SIAM J. OPTIM.
Vol. 27, No. 3, pp. 2085–2109

A SPARSITY PRESERVING CONVEXIFICATION PROCEDURE FOR INDEFINITE QUADRATIC PROGRAMS ARISING IN DIRECT OPTIMAL CONTROL*

ROBIN VERSCHUEREN†, MARIO ZANON‡, RIEN QUIRYNEN§, AND MORITZ DIEHL¶

Abstract. Quadratic programs (QP) with an indefinite Hessian matrix arise naturally in some direct optimal control methods, e.g., as subproblems in a sequential quadratic programming scheme. Typically, the Hessian is approximated with a positive definite matrix to ensure having a unique solution; such a procedure is called *regularization.* We present a novel regularization method tailored for QPs with optimal control structure. Our approach exhibits three main advantages. First, when the QP satisfies a second order sufficient condition for optimality, the primal solution of the original and the regularized problem are equal. In addition, the algorithm recovers the dual solution in a convenient way. Second, and more importantly, the regularized Hessian bears the same sparsity structure as the original one. This allows for the use of efficient structure-exploiting QP solvers. As a third advantage, the regularization can be performed with a computational complexity that scales linearly in the length of the control horizon. We showcase the properties of our regularization algorithm on a numerical example for nonlinear optimal control. The results are compared to other sparsity preserving regularization methods.

Key words. regularization, nonlinear predictive control, SQP, optimal control

AMS subject classifications. 49M20, 90C30, 49J15

DOI. 10.1137/16M1081543

1. Introduction.

In recent decades, model predictive control (MPC) has become a popular optimization based control algorithm due to its ability to control multiple-input multiple-output constrained systems. Linear MPC consists in consecutively solving optimal control problems (OCPs) with quadratic objective function, linear dynamics, and linear inequality constraints. Nonlinear model predictