(\lambda) = \mathcal{C} \mathcal{P} \mathcal{C}^\top,$$

where  $\mathcal{P} := \text{diag}([P_0^* \ P_1^* \ \dots \ P_N^*])$  and

$$\mathcal{C} := \begin{bmatrix} C_0 & -E_1 & & & \\ & C_1 & -E_2 & & \\ & & & \ddots & \\ & & & & C_{N-1} & -E_N \end{bmatrix} \in \mathbb{R}^{N n_x \times (N+1) n_z}.$$

*Proof* We compute the Hessian blocks in (9) explicitly. Differentiating (11) once more with respect to  $\lambda$ , we obtain

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

and

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

Within a fixed active set the optimal solution of (QP<sub>k</sub>) at  $\lambda$  is given by (cf., e.g., [28])

$$z_k^*(\lambda) = -P_k^{*-1} m_k(\lambda) = -P_k^{*-1} (g_k + E_k^\top \lambda_k + C_k^\top \lambda_{k+1}).$$

Accordingly, the Hessian blocks (cf. Equation (10)) are computed as

$$U_k = -E_k P_k^* C_k^\top \tag{12}$$

and

$$W_k = C_{k-1} P_{k-1}^* C_{k-1}^\top + E_k P_k^* E_k^\top, \tag{13}$$

which concludes the proof.  $\square$

*Remark 6* Under the assumption of LICQ, a nullspace basis matrix  $Z_k^*$  can be constructed as  $Z_k^* := \begin{bmatrix} -B^{-1}N \\ I \end{bmatrix}$  by partitioning the matrix of active constraint rows in the solution  $z_k^*(\lambda)$ ,  $D_k^*|_{\text{act}} =: [B \ N] \in \mathbb{R}^{n_{\text{act}} \times n_z}$ , into an invertible matrix  $B \in \mathbb{R}^{n_{\text{act}} \times n_{\text{act}}}$  (w.l.o.g. the first  $n_{\text{act}}$  columns by reordering) and a  $n_{\text{act}} \times (n_z - n_{\text{act}})$  matrix  $N$  (see [28, 19] for details).

*Remark 7* It is important to note that  $P_k^*$  can be obtained relatively cheaply from a null-space QP solver like qpOASES [13] that directly provides  $Z_k^*$  and a Cholesky factor  $R$  for  $R^\top R = Z_k^{*\top} H_k Z_k^*$ , see [12]. For the special case of diagonal Hessian matrices  $H_k = \text{diag}(h_k^1, \dots, h_k^{n_z})$  and simple bounds, the projection  $P_k^*$  is simply a diagonal matrix with either  $1/h_k^i$  or 0 entries depending on whether the corresponding variable bound is inactive or active. The calculation of the Hessian blocks can then be accelerated even further using diadic products as proposed in [14].

### 3.4 Solution of the Newton system and regularization

By Lemma 3,  $\mathcal{M}(\lambda)$  is positive semidefinite, as  $f^*(\lambda)$  is concave. The block-tridiagonal structure of the  $\mathcal{M}(\lambda)$ , cf. Equation (9), can be exploited for the efficient solution of the Newton system (6). Observing that a lower triangular factor  $L$  of  $\mathcal{M}(\lambda) = LL^\top$  possesses the same structural zero-blocks below the diagonal, we suggest to employ a banded Cholesky decomposition. This

factorization differs from a regular Cholesky decomposition (see, e.g., [28]) by skipping all redundant blocks left and below the subdiagonal block  $U_k^\top$  of each block column  $k$ , thus reducing the computational complexity from  $O(N^3 n_x^3)$  to  $O(N n_x^3)$  floating point operations (FLOPs).

In the case of jointly redundant active constraints in several (QP $_k$ ) via the stage coupling constraints (P2),  $\mathcal{M}(\lambda)$  may become rank-deficient [26]. We propose to overcome this by applying regularization. In the software package qpDUNES, where we implemented the dual Newton strategy (see also Section 7), both a Levenberg-Marquadt-type regularization and a “on-the-fly” regularization are available. While the former one uses

$$\tilde{\mathcal{M}}(\lambda^i) := \mathcal{M}(\lambda^i) + \gamma \cdot I \quad (14)$$

with a (small) constant regularization parameter  $\gamma \in \mathbb{R}^+$  instead of  $\mathcal{M}(\lambda^i)$  in the Newton system (6) on detection of singularity during the initial banded Cholesky factorization, the latter only regularizes those diagonal elements for which the crucial division step in the Cholesky decomposition cannot be performed due to singularity (similarly to the modified Cholesky factorization described in [28], based on [18]). We note that the “on-the-fly” regularization ensures positive definiteness of the resulting  $\tilde{\mathcal{M}}(\lambda^i)$  (as it has a unique Cholesky decomposition) and avoids the need of restarting the factorization, but may numerically be less stable.

### 3.5 A reverse Cholesky factorization for improved stability

Particularly in the context of MPC, one expects rather many active constraints in the beginning of the control horizon, and few to none towards the end of the horizon. We aim to exploit this knowledge by applying a Cholesky factorization to  $\mathcal{M}$  (we omit the  $\lambda$ -dependency in this section for notational convenience) in reverse order, i.e., starting from the last row/column, as detailed in Algorithm 2. Instead of a factorization  $\mathcal{M} = LL^\top$ , we obtain a Cholesky-like factorization  $\mathcal{M} = \mathcal{R}\mathcal{R}^\top$ , that is equally well suited for an efficient solution of the Newton system (6). To see this, observe that Algorithm 2 is equivalent to a standard Cholesky factorization applied to  $\hat{\mathcal{M}} := \mathcal{J}\mathcal{M}\mathcal{J}^\top$  after a full row and column

permutation through  $\mathcal{J} := \begin{bmatrix} & & 1 \\ & \cdot & \\ 1 & & \end{bmatrix}$ . The advantage of applying this reverse

Cholesky factorization in the dual Newton strategy is twofold. First, observe that a diagonal block  $W_k$  only changes from one Newton iteration to the next if the active set on stage  $k$  or stage  $k - 1$  changes, and an off-diagonal block  $U_k$  only changes if the active set on stage  $k$  changes (in particular note that  $\mathcal{M}$  only needs to be recomputed in blocks with active set changes). As Algorithm 2 only uses data from the last  $k$  block rows (and columns) in block iteration  $k$ , it is sufficient to restart the factorization from the block row that corresponds to the last active set change. Furthermore, we can also expect better numerical properties of  $R$ , as the principal submatrix corresponding to stages without

**Algorithm 2:** Structure-exploiting reverse Cholesky factorization

---

**Input:** Newton Hessian matrix  $\mathcal{M}$   
**Output:** Cholesky-like factor  $R$  for  $\mathcal{M} = \mathcal{R}\mathcal{R}^\top$

```

1 for  $k = N : 1$  do                               /* go by block columns */
2   for  $j = k \cdot n_x : (k - 1) \cdot n_x + 1$  do /* go by columns */
3      $w = \mathcal{M}_{jj}$ 
4      $\bar{l} = \min(N \cdot n_x, (k + 1) \cdot n_x)$  /* end of row fill in */
5     for  $l = j + 1 : \bar{l}$  do                       /* subtract row tail */
6        $w = w - \mathcal{R}_{jl}^2$ 
7      $\mathcal{R}_{jj} = \sqrt{w}$ 
8      $\bar{i} = \max(1, (k - 2) \cdot n_x + 1)$  /* end of column fill in */
9     for  $i = j - 1 : \bar{i}$  do                       /* write rest of column */
10       $w = \mathcal{M}_{ij}$ 
11      if  $i > (k - 1) \cdot n_x$  then                /* end of row fill in */
12         $\bar{l} = \min(N \cdot n_x, (k + 1) \cdot n_x)$ 
13      else
14         $\bar{l} = \min(N \cdot n_x, k \cdot n_x)$ 
15      for  $l = j + 1 : \bar{l}$  do                       /* subtract row tail */
16         $w = w - \mathcal{R}_{jl} \cdot \mathcal{R}_{il}$ 
17       $\mathcal{R}_{ij} = w / \mathcal{R}_{jj}$ 

```

---

active state constraints is positive definite (recall that rank-deficiency of  $\mathcal{M}$  can only arise from a redundancy in active stage constraints over several stages) and of similar conditioning as the original problem; a significant worsening of the conditioning can only appear in block-rows with active stage constraints, which, in a typical MPC setting, tend to appear rather on the earlier than on the later stages, and thus enter later in Algorithm 2 compared to the standard Cholesky factorization.

To formalize this, we identify the reverse Cholesky factorization with the discrete time Riccati recursion in the following. Let us regard the (possibly regularized) Newton Hessian  $\mathcal{M} \succ 0$  in block form as defined in Equation (9). Then, the reverse Cholesky factorization (Algorithm 2) is easily seen to be given by the recursion

$$\begin{aligned} X_{k-1} &= W_{k-1} - U_{k-1} \cdot X_k^{-1} \cdot U_{k-1}^\top \\ X_N &= W_N, \end{aligned} \quad (15)$$

where the Cholesky factor  $\mathcal{R}$  in block form is given by

$$\mathcal{R} = \begin{bmatrix} \mathcal{R}_{1,1} & \mathcal{R}_{1,2} & & & \\ & \mathcal{R}_{2,2} & \ddots & & \\ & & \ddots & \mathcal{R}_{N-1,N} & \\ & & & & \mathcal{R}_{N,N} \end{bmatrix}$$

with upper triangular blocks  $\mathcal{R}_{k,k}$  given implicitly (but uniquely) by  $X_k =: \mathcal{R}_{k,k} \mathcal{R}_{k,k}^\top \forall i \in \mathcal{S}_0$  and dense blocks  $\mathcal{R}_{k,k+1} = U_k \cdot \mathcal{R}_{k+1,k+1} \forall i \in \mathcal{S}_{0,N}$ . Note that in this context the subscripts  $\mathcal{R}_{k,k}$  refer to the block entries of  $\mathcal{R}$  rather than to the individual entries (as used in Algorithm 2). We refer to  $X_k$  as Cholesky iterates in the following.

For LTI systems without active constraints it holds (cf. Lemma 5)

$$\begin{aligned} W_k &= CH^{-1}C^\top + EH^{-1}E^\top \\ U_k &= -EH^{-1}C^\top \end{aligned}$$

for  $k \in \mathcal{S}_{0,N}$ , where (analogously to QPs (1) and (P))  $C = \begin{bmatrix} A & B \end{bmatrix}$  are the dynamics,  $E = \begin{bmatrix} I & 0 \end{bmatrix}$  is the state selection matrix, and  $H = \begin{bmatrix} Q & S^\top \\ S & R \end{bmatrix}$  is the quadratic objective weight. Due to a possibly different choice of the Hessian on the last interval ( $H \equiv P$ ), it holds

$$W_N = CH^{-1}C^\top + P^{-1}. \quad (16)$$

**Theorem 6** *If  $P$  is the solution to the discrete time algebraic Riccati equation*

$$P = Q + A^\top P A - (S + A^\top P B) (R + B^\top P B)^{-1} (S^\top + B^\top P A),$$

*then the Cholesky iterates  $X$  are constant, i.e., recursion (15) is stationary. In particular, it holds  $X_k := P^{-1} + CH^{-1}C^\top = W_N \forall k \in \mathcal{S}_0$ .*

*Proof* The proof is done by calculation. We start from the the Cholesky recursion property, apply the assumed relation between Cholesky and Riccati iterates, and transform the expression into the form of the Riccati recursion. We have

$$\begin{aligned} X_{k-1} &= W_{k-1} - U_{k-1} X_k^{-1} U_{k-1}^\top \\ \Leftrightarrow P^{-1} + CH^{-1}C^\top &= CH^{-1}C^\top + EH^{-1}E^\top \\ &\quad - EH^{-1}C^\top (P^{-1} + CH^{-1}C^\top)^{-1} CH^{-1}E^\top \end{aligned}$$

and therefore

$$P^{-1} = EH^{-1}E^\top - EH^{-1}C^\top (P^{-1} + CH^{-1}C^\top)^{-1} CH^{-1}E^\top. \quad (17)$$

Using the Schur complement  $\bar{Q} = Q - S^\top R^{-1} S$  it is well known from elementary linear algebra that the inverse  $H^{-1}$  can be expressed by

$$H^{-1} = \begin{bmatrix} Q & S^\top \\ S & R \end{bmatrix}^{-1} = \begin{bmatrix} \bar{Q}^{-1} & -\bar{Q}^{-1} S^\top R^{-1} \\ -R^{-1} S \bar{Q}^{-1} & R^{-1} + R^{-1} S \bar{Q}^{-1} S^\top R^{-1} \end{bmatrix}.$$

Using this,  $C = \begin{bmatrix} A & B \end{bmatrix}$ , and the special structure of  $E = \begin{bmatrix} I & 0 \end{bmatrix}$ , we first see that the identities

$$\begin{aligned} EH^{-1}E^\top &= \bar{Q}^{-1}, \\ CH^{-1}E^\top &= (A - BR^{-1}S) \bar{Q}^{-1} =: \bar{C} \bar{Q}^{-1} \end{aligned}$$

and

$$\begin{aligned} CH^{-1}C^\top &= (A - BR^{-1}S) \bar{Q}^{-1} (A^\top - S^\top R^{-1}B^\top) + BR^{-1}B^\top \\ &= \bar{C} \bar{Q}^{-1} \bar{C}^\top + BR^{-1}B^\top \end{aligned}$$

hold. Thus, (17) can be written as

$$P^{-1} = \bar{Q}^{-1} - \bar{Q}^{-1} \bar{C}^\top (P^{-1} + BR^{-1}B^\top + \bar{C} \bar{Q}^{-1} \bar{C}^\top)^{-1} \bar{C} \bar{Q}^{-1}$$

Applying the Woodbury matrix identity with  $Y := P^{-1} + BR^{-1}B^\top$  we can express the right hand side term by

$$P^{-1} = (\bar{Q} + \bar{C}^\top Y^{-1} \bar{C})^{-1}$$

and thus

$$\begin{aligned} P &= \bar{Q} + \bar{C}^\top Y^{-1} \bar{C} \\ &= Q - S^\top R^{-1}S + (A^\top - S^\top R^{-1}B^\top) (P^{-1} + BR^{-1}B^\top)^{-1} (A - BR^{-1}S). \end{aligned}$$

holds. Note that the inverse matrices of  $Q, R, P$  and  $Y$  exist (and are real-valued), since we assumed QP (1) convex. Applying the Woodbury identity once again (however, in opposite direction) on  $(P^{-1} + BR^{-1}B^\top)^{-1}$ , and introducing  $\bar{R} := (R + B^\top PB)$ , we get

$$\begin{aligned} P &= Q - S^\top R^{-1}S \\ &\quad + (A^\top - S^\top R^{-1}B^\top) \left( P - PB (R + B^\top PB)^{-1} B^\top P \right) (A - BR^{-1}S) \\ &= Q - S^\top R^{-1}S \\ &\quad + A^\top (P - PB \bar{R}^{-1} B^\top P) A \\ &\quad \quad - S^\top R^{-1} B^\top (P - PB \bar{R}^{-1} B^\top P) A \\ &\quad \quad - A^\top (P - PB \bar{R}^{-1} B^\top P) BR^{-1}S \\ &\quad \quad + S^\top R^{-1} B^\top (P - PB \bar{R}^{-1} B^\top P) BR^{-1}S. \end{aligned} \tag{18}$$

Using the identity  $I = \bar{R} \bar{R}^{-1} = \bar{R}^{-1} \bar{R}$ , we further have

$$\begin{aligned} &S^\top R^{-1} B^\top (P - PB \bar{R}^{-1} B^\top P) BR^{-1}S - S^\top R^{-1}S = \\ &= S^\top R^{-1} B^\top PBR^{-1}S - S^\top R^{-1} B^\top PB \bar{R}^{-1} B^\top PBR^{-1}S - S^\top R^{-1}S \\ &= S^\top R^{-1} \bar{R} \bar{R}^{-1} B^\top PBR^{-1}S \\ &\quad - S^\top R^{-1} B^\top PB \bar{R}^{-1} B^\top PBR^{-1}S - S^\top \bar{R}^{-1} \bar{R} R^{-1}S \\ &= S^\top R^{-1} (R + B^\top PB) \bar{R}^{-1} B^\top PBR^{-1}S \\ &\quad - S^\top R^{-1} B^\top PB \bar{R}^{-1} B^\top PBR^{-1}S - S^\top \bar{R}^{-1} (R + B^\top PB) R^{-1}S \\ &= S^\top R^{-1} R \bar{R}^{-1} B^\top PBR^{-1}S + S^\top R^{-1} B^\top PB \bar{R}^{-1} B^\top PBR^{-1}S \\ &\quad - S^\top R^{-1} B^\top PB \bar{R}^{-1} B^\top PBR^{-1}S \\ &\quad - S^\top \bar{R}^{-1} R R^{-1}S - S^\top \bar{R}^{-1} B^\top PBR^{-1}S \\ &= -S^\top \bar{R}^{-1}S \end{aligned}$$

and

$$\begin{aligned}
& -A^\top (P - PB\bar{R}^{-1}B^\top P) B\bar{R}^{-1}S = \\
& = -A^\top P B\bar{R}^{-1}S + A^\top P B\bar{R}^{-1}B^\top P B\bar{R}^{-1}S \\
& = -A^\top P B\bar{R}^{-1}\bar{R}R^{-1}S + A^\top P B\bar{R}^{-1}B^\top P B\bar{R}^{-1}S \\
& = -A^\top P B\bar{R}^{-1}(R + B^\top PB)R^{-1}S + A^\top P B\bar{R}^{-1}B^\top P B\bar{R}^{-1}S \\
& = -A^\top P B\bar{R}^{-1}RR^{-1}S - A^\top P B\bar{R}^{-1}B^\top P B\bar{R}^{-1}S + A^\top P B\bar{R}^{-1}B^\top P B\bar{R}^{-1}S \\
& = -A^\top P B\bar{R}^{-1}S.
\end{aligned}$$

Analogously it holds

$$-S^\top R^{-1}B^\top (P - PB\bar{R}^{-1}B^\top P) A = -S^\top \bar{R}^{-1}B^\top P A.$$

Therefore (18) is equivalent to

$$\begin{aligned}
P &= Q - S^\top \bar{R}^{-1}S \\
&\quad - S^\top \bar{R}^{-1}B^\top P A - A^\top P B\bar{R}^{-1}S + A^\top (P - PB\bar{R}^{-1}B^\top P) A \\
&= Q + A^\top P A - (S^\top + A^\top PB) \bar{R}^{-1} (S + B^\top P A),
\end{aligned}$$

which concludes the proof.  $\square$

This proof is easily seen to also extend to the LTV case without active constraints, where we have (cf. Lemma 5):

$$\begin{aligned}
W_k &= C_{k-1}H_{k-1}^{-1}C_{k-1}^\top + E_kH_k^{-1}E_k^\top & \forall k \in \mathcal{S}_{0,N} \\
U_k &= -E_kH_k^{-1}C_k^\top & \forall k \in \mathcal{S}_{0,N} \\
W_N &= C_{N-1}H_{N-1}^{-1}C_N^\top + H_N^{-1},
\end{aligned}$$

with  $C_k = [A_k \ B_k]$ ,  $\forall k \in \mathcal{S}_N$ ,  $E_k = [I \ 0]$ ,  $\forall k \in \mathcal{S}_N$ ,  $H_k = \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix}$ ,  $\forall k \in \mathcal{S}_N$ , and  $H_N = Q_N$ .

**Corollary 7** *Let  $W_k$ ,  $k \in \mathcal{S}_0$  and  $U_k$ ,  $k \in \mathcal{S}_{0,N}$  be computed from an LTV system without (active) state constraints. Then, the Cholesky iterates  $X_k$ ,  $k \in \mathcal{S}_0$  from the Cholesky recursion (15) can be identified with the discrete time time-varying algebraic Riccati recursion*

$$P_N = Q_N \tag{19a}$$

$$\begin{aligned}
P_{k-1} &= Q_{k-1} + A_{k-1}^\top P_k A_{k-1} & (19b) \\
&\quad - (S_{k-1} + A_{k-1}^\top P_k B_{k-1}) (R_{k-1} + B_{k-1}^\top P_k B_{k-1})^{-1} \\
&\quad \quad \quad (S_{k-1}^\top + B_{k-1}^\top P_k A_{k-1})
\end{aligned}$$

via  $X_k = P_k^{-1} + C_{k-1}H_{k-1}^{-1}C_{k-1}^\top \quad \forall k \in \mathcal{S}_0$ .



*Proof* The proof follows exactly the lines of the proof to Theorem 6, yet keeping the matrix block indices. In particular, one has

$$P_{k-1}^{-1} = E_{k-1} H_{k-1}^{-1} E_{k-1}^\top - E_{k-1} H_{k-1}^{-1} C_{k-1}^\top (P_k^{-1} + C_{k-1} H_{k-1}^{-1} C_{k-1}^\top)^{-1} C_{k-1} H_{k-1}^{-1} E_{k-1}^\top$$

in place of (17) and transforms it into (19b) using the same matrix identities.  $\square$

### 3.6 Choice of the Newton step size

Due to the piecewise quadratic nature of  $f^*(\lambda)$ , a globalization strategy is needed. For computational efficiency close to the solution, where we assume the quadratic model of the dual function  $f^*(\lambda)$  to be accurate, we propose to employ a line search technique to find an (approximate) solution to

$$\alpha^i := \arg \max_{0 \leq \alpha \leq 1} f(\lambda^i + \alpha \Delta \lambda^i) \quad (20)$$

In contrast to general nonsmooth optimization, an exact line search is possible at reasonable cost in our context. In particular,  $f^*(\lambda^i + \alpha \Delta \lambda)$  is a one-dimensional piecewise quadratic function along the search direction  $\Delta \lambda$ . An exact quadratic model in search direction can be built up evaluating each  $f_k^*(\lambda^i + \alpha \Delta \lambda)$  at each value of  $\alpha \in [0, 1]$  that corresponds to an active set change together with the slope information given by  $\frac{\partial f_k^*(\lambda^i + \alpha \Delta \lambda)}{\partial \alpha} = \frac{\partial f_k^*}{\partial \lambda}(\lambda^i + \alpha \Delta \lambda) \cdot \Delta \lambda$ . The gradient in each active set change can be evaluated cheaply from Equation (11) with the current  $z_k^*(\lambda^i + \alpha \Delta \lambda)$ .

Note that a parametric active set strategy like qpOASES traverses these points  $z_k^*(\lambda^i + \alpha \Delta \lambda)$  for values of  $\alpha$  corresponding to active-set changes naturally while computing  $z_k^*(\lambda^i + \Delta \lambda)$  for the full step  $\Delta \lambda$  (cf. [12]).

When employing the clipping operation (7) for the solution of the stage problems (QP<sub>k</sub>), the points of active-set changes along the search direction can analogously be determined by a simple ratio test.

For a practical implementation, also heuristic backtracking-based search strategies seem appropriate in this context. Particularly the fact that the gradient evaluation  $\mathcal{G}(\lambda)$  comes almost at the same cost as a function evaluation  $f^*(\lambda)$  can be exploited within the line search. A search strategy that seemed to perform particularly well in practice was a combination of a fast backtracking line search, allowing to quickly detect very small step sizes, with a bisection interval search for refinement.

While we make use of a backtracking line search to quickly decrease the maximum step size  $\alpha_{\max}$ , the minimum step size  $\alpha_{\min}$  is given by the minimum scaling of the search direction that leads to an active set change on any stage. While this is intuitively clear, as each region with a constant active set is quadratic and Newton's method takes a step towards the minimum of a local quadratic function approximation, we give a formal proof for this in the

following section, in the context of convergence (Lemma 8). This guaranteed minimum step size is indicated by  $\lambda_{\alpha_{\min}}^i$  in Figure 1.

*Remark 8* We obtain all  $\alpha$ -values at which active set changes occur at no extra cost when employing an online active set strategy to solve each  $(\text{QP}_k)$ . Therefore, taking the minimum over all these  $\alpha$ -values over all stages  $k \in \mathcal{S}$  provides us with a lower bound for  $\alpha_{\min}$ . If the solution to  $(\text{QP}_k)$  is computed by Equation (7), points of active-set changes can still be obtained cheaply by comparing the unconstrained to the clipped solution in each component.

#### 4 Finite convergence of the algorithm

Convergence of non-smooth Newton methods has been proven before for functions with similar properties [29, 26, 17]. In the following we show convergence in the specific setting present in this paper; this allows us to present a shorter proof, that (in our opinion) is easier to follow. Based on these results we further establish an infeasibility detection mechanism in Section 5, which according to our knowledge is novel to QP solvers based on nonsmooth Newton methods.

A bit of notation is needed throughout this and the following section. By  $\mathcal{M}(\lambda)$  we refer to the Hessian matrix of (D), as defined in Equations (9-10). The possibly regularized version of  $\mathcal{M}(\lambda)$  used in the solution step of the Newton system (6) is denoted by  $\tilde{\mathcal{M}}(\lambda)$ . We will omit the dependency on  $\lambda$  occasionally for notational convenience when it is clear from the context. The active set of stage constraints at  $\lambda$  is denoted by  $\mathcal{A}^*(z^*(\lambda))$ .

We start with a rather obvious result, that nonetheless is crucial for the practical performance of the Dual Newton Strategy.

**Lemma 8 (Local one-step convergence)** *Let (P) be feasible. Let  $\lambda^i$  be the current dual iterate in Algorithm 1. Let  $\mathcal{M}(\lambda^i)$  be positive definite, i.e., no regularization is needed during Step 7 in Algorithm 1, and let  $\Delta\lambda$  be the solution of the Newton equation (6). Then, if  $\mathcal{A}^*(z^*(\lambda^i)) = \mathcal{A}^*(z^*(\lambda^i + \Delta\lambda))$ , it holds that  $\lambda^{i+1} := \lambda^i + \Delta\lambda$  solves Problem (D). In particular it holds  $\operatorname{argmax}_{0 \leq \alpha \leq 1} f^*(\lambda^i + \alpha\Delta\lambda) = 1$ .*

*Proof* Recall that  $f^*$  is piecewise quadratic in  $\lambda$  by Lemma 3. By the construction in Lemma 5 we know that  $\mathcal{M}(\lambda^i)$  is constant within each region  $A(\lambda^i)$ , since the active set is fixed. By its definition, the Newton step  $\Delta\lambda$  points to the maximum of the quadratic function characterizing  $A(\lambda^i)$ . By concavity of  $f^*$  it follows that  $\lambda^i + \Delta\lambda$  has to be the maximum of  $f^*$  and thus solves (D). In particular  $\operatorname{argmax}_{0 \leq \alpha \leq 1} f^*(\lambda^i + \alpha\Delta\lambda) = 1$ .  $\square$

Lemma 8 is applied twofold in the dual Newton strategy. First, it allows us to make our line search smarter by only considering step sizes that lead to at least one active set change (or full steps) as mentioned above in Section 3.6. Second, it shows that once the correct region of the solution is identified we have a one-step convergence to the exact solution (up to numerical accuracy), cf. Section 2.4. Next, we show global convergence.

**Theorem 9 (Global convergence)** *Let (P) be feasible. Let  $\lambda^0 \in \mathbb{R}^{Nn_x}$  and let  $\lambda^i \in \mathbb{R}^{Nn_x}$  be defined recursively by  $\lambda^{i+1} := \lambda^i + \alpha^i \Delta \lambda^i$ , where  $\Delta \lambda^i$  is the (possibly regularized) solution to Equation (6), and  $\alpha^i$  is the solution to Equation (20). Then the sequence  $\{\lambda^i\}_{i \in \mathbb{N}_0} \subset \mathbb{R}^{Nn_x}$  converges to the unique maximum  $\hat{\lambda}$  with  $\mathcal{G}(\hat{\lambda}) = 0$ .*

*Proof* The sequence  $\{\lambda^i\}_{i \in \mathbb{N}_0}$  induces a sequence  $\{f^i := f^*(\lambda^i)\}_{i \in \mathbb{N}_0} \subseteq \mathbb{R}$ . By definition of the exact line search (20) it holds that  $f^{i+1} \geq f^i$ , i.e.,  $\{f^i\}_{i \in \mathbb{N}_0}$  is monotonously increasing. Since (P) is feasible,  $f^*(\lambda)$  is a bounded, concave function by Lemma 3 and duality theory. By the Bolzano-Weierstrass Theorem  $\{f^i\}_{i \in \mathbb{N}_0}$  thus converges to an accumulation point  $\hat{f}$ .

Due to monotonicity of  $\{f^i\}_{i \in \mathbb{N}_0}$  it holds that  $\{\lambda^i\}_{i \in \mathbb{N}_0}$  is contained in the superlevel set

$$\mathcal{F} := \{\lambda \in \mathbb{R}^{Nn_x} \mid f^*(\lambda) \geq f^*(\lambda^0)\},$$

which is compact since  $f^*(\lambda)$  is a bounded concave function. A convergent subsequence  $\{\lambda^{i^{(1)}}\} \subseteq \{\lambda^i\}_{i \in \mathbb{N}_0}$  therefore has to exist and its limit  $\hat{\lambda}$  fulfills  $f^*(\hat{\lambda}) = \hat{f}$  due to the induced monotonicity of  $f^*(\lambda^{i^{(1)}})$ .

What remains to show is that  $\hat{\lambda}$  indeed maximizes  $f^*(\lambda)$ , i.e.  $\mathcal{G}(\hat{\lambda}) = 0$ . Assume contrarily  $\mathcal{G}(\hat{\lambda}) \neq 0$ . Since  $\tilde{\mathcal{M}}(\lambda^i)$  is strictly positive definite and due to the fixed regularization even bounded away from 0 in norm, it holds that  $\hat{\Delta} \lambda = \tilde{\mathcal{M}}(\hat{\lambda})^{-1} \mathcal{G}(\hat{\lambda}) \neq 0$ , cf. Equation (6), is an ascent direction. Then  $\hat{\alpha} > 0$  holds for the solution of Equation (20), and by  $C^1$ -continuity of  $f^*(\lambda)$  we can conclude that there is a  $\delta > 0$  with

$$f^*(\hat{\lambda} + \hat{\alpha} \hat{\Delta} \lambda) \geq f^*(\hat{\lambda}) + \delta.$$

Since  $\{\lambda^{i^{(1)}}\}$  converges to  $\hat{\lambda}$ , an index  $\bar{i} \in \mathbb{N}$  exists, such that for all  $i^{(1)} \geq \bar{i}$  we have  $\Delta \lambda^{i^{(1)}}$  sufficiently close to  $\hat{\Delta} \lambda$  and  $\lambda^{i^{(1)}}$  close enough to  $\hat{\lambda}$  such that

$$f^*(\lambda^{i^{(1)}} + \alpha^{i^{(1)}} \Delta \lambda^{i^{(1)}}) \geq f^*(\lambda^{i^{(1)}} + \hat{\alpha} \Delta \lambda^{i^{(1)}}) \geq f^*(\hat{\lambda}) + \delta/2,$$

where the first inequality holds by the maximum property of the line search in each iteration, and the second one holds by continuity of  $f^*(\lambda)$ . This, however, would be a contradiction to  $\hat{\lambda}$  being an accumulation point of a monotonously increasing sequence, so  $\mathcal{G}(\hat{\lambda}) = 0$ , and our claim holds.  $\square$

**Lemma 10** *Let  $z^*(\lambda^*)$  be a feasible solution for (P), that fulfills the LICQ. Then  $\mathcal{M}(\lambda^*)$  is strictly positive definite.*

*Proof* From Lemma 5 we have that  $\mathcal{M}(\lambda^*) = \mathcal{C} \mathcal{P}(\lambda^*) \mathcal{C}^\top$ , where

$$\mathcal{P}(\lambda^*) = \mathcal{Z}^* (\mathcal{Z}^{*\top} \mathcal{H} \mathcal{Z}^*)^{-1} \mathcal{Z}^{*\top}$$

with  $\mathcal{Z}^* := \text{blockdiag}([Z_0^* \ Z_1^* \ \dots \ Z_N^*])$  and  $\mathcal{H} := \text{blockdiag}([H_0 \ H_1 \ \dots \ H_N])$ . As in Lemma 5, each  $Z_k^*$ ,  $k \in \mathcal{S}$  denotes a basis matrix for the active constraints in the solution of  $(QP_k)$ , in this context the solution given the subproblem parameter  $\lambda^*$ . Consider now

$$\lambda^\top \mathcal{M}(\lambda^*) \lambda = \lambda^\top \mathcal{C} \mathcal{Z}^* (\mathcal{Z}^{*\top} \mathcal{H} \mathcal{Z}^*)^{-1} \mathcal{Z}^{*\top} \mathcal{C}^\top \lambda. \quad (21)$$

Since  $\mathcal{H}$  is positive definite and  $\mathcal{Z}^*$ , being a block diagonal composition of basis matrices, has full row rank, we have  $\mathcal{Z}^{*\top} \mathcal{H} \mathcal{Z}^* \succ 0$ , and thus  $(\mathcal{Z}^{*\top} \mathcal{H} \mathcal{Z}^*)^{-1} \succ 0$ . Using equation (21), this implies  $\lambda^\top \mathcal{M}(\lambda^*) \lambda \geq 0$ .

Assume  $\lambda^\top \mathcal{M}(\lambda^*) \lambda = 0$ . Since  $(\mathcal{Z}^{*\top} \mathcal{H} \mathcal{Z}^*)^{-1} \succ 0$  this means  $\lambda^\top \mathcal{C} \mathcal{Z}^* = 0$  has to hold.

The columns of  $\mathcal{Z}^*$  however are linearly independent and span the nullspace of the active stage constraints, i.e., every vector from the nullspace of  $\mathcal{Z}^*$  can be expressed by a linear combination of active stage constraints. If now  $\lambda^\top \mathcal{C}$  lies in the nullspace of the active stage constraints this means there is a linear combination of active stage constraints that represents  $\lambda^\top \mathcal{C}$ , which is a linear combination of the stage coupling equality constraints. Since LICQ holds we can conclude that  $\lambda = 0$ , and thus  $\mathcal{M}(\lambda^*) \succ 0$  holds.  $\square$

**Corollary 11 (Finite termination of Algorithm 1)** *Let  $z^*(\lambda^*)$  be a feasible solution for (P) that fulfills the LICQ. Let  $\lambda^0, \lambda^1, \dots$  be computed from Algorithm 1 (in exact arithmetic). Then  $\{\lambda^i\}_{i \in \mathbb{N}_0}$  becomes stationary after finitely many iterations, i.e.,  $\exists \bar{i} : \lambda^i = \lambda^* \forall i \geq \bar{i}$ .*

*Proof* From Lemma 1 we know that  $z_k^*(\lambda)$  depends continuously on  $\lambda$ . If no stage constraints of  $(\text{QP}_k)$  are weakly active in  $\lambda^*$ , then  $\lambda^*$  lies in the strict interior of a region  $A^*$  (with  $\mathcal{M}(\lambda)$  constant on  $A^*$ ) in the dual  $\lambda$  space. According to Theorem 9 there is a finite iteration index  $\bar{i} < \infty$  with  $\lambda^{\bar{i}} \in A^*$ . Due to Lemma 10 we have that  $\mathcal{M}(\lambda)$  is non-singular on  $A^*$ , and Lemma 8 guarantees convergence in the next iteration.

If there are weakly active constraints in  $\lambda^*$  (i.e.,  $\lambda^*$  lies on the boundary between several, but a finite number of, nonempty regions  $A^{(j_1)}, \dots, A^{(j_n)}$  in Figure 1), then due to  $C^1$  continuity of  $f$ , each quadratic function defining  $f$  on  $A^{(j_i)}$ ,  $i \in \{1, \dots, n\}$  needs to have its maximum in  $\lambda^*$ . We can define a ball  $B_\epsilon(\lambda^*)$  of fixed radius  $\epsilon > 0$  around  $\lambda^*$  with the property that for every  $\lambda \in B_\epsilon(\lambda^*)$  the dual function  $f$  on a region  $A^{(j_i)}$  containing  $\lambda$  is again defined by a quadratic function having its maximum in  $\lambda^*$ . By the identical argument as before we can conclude finite termination of algorithm 1 due to Theorem 9, Lemma 10, and Lemma 8.  $\square$

## 5 Infeasibility handling

If the primal problem (P) is infeasible, two possible consequences for the dual (D) arise. If a stage  $k$  and a selection of stage constraints  $\{\underline{d}_k^i \leq D_k^i z_k \leq \bar{d}_k^i\}_{i \in \mathcal{I}_k}$ ,  $\mathcal{I}_k \subseteq \{1, \dots, n_d\}$  exists that cannot be satisfied by any  $z_k \in \mathbb{R}^{n_z}$ , then and only then the dual problem (D) is infeasible as well, as no choice of  $\lambda$  will render  $(\text{QP}_k)$  feasible (see also Remark 1).

Otherwise, if all  $(\text{QP}_k)$  are feasible and yet (P) is infeasible, we have the following Lemma:

**Lemma 12** *Let all  $(\text{QP}_k)$  be feasible. If (P) is infeasible, the dual function  $f^*(\lambda)$  is unbounded and there exists (at least) one region in  $\lambda$  space,  $\emptyset \neq A^{\text{inf}} \subseteq \mathbb{R}^{N_{n_x}}$ , with constant  $\mathcal{M}_{\text{inf}}$  on  $A^{\text{inf}}$  and*

- i)  $\mathcal{M}_{\text{inf}}$  singular, i.e.,  $\exists \lambda \neq 0 : \mathcal{M}_{\text{inf}}\lambda = 0$ ,
- ii)  $\forall \bar{f} \in \mathbb{R} \exists \hat{\lambda} \in A^{\text{inf}} : f^*(\hat{\lambda}) > \bar{f}$ ,
- iii) for all  $A^{(j)}$ , defined by a representative  $\lambda^{(j)}$ , with  $\mathcal{M}(\lambda^{(j)}) \succ 0$  it holds

$$\exists \hat{\lambda} \in A^{\text{inf}} \forall \bar{\lambda} \in A^{(j)} : f^*(\hat{\lambda}) > f^*(\bar{\lambda}).$$

*Proof* Since all  $(\text{QP}_k)$  are feasible,  $f^*(\lambda)$  exists and (D) is feasible. Thus, (D) has to be unbounded by duality theory.

Let  $\lambda^{(j)} \in A^{(j)}$  for any  $A^{(j)}$ . The mapping  $\lambda \rightarrow f^*(\lambda)$  is onto an interval that contains the half-open interval  $[f^*(\lambda^{(j)}), +\infty)$  since  $f^*(\lambda)$  is continuous and unbounded. Since there is only a finite number of regions by Definition 2, property (ii) holds.

Assume now (i) is violated, i.e.,  $\exists A^{(j)}$  with  $\mathcal{M}(\cdot) \succ 0$  on  $A^{(j)}$  and  $\forall \bar{f} \in \mathbb{R} \exists \lambda \in A^{(j)} : f^*(\lambda) > \bar{f}$ . Since  $\mathcal{M}(\cdot) \succ 0$  is constant on  $A^{(j)}$  we have that  $f^*(\lambda)$  is strictly, and even strongly concave on  $A^{(j)}$ . Therefore  $\exists \bar{f} < \infty$  with  $f^*(\lambda) \leq \bar{f} \forall \lambda \in A^{(j)}$ , a contradiction. Therefore (i) holds.

In particular  $f^*(\lambda)$  is bounded (from above) on each  $A^{(j)}$  with  $\mathcal{M}(\cdot) \succ 0$ . Since there is only a finite number of regions, property (ii) implies (iii).  $\square$

Lemma 12 tells us that unboundedness of the dual objective function value can only occur in regions with singular Newton Hessian Matrix  $\mathcal{M}(\cdot)$ . We further characterize these unbounded regions in the following.

**Definition 3 (Infinite ray)** For  $\Delta\lambda \neq 0$  we call a pair  $(\bar{\lambda}, \Delta\lambda)$  an *infinite ray*, if there is a region  $A^{\text{inf}} \subseteq \mathbb{R}^{N_{n_x}}$  represented by  $\bar{\lambda} \in A^{\text{inf}}$  such that

1. the ray is contained in the region:  $\bar{\lambda} + \delta \cdot \Delta\lambda \in A^{\text{inf}} \quad \forall \delta > 0$ ,
2. the ray is in the nullspace of the region's Hessian:  $\mathcal{M}(\bar{\lambda})\Delta\lambda = 0$ ,
3. the ray is an ascent direction:  $\mathcal{G}(\bar{\lambda})^\top \Delta\lambda > 0$ .

Clearly every region containing an infinite ray is an unbounded region in the sense of Lemma 12. We additionally have the following characterizations.

*Remark 9* As  $\Delta\lambda$  lies in the nullspace of the negated dual Hessian  $\mathcal{M}(\cdot)$ , the dual gradient  $\mathcal{G}(\cdot)$  along a ray  $(\bar{\lambda}, \Delta\lambda)$  is constant.

**Lemma 13** Let  $(\bar{\lambda}, \Delta\lambda)$  be an infinite ray contained in  $A^{\text{inf}}$ . For every  $\tilde{\lambda} \in A^{\text{inf}}$  it holds  $\tilde{\lambda} + \gamma \cdot \Delta\lambda \in A^{\text{inf}} \quad \forall \gamma > 0$ , i.e.,  $(\tilde{\lambda}, \Delta\lambda)$  is an infinite ray as well.

*Proof* Since  $A^{\text{inf}}$  is convex by Lemma 2 and both  $\tilde{\lambda} \in A^{\text{inf}}$  and  $\tilde{\lambda} + \gamma \cdot \Delta\lambda \in A^{\text{inf}}$  for each choice of  $\gamma > 0$ , it holds

$$\beta \cdot \tilde{\lambda} + (1 - \beta) \cdot (\tilde{\lambda} + \gamma \cdot \Delta\lambda) \in A^{\text{inf}} \quad \forall \beta \in [0, 1]. \quad (22)$$

From Lemma 2 we also have that  $A^{\text{inf}}$  is polyhedral and therefore closed in  $\overline{\mathbb{R}}^{N_{n_x}}$ , where  $\overline{\mathbb{R}} := \mathbb{R} \cup \{-\infty, +\infty\}$ . Thus, the limit of (22) for  $\gamma \rightarrow \infty$  is contained in  $A^{\text{inf}}$ , and the claim holds.  $\square$

Lemma 13 is interesting as it tells us that  $A^{\text{inf}}$  has a cone- or beam-like shape. This will play a role in the following, when we characterize certificates for an unbounded dual problem. We have the following definition and theorem:

**Definition 4 (Ridge)** Let  $(\lambda^\dagger, \Delta\lambda^\dagger)$  be an infinite ray with  $\Delta\lambda^\dagger \neq 0$ . We call  $(\lambda^\dagger, \Delta\lambda^\dagger)$  a *ridge* if it holds  $\Delta\lambda^\dagger = \gamma \cdot \mathcal{G}(\lambda^\dagger)$  for a  $\gamma > 0$ , i.e., if  $\Delta\lambda^\dagger$  is aligned with the gradient of the dual function  $f^*$  along the ray it is defining.

**Theorem 14** *Let all  $(\text{QP}_k)$  be feasible. If  $(\text{P})$  is infeasible, then a ridge  $(\lambda^\dagger, \Delta\lambda^\dagger)$  exists.*

*Proof* From Lemma 12 we know that a region  $A^{\text{inf}}$  exists and is non-empty; we further know that  $f^*(\lambda)$  is an unbounded concave quadratic function on  $A^{\text{inf}}$ . An infinite ray  $(\tilde{\lambda}, \Delta\lambda)$  in the sense of Definition 3 therefore has to exist in  $A^{\text{inf}}$ . From Lemma 13 we have that in this case  $(\tilde{\lambda}, \Delta\lambda)$  is an infinite ray as well for every  $\tilde{\lambda} \in A^{\text{inf}}$ .

Let us regard  $A_{\cup}^{\text{inf}} := A^{\text{inf},1} \cup A^{\text{inf},2} \cup \dots \cup A^{\text{inf},n}$ , the union of all unbounded regions in the sense of Lemma 12. Clearly for all  $\bar{f} \in \mathbb{R}$  the superlevel set

$$\bar{A}_{\cup}^{\text{inf}} := \{\lambda \in A_{\cup}^{\text{inf}} \mid f^*(\lambda) \geq \bar{f}\} \quad (23)$$

is convex, as  $f^*(\lambda)$  is concave. Since  $f^*(\lambda)$  is continuous and unbounded on  $\bar{A}_{\cup}^{\text{inf}}$ , it holds that  $\bar{A}_{\cup}^{\text{inf}}$  is  $(Nn_x)$ -dimensional, i.e., full-dimensional; otherwise a directional vector  $e \in \mathbb{R}^{Nn_x}$  would exist with  $\tilde{\lambda} \in \text{int}(\bar{A}_{\cup}^{\text{inf}})$ , i.e.,  $f^*(\tilde{\lambda}) > \bar{f}$  and  $f^*(\tilde{\lambda} + \epsilon \cdot e) < \bar{f} \quad \forall \epsilon > 0$ , a violation of continuity.

Assume for the moment that every region  $A^{\text{inf},j}$  only contains one infinite ray (up to translation and scaling). Consider the nonempty intersection of  $\bar{A}_{\cup}^{\text{inf}}$  with a  $(Nn_x - 1)$ -dimensional Hyperplane  $F \subset \mathbb{R}^{Nn_x}$ . If  $F \cap \bar{A}_{\cup}^{\text{inf}}$  does not contain a singular ray,  $f^*(\lambda)$  has to be bounded on  $F \cap \bar{A}_{\cup}^{\text{inf}}$ , as  $f^*(\lambda)$  is composed from only a finite number of concave quadratic functions. In particular  $f^*(\lambda)$  attains a maximum  $\hat{\lambda}$  somewhere in the intersection.

This maximum  $\hat{\lambda}$  is characterized by the fact that the dual gradient  $\mathcal{G}(\hat{\lambda})$  is orthogonal to  $F$  if  $\hat{\lambda} \in \text{int}(F \cap \bar{A}_{\cup}^{\text{inf}})$ . Since the dual  $\lambda$  space is  $(Nn_x)$ -dimensional  $\mathcal{G}(\hat{\lambda})/\|\mathcal{G}(\hat{\lambda})\|$  is uniquely defined by  $F$  and vice versa. Recall that  $(\hat{\lambda}, \Delta\lambda)$  is an infinite ray in every  $\hat{\lambda} \in \bar{A}_{\cup}^{\text{inf}}$  for one fixed  $\Delta\lambda$  as shown above. Since  $\mathcal{G}(\lambda)$  is continuous, varying (i.e., “rotating”)  $F$  eventually has to yield a  $\hat{\lambda}$  with  $\mathcal{G}(\hat{\lambda}) = \gamma\Delta\lambda$  for some  $\gamma > 0$ .

If now there is a region  $A_j^{\text{inf}}$  with more than one infinite ray (i.e., the directional vectors  $\Delta\lambda$  of the infinite rays span a space of dimensionality  $k > 1$ ) the same argument can be applied, but with a  $(Nn_x - k)$ -dimensional Hyperplane  $F \subset \mathbb{R}^{Nn_x}$ . In this case  $\mathcal{G}(\hat{\lambda})$  is not uniquely defined anymore, but lies in the normal space of  $F$ . By the same argument as above there has to be an  $F$  whose normal space coincides with the space spanned by the directional vectors  $\Delta\lambda$  of the infinite rays of  $A_j^{\text{inf}}$ , and thus again there is an infinite ray  $(\hat{\lambda}, \Delta\lambda)$  coinciding with the gradient  $\mathcal{G}(\hat{\lambda})$  at its base point  $\hat{\lambda}$ .  $\square$

*Remark 10* Let  $(\lambda^\dagger, \mathcal{G}(\lambda^\dagger)) \subseteq A^\dagger \subseteq \bar{A}_{\cup}^{\text{inf}}$  a ridge. Clearly,  $\mathcal{M}(\lambda^\dagger)$  is rank deficient (and constant on  $A^\dagger$ ) and thus regularized with  $\delta I \succ 0$  by Algorithm 1.

By Theorem 14 and Definitions 3 and 4 it holds  $\mathcal{M}(\lambda^\dagger)\mathcal{G}(\lambda^\dagger) = 0$  and we have

$$\begin{aligned}\mathcal{G}(\lambda^\dagger) &= \frac{1}{\delta} \delta \mathcal{G}(\lambda^\dagger) + \frac{1}{\delta} \underbrace{\mathcal{M}(\lambda^\dagger) \mathcal{G}(\lambda^\dagger)}_{=0} = \frac{1}{\delta} \underbrace{(\mathcal{M}(\lambda^\dagger) + \delta I)}_{>0} \mathcal{G}(\lambda^\dagger) \\ \Leftrightarrow \tilde{\mathcal{M}}(\lambda^\dagger)^{-1} \mathcal{G}(\lambda^\dagger) &= \frac{1}{\delta} \mathcal{G}(\lambda^\dagger) \\ \Leftrightarrow \Delta \lambda^\dagger &= \frac{1}{\delta} \mathcal{G}(\lambda^\dagger),\end{aligned}$$

i.e., Algorithm 1 only performs gradient steps and thus remains on the ridge. Therefore we can see a ridge as the analogon to a fixed point in the case where the dual function  $f^*(\lambda)$  is unbounded.

**Lemma 15 (Minimality of the ridge gradient)** *Let  $(\lambda^\dagger, \Delta \lambda^\dagger)$  be a ridge. Then*

$$\|\mathcal{G}(\lambda^\dagger)\|_2 \leq \|\mathcal{G}(\lambda)\|_2 \quad (24)$$

for all  $\lambda \in \mathbb{R}^{Nn_x}$ . Inequality (24) is furthermore strict except for those  $\lambda$  that lie on a ridge, i.e., for which there is a ridge  $(\bar{\lambda}, \bar{\Delta} \bar{\lambda})$  such that  $\lambda = \bar{\lambda} + \gamma \bar{\Delta} \bar{\lambda}$  for a  $\gamma \geq 0$ . In particular the ridge  $(\lambda^\dagger, \Delta \lambda^\dagger) \subseteq A^\dagger$  is unique up to scaling of  $\Delta \lambda^\dagger$  and translations from the nullspace of  $\mathcal{M}(\lambda^\dagger)$ .

*Proof* Let  $\lambda \in \mathbb{R}^{Nn_x}$  and let  $(\lambda^\dagger, \Delta \lambda^\dagger)$  be a ridge. Due to concavity of  $f^*(\cdot)$  we have  $(\mathcal{G}(\lambda) - \mathcal{G}(\lambda^\dagger))^\top (\lambda^\dagger - \lambda) \geq 0$ . Since the  $\mathcal{G}(\cdot)$  is constant along the ridge,  $(\lambda^\dagger, \Delta \lambda^\dagger)$  and  $\Delta \lambda^\dagger$  and  $\mathcal{G}(\lambda^\dagger)$  only differ by a positive scalar factor, and also

$$(\mathcal{G}(\lambda) - \mathcal{G}(\lambda^\dagger))^\top \frac{1}{\gamma} (\lambda^\dagger + \gamma \mathcal{G}(\lambda^\dagger) - \lambda) \geq 0$$

holds for all  $\gamma > 0$ . Since concavity of  $f^*(\cdot)$  clearly also holds on the extended domain  $\bar{\mathbb{R}}^{Nn_x}$ , where  $\bar{\mathbb{R}} = \mathbb{R} \cup \{-\infty, +\infty\}$ , we have  $(\mathcal{G}(\lambda) - \mathcal{G}(\lambda^\dagger))^\top \mathcal{G}(\lambda^\dagger) \geq 0$  in the limit for  $\gamma \rightarrow \infty$ . This is equivalent to

$$\|\mathcal{G}(\lambda^\dagger)\|_2^2 = \mathcal{G}(\lambda^\dagger)^\top \mathcal{G}(\lambda^\dagger) \leq \mathcal{G}(\lambda)^\top \mathcal{G}(\lambda^\dagger) \leq \|\mathcal{G}(\lambda)\|_2 \cdot \|\mathcal{G}(\lambda^\dagger)\|_2,$$

where the last inequality is the Cauchy-Schwarz inequality. Thus (24) holds.

For the proof of the second part of the Lemma, we note that the Cauchy-Schwarz inequality is strict unless  $\mathcal{G}(\lambda)$  is a positive multiple of  $\mathcal{G}(\lambda^\dagger)$ , so  $\|\mathcal{G}(\lambda)\|_2 = \|\mathcal{G}(\lambda^\dagger)\|_2$  implies  $\mathcal{G}(\lambda) = \mathcal{G}(\lambda^\dagger)$ . Let us therefore consider an arbitrary  $\lambda \in \mathbb{R}^{Nn_x}$  with  $\mathcal{G}(\lambda) = \mathcal{G}(\lambda^\dagger)$ . Due to concavity we have

$$\begin{aligned}f^*(\beta \lambda + (1 - \beta) \lambda^\dagger) &\leq f^*(\lambda) + \mathcal{G}(\lambda)^\top (\beta \lambda + (1 - \beta) \lambda^\dagger - \lambda) \\ &= f^*(\lambda) + (1 - \beta) \cdot \mathcal{G}(\lambda)^\top (\lambda^\dagger - \lambda)\end{aligned} \quad (25)$$

for  $\beta \in [0, 1]$ , and analogously

$$f^*(\beta \lambda + (1 - \beta) \lambda^\dagger) \leq f^*(\lambda^\dagger) - \beta \cdot \mathcal{G}(\lambda^\dagger)^\top (\lambda^\dagger - \lambda). \quad (26)$$

We can multiply (25) by  $\beta$  and (26) by  $(1 - \beta)$ , add both inequalities up, and, using  $\mathcal{G}(\lambda) = \mathcal{G}(\lambda^\dagger)$ , yield

$$f^*(\beta\lambda + (1 - \beta)\lambda^\dagger) \leq \beta f^*(\lambda) + (1 - \beta) f^*(\lambda^\dagger)$$

Due to concavity of  $f^*(\cdot)$  also the converse inequality holds and we have

$$f^*(\beta\lambda + (1 - \beta)\lambda^\dagger) = \beta f^*(\lambda) + (1 - \beta) f^*(\lambda^\dagger), \quad (27)$$

i.e., linearity of  $f^*(\cdot)$  on the interval between  $\lambda$  and  $\lambda^\dagger$ . Once more since  $\mathcal{G}(\cdot)$  is constant along the ridge  $(\lambda^\dagger, \Delta\lambda^\dagger)$ , (27) holds analogously for all  $\tilde{\lambda}^\dagger := \lambda^\dagger + \gamma\Delta\lambda^\dagger$  in place of  $\lambda^\dagger$ , i.e.  $f^*$  is linear on the interval  $[\lambda, \lambda^\dagger + \gamma\Delta\lambda^\dagger]$  for all choices of  $\gamma > 0$ . Since the space of linear functions from  $\mathbb{R}^{N_{n_x}}$  to  $\mathbb{R}$  is closed, linearity also holds in the limit for  $\gamma \rightarrow \infty$ , which is the half-open interval  $\{\lambda + \gamma \cdot \Delta\lambda^\dagger \mid \gamma \in [0, \infty)\}$ . Since  $\mathcal{G}(\lambda) = \mathcal{G}(\lambda^\dagger)$  (which itself is a positive multiple of  $\Delta\lambda^\dagger$ ) we therefore have that  $(\lambda, \mathcal{G}(\lambda))$  is itself a ridge, which is moreover parallel to  $(\lambda^\dagger, \Delta\lambda^\dagger)$ .

Since  $f^*(\cdot)$  is specifically linear on  $[\lambda, \lambda^\dagger]$ ,  $(\lambda, \mathcal{G}(\lambda))$  differs from  $(\lambda^\dagger, \Delta\lambda^\dagger)$  only by a shift that lies in the nullspace of  $\mathcal{M}(\lambda^\dagger)$  as we claimed above.  $\square$

**Theorem 16 (Convergence to an infinite ray)** *If (P) is infeasible, then exactly one of the two following statements is true:*

1.  $(QP_k)$  is infeasible for at least one  $k \in \mathcal{S}$
2. Algorithm 1 converges to an infinite ray if the regularization parameter  $\delta$  is chosen sufficiently large

*Proof* Let (P) be infeasible. Due to the special time coupling structure of (P) either there exists a minimal infeasible set (a selection of constraints of minimal size that cannot be fulfilled at the same time) that is contained in the set of local stage constraints of one  $(QP_k)$ , or all minimal infeasible sets consist of local stage constraints (P3) of several stages  $k_1 < k_2 < \dots < k_n$  and the corresponding time coupling constraints between  $k_1$  and  $k_n$  (a subset of Constraints (P2)). In the former case Statement 1 holds, and infeasibility is detected by the stage QP solver on first execution. In the latter case we have that the partial dual function  $f^*(\lambda)$  exists and can be evaluated. In particular  $f^*(\lambda)$  is unbounded by Lemma 12. In the remainder of the proof we show that in this case Algorithm 1 indeed converges to an infinite ray.

As we have seen in the proof to Theorem 9 the iterates  $\lambda^i$  of Algorithm 1 defined by Equations (5), (6), and (20) induce a monotonously increasing sequence  $\{f^*(\lambda^i)\}_{i \in \mathbb{N}_0}$ . We claim that  $\{f^*(\lambda^i)\}_{i \in \mathbb{N}_0}$  does not converge if  $f^*(\lambda)$  is unbounded.

Assume contrarily that  $\{f^*(\lambda^i)\}_{i \in \mathbb{N}_0}$  converges. Clearly  $\mathcal{G}(\lambda)$  does not vanish since  $\mathcal{G}(\lambda) = 0$  would imply the existence of a feasible solution of (P) by Remark 5. By Lemma 15 we further know the existence of a

$$0 < G_{\min} := \min_{\lambda \in \mathbb{R}^{N_{n_x}}} \|\mathcal{G}(\lambda)\|. \quad (28)$$



If now  $\{f^*(\lambda^i)\}_{i \in \mathbb{N}_0}$  converges, clearly the updates  $\alpha^i \cdot \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i)$ , have to vanish. Since there are only finitely many different  $\tilde{\mathcal{M}}(\cdot) \succ 0$  and  $\mathcal{G}(\cdot)$  is bounded away from 0 by Equation (28) this implies that

$$\alpha^i = \arg \max_{0 \leq \alpha \leq 1} f^*(\lambda^i + \alpha \Delta \lambda)$$

with  $\Delta \lambda = \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i)$ , cf. Equation (20), has to drop to 0. At each iteration  $i$  three cases for  $\alpha^i$  could appear:

- i)  $\alpha^i = 1$ .
- ii)  $\alpha^i = 0$ . We have  $\mathcal{G}(\lambda^i) \neq 0$  and  $\tilde{\mathcal{M}}(\lambda^i) \succ 0$ , so  $\tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i)$  is an ascent direction. By  $C^1$ -continuity of  $f^*(\cdot)$  an ascent is possible and thus  $\alpha^i \neq 0 \quad \forall i \in \mathbb{N}_0$ .
- iii)  $0 < \alpha^i < 1$ . Then by maximality  $\alpha^i$  fulfills  $\mathcal{G}(\lambda^i + \alpha^i \Delta \lambda)^\top \Delta \lambda = \mathcal{G}(\lambda^i + \alpha^i \Delta \lambda)^\top \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i) = 0$ . Regard  $\phi(\alpha) := \mathcal{G}(\lambda^i + \alpha \Delta \lambda)^\top \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i)$  as a function in  $\alpha$ . Clearly  $\phi(0) = \mathcal{G}(\lambda^i)^\top \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i)$  is bounded away from 0, since there are only finitely many different  $\tilde{\mathcal{M}}(\cdot) \succ 0$  and  $\mathcal{G}(\cdot)$  is bounded away from 0 by Equation (28). Further  $\phi(\alpha)$  is continuous and its derivative  $\phi'(\alpha) = -\Delta \lambda^\top \tilde{\mathcal{M}}(\lambda^i + \alpha \Delta \lambda) \Delta \lambda$  is bounded from below (for a fixed  $\Delta \lambda$ ), again since there are only finitely many different  $\tilde{\mathcal{M}}(\cdot)$ ; note that the ascent direction  $\Delta \lambda$  cannot grow to infinity for  $i \rightarrow \infty$  since  $f^*(\lambda)$  is concave and  $f^*(\lambda^i)$  is increasing. Therefore  $\alpha^i$  has to be bounded away from 0 by a problem-data specific constant that is independent from the iteration index  $i$ .

Since also  $\alpha^i$  does not converge to 0 we have a contradiction, and our claim that  $\{f^*(\lambda^i)\}_{i \in \mathbb{N}_0}$  diverges (even monotonously) holds true. In particular every fixed function value  $\bar{f}$  will be exceeded and for every suitably large  $\bar{f}$  the iterates of Algorithm 1 will remain in  $\bar{A}_\cup^{\text{inf}}$  (as defined in Equation (23)) after a finite number of iterations.

Yet in the remainder of the proof we first show local and subsequently global attractiveness of a ridge in the sense of Theorem 14.

Let  $A^\dagger$  denote the region that contains a ridge  $(\lambda^\dagger, \mathcal{G}(\lambda^\dagger))$ . For two subsequent iterates both contained in  $A^\dagger$  (characterized by a constant Newton Matrix  $\mathcal{M}$ ) we have the exact linear model  $\mathcal{G}(\lambda^{i+1}) = \mathcal{G}(\lambda^i + \alpha^i \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i)) = \mathcal{G}(\lambda^i) - \alpha^i \mathcal{M} \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i)$ .

We now claim that  $\alpha^i = 1$  if  $\mathcal{G}(\lambda^{i+1}) \in A^\dagger$ , which is true by the following reasoning: by the argumentation in (ii) above, clearly  $\alpha^i > 0$ . If  $\alpha^i < 1$ , by optimality of the line search it would need to hold

$$\begin{aligned} & \mathcal{G}(\lambda^i + \alpha^i \Delta \lambda)^\top \Delta \lambda = 0 \\ \Leftrightarrow & \quad \left( \mathcal{G}(\lambda^i) - \alpha^i \mathcal{M} \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i) \right)^\top \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i) = 0 \\ \Leftrightarrow & \quad \mathcal{G}(\lambda^i)^\top \left( I - \alpha^i \mathcal{M} \tilde{\mathcal{M}}^{-1} \right)^\top \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i) = 0. \end{aligned}$$

Since  $\tilde{\mathcal{M}}^{-1} \succ 0$  and  $\mathcal{G}(\lambda^i) \neq 0$  by Equation (28) this could only be true if  $I - \alpha^i \mathcal{M} \tilde{\mathcal{M}}^{-1}$  is semidefinite or indefinite. However, we have  $I - \alpha^i \mathcal{M} \tilde{\mathcal{M}}^{-1} = (\tilde{\mathcal{M}} - \alpha^i \mathcal{M}) \tilde{\mathcal{M}}^{-1} = ((1 - \alpha^i) \mathcal{M} + \delta I) \tilde{\mathcal{M}}^{-1} \succ 0$  for  $\alpha^i < 1$  (recall that  $\mathcal{M}$  is singular on  $A^\dagger$  by assumption, hence  $\tilde{\mathcal{M}} = \mathcal{M} + \delta I$ ), a contradiction, and thus  $\alpha^i = 1$  holds.

To see local attractiveness of the ridge, consider the mapping

$$\begin{aligned} \psi(\mathcal{G}(\lambda^i)) &:= \|\mathcal{G}(\lambda^i) - \mathcal{G}(\lambda^{i+1})\| \\ &= \|\mathcal{G}(\lambda^i) - \tilde{\mathcal{M}} \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i) + \mathcal{M} \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i)\|. \end{aligned}$$

With  $\tilde{\mathcal{M}} = \mathcal{M} + \delta I$  we have

$$\begin{aligned} \psi(\mathcal{G}(\lambda^i)) &= \|\mathcal{G}(\lambda^i) - \delta \tilde{\mathcal{M}}^{-1} \mathcal{G}(\lambda^i)\| \\ &= \|(I - \delta \tilde{\mathcal{M}}^{-1}) \mathcal{G}(\lambda^i)\|. \end{aligned}$$

Clearly  $\tilde{\mathcal{M}}^{-1}$  is strictly positive definite, and  $I - \delta \tilde{\mathcal{M}}^{-1} = (\tilde{\mathcal{M}} - \delta I) \tilde{\mathcal{M}}^{-1} = \mathcal{M} \tilde{\mathcal{M}}^{-1}$  is positive semidefinite, so  $\psi(\cdot)$  is a contraction. Therefore  $\mathcal{G}(\lambda^i)$  converges for  $i \rightarrow \infty$  to a vector  $\mathcal{G}^*$  that fulfills  $\mathcal{M} \tilde{\mathcal{M}}^{-1} \mathcal{G}^* = 0$ , i.e., that leads to constant updates in Algorithm 1 that lie in the nullspace of  $\mathcal{M}$ . By Banach's fixed point theorem this fixed point is unique and therefore has to be aligned with the ridge  $(\lambda^\dagger, \mathcal{G}(\lambda^\dagger))$ . Note that due to the cone-like quadratic shape of  $A^\dagger$  (cf. Lemma 13) each subsequent iterate  $\lambda^{i+1}$  is contained in  $A^\dagger$  if  $\lambda^i \in A^\dagger$ .

It remains to show that the iterates  $\lambda^i$  of Algorithm (1) also converge globally to  $A^\dagger$ . To see this, we consider the auxiliary function

$$\tilde{f}(\lambda) := f^*(\lambda) - \mathcal{G}(\lambda^\dagger)^\top \lambda.$$

This function is clearly piecewise quadratic and convex, just like  $f^*$ , and its second derivative equals the second derivative of  $f^*$ , while the first derivative of  $\tilde{f}$  is given by  $\frac{\partial \tilde{f}}{\partial \lambda} = \mathcal{G}(\lambda)^\top - \mathcal{G}(\lambda^\dagger)^\top$ . It is furthermore bounded, since due to concavity we have for every  $\lambda \in \mathbb{R}^{N_{n_x}}$

$$\tilde{f}(\lambda) \leq \tilde{f}(\lambda^\dagger) + (\mathcal{G}(\lambda^\dagger) - \mathcal{G}(\lambda^\dagger))^\top (\lambda - \lambda^\dagger) = \tilde{f}(\lambda^\dagger).$$

The iterates  $\lambda^i$  of Algorithm (1) are increasing in  $\tilde{f}$  iff the step directions  $\tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i)$  are always ascent directions, i.e., if

$$\alpha^i (\mathcal{G}(\lambda^i) - \mathcal{G}(\lambda^\dagger))^\top \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i) \stackrel{!}{>} 0$$

holds. We have shown above that  $\alpha^i$  is bounded away from 0 by an iteration-independent constant and therefore we equivalently have the condition

$$\delta \mathcal{G}(\lambda^i)^\top \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i) \stackrel{!}{>} \delta \mathcal{G}(\lambda^\dagger)^\top \tilde{\mathcal{M}}(\lambda^i)^{-1} \mathcal{G}(\lambda^i), \quad (29)$$

where  $\delta$  is the regularization parameter in Algorithm 1. Since  $\delta \tilde{\mathcal{M}}(\lambda^i)^{-1}$  is positive definite, each of its finitely many distinct values  $\delta \tilde{\mathcal{M}}_{(j)}^{-1}$  defines a

scalar product that induces a norm on  $\mathbb{R}^{Nn_x}$ . Applying the Cauchy-Schwarz inequality to (29) we have that the iterates  $\lambda^i$  are increasing iff

$$\begin{aligned} \|G(\lambda^i)\|_{\delta\tilde{\mathcal{M}}_{(j)}^{-1}}^2 &\stackrel{!}{>} \|G(\lambda^\dagger)\|_{\delta\tilde{\mathcal{M}}_{(j)}^{-1}} \cdot \|G(\lambda^i)\|_{\delta\tilde{\mathcal{M}}_{(j)}^{-1}} \\ \Leftrightarrow \|G(\lambda^i)\|_{\delta\tilde{\mathcal{M}}_{(j)}^{-1}} &\stackrel{!}{>} \|G(\lambda^\dagger)\|_{\delta\tilde{\mathcal{M}}_{(j)}^{-1}}. \end{aligned} \quad (30)$$

While  $\lambda^i$  is not contained in a ridge, we have  $\|G(\lambda^i)\|_2 > \|G(\lambda^\dagger)\|_2$  from Lemma 15; furthermore  $I - \delta\tilde{\mathcal{M}}_{(j)}^{-1} = I - \delta(\mathcal{M}_{(j)} + \delta I)^{-1}$  vanishes for sufficiently large choices of  $\delta$ , fulfilling (30) and thus showing that the iterates  $\lambda^i$  are also ascending in the auxiliary function  $\tilde{f}$ . Since  $\tilde{f}$  is bounded, we can conclude that the region  $A^\dagger$ , containing a ridge, is eventually reached, thus concluding the proof of global convergence.  $\square$

Theorem 16 can algorithmically be used to detect infeasibility of (P). If any (QP<sub>k</sub>) is infeasible, it will be detected by the stage QP solver (either qpOASES, or during the clipping operation) on the first execution and (P) is immediately known to be infeasible. Otherwise, in the case of infeasibility through the coupling constraints, we know by Theorem 16 that the iterates of Algorithm 1 will eventually converge to a ridge. Here, we could gradually increase the regularization parameter  $\delta$ , if the iterates remain in regions with singular Hessians. A check whether  $\mathcal{G}(\lambda^i)$  is in the nullspace of  $\mathcal{M}(\lambda^i)$  (i.e.,  $\mathcal{M}(\lambda^i)\mathcal{G}(\lambda^i) \approx 0$ ) every few (regularized) iterations without an active-set change, combined with a check whether any active-set change occurs at all in the Newton direction will eventually conclude infeasibility. Note that the check for active-set changes in the Newton direction can cheaply be performed both in qpOASES and the clipping QP solver by simply considering the signs in the ratio test of the active and inactive constraints, cf. Section 3.6. Additionally, practical infeasibility can be concluded, when the objective function value exceeds a certain large threshold (caused by an explosion of the norm of the iterates  $\lambda^i$ ).

*Remark 11* In practice, the dual iterates  $\lambda^i$  in Algorithm 1 were indeed always observed to grow very fast and reach a ridge quickly in infeasible problems due to the initially small regularization.

We note that the theoretical result of Theorem 16 is somewhat unsatisfactory, since it requires a modification of the regularization parameter  $\delta$  (or even to only do gradient steps in regions with a singular Hessian). We initially aimed at proving Theorem 16 for generic regularized Newton steps, i.e., independent of  $\delta$ . Despite some considerable effort, we could not come up with a formal argument that links Condition (30) with Lemma 15, since the ordering relations in the Euclidian norm on the one hand and the norm induced by  $\delta\tilde{\mathcal{M}}_{(j)}^{-1}$  on the other hand might be different. Still, we are confident that also general regularized Newton updates, independent of the choice of  $\delta$ , are increasing in  $\tilde{f}$  on a global scale (though not necessarily monotonously), and thus formulate the following conjecture.

**Conjecture 17** *Theorem 16 holds independently for all choices of the regularization parameter  $\delta > 0$ .*

## 6 Concurrency in the Dual Newton Strategy

One important advantage of the Dual Newton Strategy is that, opposed to conventional active-set or interior-point methods, it is an easily parallelizable algorithm. Analyzing Algorithm 1 in this respect, we observe that all stage QPs can be solved concurrently in  $N + 1$  threads in Step 2. Next, each block of the dual gradient  $\mathcal{G}(\lambda)$  only depends on the solution of two neighboring stage QPs (cf. Equation 8), and therefore the setup can be done concurrently in  $N$  threads (Step 3). Also in the setup of the symmetric Newton matrix  $M$ , Step 6, each diagonal block only depends on the solution of two adjacent stage QP solutions, while each off-diagonal block only depends on the solution of one stage QP; therefore the workload of the setup of  $M$  can be distributed on  $N$  threads almost equally. During the line search procedure in Step 8 of Algorithm 1, the expensive steps in each iteration consist of solving all stage QPs for the new step size guess and computing the corresponding gradient, both of which can be spread over  $N + 1$  threads with almost equal workload.

Therefore, so far all steps of significant workload can be run fully in parallel in each major iteration of Algorithm 1, except for the solution of the structured linear system, Step 7. In a sequential implementation, solving the system based on a structure-exploiting reverse Cholesky factorization as proposed in Section 3.5 seems most efficient. This algorithm, however, is completely serial on the higher level and therefore cannot be expected to scale well with the number of available computing threads on modern and forthcoming multi-core CPU architectures. We therefore propose a parallelizable solution algorithm for the structured Newton system in Section 6.1, as an alternative to Algorithm 2. The overall cost of this algorithm is roughly twice as high in number of floating point operations (FLOPs) as Algorithm 2, but it comes at a parallel complexity of only  $\mathcal{O}(\log N)$  on  $N$  threads; therefore, the overall time complexity of one iteration of the dual Newton strategy comes down to only  $\mathcal{O}(\log N)$  FLOPs, compared to  $\mathcal{O}(N^3)$  in active-set methods, and  $\mathcal{O}(N)$  in an interior point method (to the best of our knowledge the best known complexity bound for tailored algorithms).

### 6.1 A parallel solution algorithm for the Newton system

The core idea behind the algorithm proposed in this section is to exploit the block-tridiagonal structure of the Newton matrix  $\mathcal{M}$  in Equation (6) in a cyclic reduction strategy. We note that a similar and to some extent more general algorithm has been proposed in [37] for the parallel solution of a linear system of similar structure, that originated from a related but slightly different problem. The work [37] contains a deeper analysis of the applicability and extensions of



and in general, we can recover  $\lambda_i$  from  $\lambda_{i-}$  and  $\lambda_{i+}$  by

$$\lambda_i = D_i^{-1} (g_i - U_{i-}^\top \lambda_{i-} - U_i \lambda_{i+}), \quad (35)$$

concurrently in reverse level order of the previous elimination procedure. Here  $\lambda_{i-}$  and  $\lambda_{i+}$  denote the  $\lambda$ -blocks preceding and succeeding  $\lambda_i$  in the system of equations remaining in the reduction step that eliminated  $\lambda_i$ .

The complete solution algorithm can be summarized as follows:

---

**Algorithm 3:** A parallel solution algorithm for Equation (6)

---

**Input:** Newton system given by  $g^{(0)} = [g_0^\top, \dots, g_N^\top]^\top$ ,  
 $D^{(0)} = \text{blockdiag}(D_0, \dots, D_N)$ ,  $U^{(0)} = \text{blockdiag}(U_0, \dots, U_{N-1})$

**Output:** Solution  $\lambda = [\lambda_0^\top, \dots, \lambda_N^\top]^\top$  to Equation (6)

```

1  $k_{\max} = \lceil \log_2(N-1) \rceil$ 
2 for  $k = 1 : k_{\max}$  do                                     /* factor step */
3   for  $i = 2^{k-1} : 2^k : N-1$  do in parallel
4      $i- = i - 2^{k-1}$ 
5      $i+ = \min(i + 2^k, N)$ 
6     compute  $D_i^{(k)}, U_i^{(k)}, g_i^{(k)}$  from Eq. (32) with
7        $D_\star = D_\star^{(k-1)}, U_\star = U_\star^{(k-1)}, g_\star = g_\star^{(k-1)} \quad \forall \star \in \{i, i-, i+\}$ 
7 Compute  $\lambda_0$  from Eq. (33) with  $\bar{U}_0 = U_0^{(k_{\max})}, \bar{g}_0 = g_0^{(k_{\max})}$ ,
    $\bar{D}_0 = D_0^{(k_{\max})}, \bar{D}_N = D_N^{(k_{\max})}$  using a Cholesky decomposition
8 Compute  $\lambda_N$  from Eq. (34) with
    $\bar{D}_N = D_N^{(k_{\max})}, \bar{g}_N = g_N^{(k_{\max})}, \bar{U}_0 = U_0^{(k_{\max})}$ 
9 for  $k = k_{\max} : -1 : 1$  do                                 /* solve step */
10  for  $i = 2^{k-1} : 2^k : N-1$  do in parallel
11  for
12    recover  $\lambda_i$  using Eq. (35) with
13     $D_i = D_i^{(k-1)}, g_i = g_i^{(k-1)}, U_{i-} = U_{i-}^{(k-1)}, U_i = U_i^{(k-1)}$ 

```

---

*Remark 12* Obviously the products of the block matrices  $U_\star$  and  $D_\star^{-1}$  in Equations (32) and (33) are most efficiently computed by a backsolve with a Cholesky factor of the diagonal blocks of the reduced size system,  $D_\star$ . If System (31) is linearly dependent, i.e., if  $\mathcal{M}$  is rank deficient, the Cholesky factorization of one of these blocks will fail (otherwise, i.e., if all  $D_\star$  have full rank, (33-35) would constitute an linear injective mapping  $\lambda \rightarrow \mathcal{G}$ ), and we can restart the factorization with a regularized  $\mathcal{M}$ , analogously to Section 3.4.

## 7 Open-source software implementation

The dual Newton strategy has been implemented in the open-source software package qpDUNES, which is available for download at [1]. 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 [27, 11] and NMPC-Controllers [20].

Problems can be set up and solved from a C/C++ environment as well as conveniently from MATLAB. qpDUNES provides set-up and solve routines (both cold- and warm-started) for multi-stage QPs, as well as for linear time-invariant (LTI) and for linear time-varying (LTV) MPC problems in both environments.

Please note that at the time of writing qpDUNES only supports box-constrained QPs with diagonal quadratic terms in the objective function. A version that employs a modified version of the open-source parametric active-set strategy qpOASES [13] and thus can deal with general convex multi-stage QPs is only in experimental stadium, as for high efficiency a deep interlacing on memory level is crucial.

## 8 Numerical Performance

### 8.1 Double Integrator

The first benchmark example is motivated by the energy optimal control of a cart on a rail in the context of a Badminton robot [34]. The dynamics of the system boil down to a simple double integrator with two states, position and velocity, and acceleration as control input. The optimization problem obtained after discretization is a convex QP in the form of Problem (P) with  $Q_k = \sigma \cdot \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$ ,  $R_k = [1]$ ,  $S_k = 0$ ,  $q_k = r_k = c_k = 0$ ,  $A_k = \begin{bmatrix} 1 & 0.01 \\ & 1 \end{bmatrix}$ ,  $B_k = \begin{bmatrix} 0 \\ 0.01 \end{bmatrix}$ ,  $D_k = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}$ ,  $\underline{d}_k = [-1.9 \ -3 \ -30]^\top$ , and  $\bar{d}_k = [1.9 \ 3 \ 30]^\top$  for all stages  $k \in \mathcal{S}$ , and  $k \in \mathcal{S}_0$ , respectively. Additionally we have the initial value constraint fixing  $x_0 = [-1 \ 0]^\top$  and two arrival constraints demanding the cart to arrive at position 0 at a certain index  $\bar{k}$  and staying there for at least 10ms (one time discretization step), giving the robot arm time to hit the shuttlecock. A small regularization term  $\sigma > 0$  in this formulation ensures positive definiteness of the  $Q_k$ .

This benchmark problem is particularly interesting, as it directly shows the limitations of the dual Newton strategy. Purely energy-minimal operation of the badminton robot would correspond to  $\sigma = 0$ , resulting in a non-strictly

convex QPs. To be able to treat this problem with the dual Newton strategy, regularization is always required. Despite this factual drawback, we still believe that the dual Newton strategy is well suited, since in almost all practical applications regularization is beneficial and often even leads to more desirable properties of the obtained solutions.

Additionally to the small regularization parameter (leading to a rather badly conditioned dual function), we choose arrival times close to infeasibility, ensuring that many state constraints become active in the solution. As discussed in section 3.4, scenarios with many active state constraints tentatively are particularly challenging for the dual Newton strategy. Also note in this context that the unconstrained optimum (favoring no action) lies far outside the feasible region.

We compare the average computation time over the first 20 MPC iterations again for horizon lengths of  $N = 50, 100, 150, 200$ , where the cart is forced to arrive at the desired position at  $\bar{k} = 50$ . We add random noise to the simulated system dynamics, and choose  $\sigma = 10^{-4}$ . If the MPC problem was rendered infeasible by the (precomputed) noise vector, we discard this instance and generate a new noise vector.

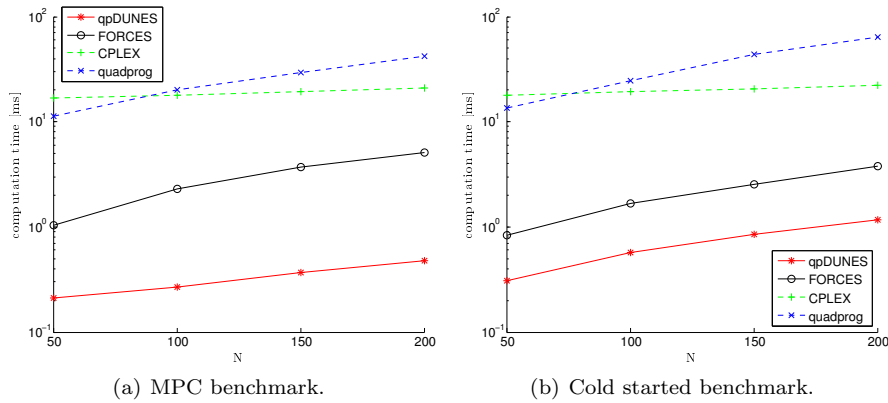
We report computation times on a standard desktop PC featuring a 3.4 GHz Intel i7 CPU under a Ubuntu 13.04 Linux for our method qpDUNES in comparison against FORCES [11], a very recent and highly efficient structure exploiting interior point method that uses automatic code generation to create a custom solver tailored for the dynamics of a MPC problem. We run FORCES with default settings, as we did not observe any significant performance improvement in other configurations. It was indicated in [33] that FORCES will outperform even very efficient active-set methods on prediction horizons of the considered length. It was indicated in [11] that FORCES also outperforms other tailored interior-point methods rigorously.

For completeness we include a comparison against the popular solvers CPLEX 12.5.1 [21], and Matlab's quadprog. CPLEX implements efficient general purpose QP solvers. Both, an active set method and an interior point method is available. We chose to use CPLEX in automatic mode (with parallelization switched off), as fixing CPLEX to either method rather lead to a performance decrease than increase. Matlab's quadprog was also run in default configuration, using its interior-point solver. We note that the active set method was observed to perform orders of magnitude worse than the interior-point method and is therefore not included in the comparison.

All benchmarks were run from Matlab R2013a, as FORCES is currently only provided via a mex-interface. CPLEX and quadprog were called as pre-compiled libraries for Linux through their default Matlab interface. FORCES was downloaded through its Matlab interface as a custom solver tailored to the model dynamics. Both FORCES and qpDUNES were compiled to a `.mex` file using gcc 4.7.3 with standard code optimization options.

We perform the MPC simulation 1000 times, with different random noise vectors, and report averaged computation times in milliseconds in Figure 2(a). We observe that both customized solvers, qpDUNES and FORCES, perform





**Fig. 2** Computation times of the double integrator benchmark.

observably well. Still, qpDUNES performs yet a factor of 4.9 – 10.6 better on this benchmark problem than the runner-up FORCES.

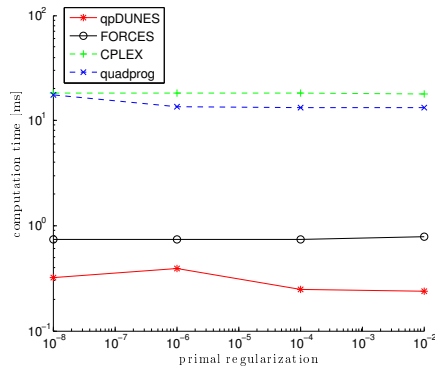
To get a glimpse on worst-case computation times (comparisons that tentatively favor interior-point methods), we additionally report solution times of a single QP without any prior knowledge about the solution (i.e., cold started). We chose horizon lengths of  $N = 50, 100, 150, 200$  and forced the cart to arrive at the desired position at  $\bar{k} = 45$  (almost the minimum time possible).

Even though qpDUNES is not tailored for a single QP solution, it was observed that qpDUNES outperforms the other considered solvers by a factor of 2.7 – 3.3 even on this benchmark scenario, cf. Figure 2(b). The numbers reported are in milliseconds, and averaged over 1000 identical (cold started) runs.

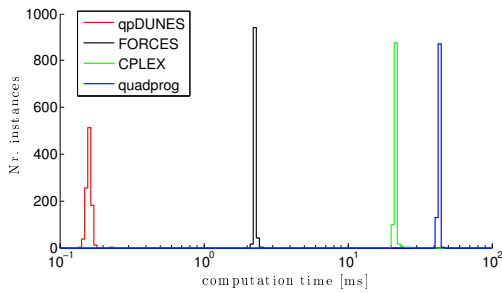
In a third comparison, we analyze the effect of different primal regularization parameters in the dual Newton strategy. For a horizon length of  $N = 50$ , and an arrival index of  $\bar{k} = 45$ , we compare values for  $\sigma$  between  $10^{-2}$  and  $10^{-8}$  (possibly causing very ill-conditioned dual functions) in Figure 3. Again, computation times are reported in milliseconds, and averaged over 1000 identical (cold started) runs. As expected, we observe that the choice of  $\sigma$  barely has any influence on the performance of the interior point methods. The dual Newton strategy in contrast is sensitive to the choice of  $\sigma$ , and there is a tendency to higher computational demand for smaller primal regularization parameters (i.e., a more ill-conditioned dual problem), as anticipated. Still, for all considered regularization parameters, qpDUNES outperformed its competitors.

## 8.2 Chain of masses

The second benchmark example is taken from [35]. MPC is used to control a chain of six oscillating masses that are connected by springs to each other and



**Fig. 3** Computation times of one cold started QP solution for different primal regularization parameters.



Solver	Computation time
qpDUNES	0.16 ms
FORCES	2.27 ms
CPLEX	21.41 ms
quadprog	42.83 ms

(a) Computation time histogram. Peaks from left to right: qpDUNES, FORCES, CPLEX, quadprog.

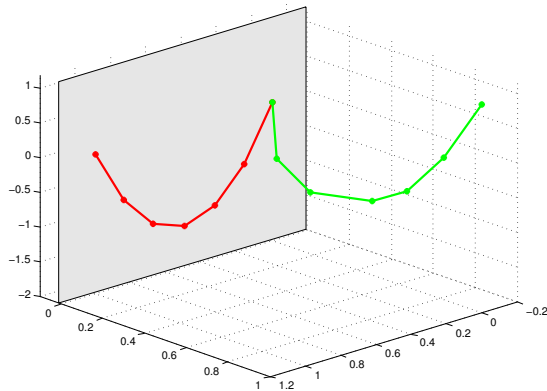
(b) Average computation times of one QP solution

**Fig. 4** Chain of masses benchmark problem.

to a wall on each side. We use the same parameters as stated in [35], and thus end up with an MPC problem of 12 states, 3 control inputs and a prediction horizon of 30 intervals. We simulate the MPC problem on 100 time steps.

We computed 1000 random noise vectors that perturb the positions of the masses. We again compare the computation times obtained from qpDUNES with those obtained from the solvers FORCES, CPLEX, and quadprog.

Table 4(b) shows average computation times for one QP solution, and Figure 4(a) shows a histogram of the average iteration times over the different instances. It can be seen that due to the efficient warmstarting, qpDUNES is at least one order of magnitude faster than the other solvers considered, even in presence of perturbations. We note that qpDUNES was rarely observed to exceed 4 iterations, while FORCES needed 6 to 9 iterations in most cases. Since the factorization in qpDUNES can be warmstarted, and the structure of the band-matrix is similar to the one occurring in interior-point methods, each iteration in the dual Newton strategy is roughly at most as expensive as an iteration of an interior point method. It should be noted that in this



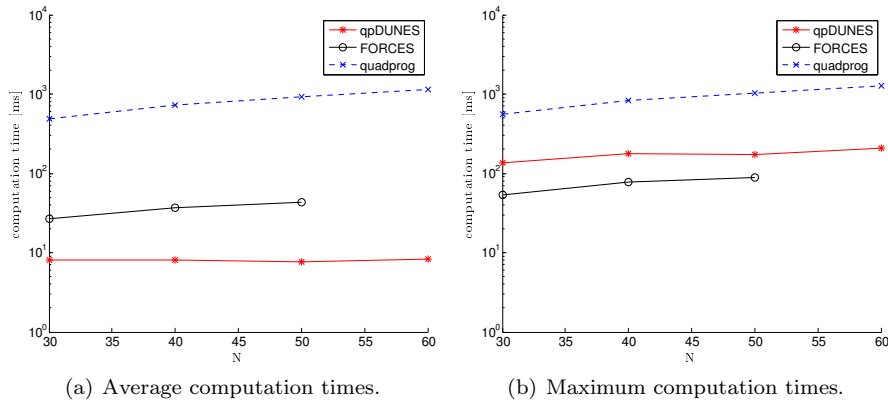
**Fig. 5** Hanging chain scenario. The chain in steady state is depicted in red, while the initial state is drawn in green.

benchmark problem significantly less constraints become active, a fact from which qpDUNES benefits overproportionally.

### 8.3 Hanging chain

To comment on the weaknesses of the dual Newton strategy, we consider a third problem, which has been used for benchmarking in several papers before, see [36, 12, 33]. The problem features again a chain of masses, yet in three-dimensional space, connected by springs, that assumes its steady state very close to a wall, cf. Figure 5. One end of the chain is fixed at the origin, while the other one is free and can be controlled by its velocities in  $x$ ,  $y$ , and  $z$  direction. Note that, as in [33], we placed the wall closer to the equilibrium position than it has been considered in the original setting from [36]. This means that potentially a large amount of state constraints becomes active in the solution, and thus we yield a more challenging problem, particularly for the dual Newton strategy. As in [12] we perform linear MPC based on a linearization in the steady state, trying to stabilize the problem quickly at its equilibrium. For the detailed model equations we refer to [36].

We consider a chain of 5 Masses, which results in a system of 33 states (the free masses' positions and velocities) and 3 controls (the last masses' velocities), on a varying horizon length between 30 and 60 intervals. Noise is added on the velocities of each mass in each simulation step (adding noise directly on the positions might result in an infeasible problem very easily due to the closeness between the equilibrium and the constraining wall). We explicitly note that systems of such a ratio between state dimensionality and horizon length are not the targeted application domain for the dual Newton strategy. Still, if we consider average computation times over 50 MPC iterations, we observe that the dual Newton strategy performs reasonably well, cf. Figure 6(a). Our solver qpDUNES performs a factor of 3.4 to 5.6 faster in this comparison



**Fig. 6** The hanging chain benchmark problem.

than FORCES, the tightest competitor. It is interesting to observe that the computation time is more or less constant over all considered horizon lengths. This can be explained by the efficient warmstarting of the factorization in the MPC context, cf. Section 3.5. Moreover, a longer horizon will cause that the equilibrium is reached (first at the end of the prediction horizon) at an earlier point in the simulation, thus making the multiplier shift more effective from this point on.

In this comparison we did not include CPLEX, as its Barrier method experienced numerical problems in almost all considered noise scenarios and therefore terminated early and suboptimally. We were also not able to obtain FORCES code for a problem of 60 or more intervals length with this number of states; note that this is not a numerical limitation of the solver, but rather a technical limitation of FORCES' download server, which generates the code specifically for each problem instance.

Despite all these shortcomings of the comparison, we believe that this very challenging problem yields some important insights when we consider maximum computation times in Figure 6(b). Here we observe that FORCES outperforms qpDUNES by a factor of 2.5 to 2 on the considered horizon lengths. Due to the fact that the objective is driving the optimization variables to points with many state constraints active (about 70 active constraints in the solution for the  $N = 50$  setting), qpDUNES takes many iterations that require regularization of the dual Hessian matrix. Overall, qpDUNES needed about twice as many iterations in the maximum than FORCES; additionally, many line search iterations were required when the dual Hessian needed to be regularized.

Nonetheless, this comparison also yields some positive news for the dual Newton strategy. We observed that the high computation times of qpDUNES were exclusively observed in the first MPC iteration, when qpDUNES was cold-started, and the initial condition is far away from the equilibrium. When

excluding this first iteration from the comparison, qpDUNES outperformed FORCES again by a factor of 3 to 6.1 also in maximum computation times for one MPC iteration. Therefore we are still convinced that the dual Newton strategy is of practical relevance.

## 9 Conclusions and outlook

We presented a novel algorithm, the dual Newton strategy, for the solution of convex quadratic programming problems that arise in optimal control and estimation contexts. Besides theoretic contributions, we introduced qpDUNES, an open-source implementation of the dual Newton strategy, that yielded very promising results in comparison with state of the art solvers on a number of benchmark problems.

We envisage to utilize qpDUNES as a QP subproblem solver in an SQP procedure for nonlinear model predictive control in the near future. We further want to extend the software implementation by several features that were presented in this paper, yet are still to be implemented in a reliable fashion, like the infeasibility detection (so far only prototypic), an efficient support for more general QPs and an efficient parallel implementation.

Besides these short term goals, also algorithmic extensions are conceivable. Since the dual Newton strategy can efficiently be warmstarted, one way to avoid singular Newton matrices at all would be by relaxing all stage constraints and penalizing their violation by an adaptive weighted term in the objective. Even further, the dual Newton strategy could be modified to treat general convex programming problems, using a convex subproblem solver.

## ACKNOWLEDGMENTS

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.

## References

1. qpDUNES Website. <http://mathopt.de/qpDunes>, 2013.
2. Alberto Bemporad and Panagiotis Patrinos. Simple and certifiable quadratic programming algorithms for embedded linear model predictive control. In *4th IFAC Nonlinear Model Predictive Control Conference.*, pages 14–20, 2012.

3. A.B. Berkelaar, K. Roos, and T. Terkaly. *Recent Advances in Sensitivity Analysis and Parametric Programming*, chapter 6: The Optimal Set and Optimal Partition Approach to Linear and Quadratic Programming. Kluwer Publishers, Dordrecht, 1997.
4. D.P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.
5. M.J. Best. *Applied Mathematics and Parallel Computing*, chapter An Algorithm for the Solution of the Parametric Quadratic Programming Problem, pages 57–76. Physica-Verlag, Heidelberg, 1996.
6. L.T. Biegler. Advances in nonlinear programming concepts for process control. *Journal of Process Control*, 8(5):301–311, 1998.
7. H.G. Bock. Recent advances in parameter identification techniques for ODE. In P. Deuffhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*. Birkhäuser, Boston, 1983.
8. Y.-H. Dai and R. Fletcher. New algorithms for singly linearly constrained quadratic programs subject to low and upper bounds. *Mathematical Programming*, 106(3):403–421, 2006.
9. John M. Danskin. *The theory of Max-Min and its application to weapons allocation problems*. Springer-Verlag, Berlin, New York, 1967.
10. M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.
11. A. Domahidi, A. Zraggen, M.N. Zeilinger, M. Morari, and C.N. Jones. Efficient Interior Point Methods for Multistage Problems Arising in Receding Horizon Control. In *IEEE Conference on Decision and Control (CDC)*, pages 668 – 674, Maui, HI, USA, December 2012.
12. H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.
13. H.J. Ferreau, C. Kirches, A. Potschka, H.G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 2013. (under review).
14. H.J. Ferreau, A. Kozma, and M. Diehl. 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, Noordwijkerhout, The Netherlands*, 2012.
15. A.V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. Academic Press, New York, 1983.
16. R. Fletcher. *Practical Methods of Optimization*. Wiley, Chichester, 2nd edition, 1987.
17. M. Gerdt and M. Kunkel. A nonsmooth Newton’s method for discretized optimal control problems with state and control constraints. *Journal of Industrial and Management Optimization*, 4:247–270, 2008.
18. P. E. Gill, W. Murray, and M. H. Wright. *Practical optimization*. Academic Press, London, 1981.
19. P.E. Gill and W. Murray. Numerically Stable Methods for Quadratic Programming. *Mathematical Programming*, 14:349–372, 1978.
20. 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.
21. IBM Corp. *IBM ILOG CPLEX V12.1, User’s Manual for CPLEX*, 2009.
22. C. Kirches, H.G. Bock, J.P. Schlöder, and S. Sager. Block structured quadratic programming for the direct multiple shooting method for optimal control. *Optimization Methods and Software*, 26:239–257, April 2010.
23. C. Kirches, L. Wirsching, H.G. Bock, and J.P. Schlöder. Efficient Direct Multiple Shooting for Nonlinear Model Predictive Control on Long Horizons. *Journal of Process Control*, 22(3):540–550, 2012.
24. P. Kühn, M. Diehl, T. Kraus, J. P. Schlöder, and H. G. Bock. A real-time algorithm for moving horizon state and parameter estimation. *Computers & Chemical Engineering*, 35(1):71–83, 2011.

25. 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.
26. W. Li and J. Swetits. A new algorithm for solving strictly convex quadratic programs. *SIAM Journal of Optimization*, 7(3):595–619, 1997.
27. J. Mattingley and S. Boyd. *Convex Optimization in Signal Processing and Communications*, chapter Automatic Code Generation for Real-Time Convex Optimization. Cambridge University Press, 2009.
28. J. Nocedal and S. Wright. *Numerical Optimization*. Springer-Verlag, 2000.
29. L. Qi and J. Sun. A nonsmooth version of Newton’s method. *Mathematical Programming*, 58:353–367, 1993.
30. C.V. Rao, S.J. Wright, and J.B. Rawlings. Application of Interior-Point Methods to Model Predictive Control. *Journal of Optimization Theory and Applications*, 99:723–757, 1998.
31. J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, 2009.
32. S. Richter, C.N. Jones, and M. Morari. Real-time input-constrained MPC using fast gradient methods. In *Proceedings of the IEEE Conference on Decision and Control, Shanghai, China*, 2009.
33. M. Vukov, A. Domahidi, H. J. Ferreau, M. Morari, and M. Diehl. Auto-generated Algorithms for Nonlinear Model Predictive Control on Long and on Short Horizons. In *Proceedings of the 52nd Conference on Decision and Control (CDC)*, 2013. (accepted).
34. X. Wang, J. Stoev, G. Pinte, and J. Swevers. Energy optimal point-to-point motion using Model Predictive Control. In *Proceedings ASME 2012 5th Annual Dynamic Systems and Control Conference*, 2012.
35. Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.
36. L. Wirsching, H. G. Bock, and M. Diehl. Fast NMPC of a chain of masses connected by springs. In *Proceedings of the IEEE International Conference on Control Applications, Munich*, pages 591–596, 2006.
37. S.J. Wright. Partitioned dynamic programming for optimal control. *SIAM Journal on optimization*, 1(4):620–642, 1991.
38. E. Zafriou. Robust model predictive Control of processes with hard constraints. *Computers & Chemical Engineering*, 14(4–5):359–371, 1990.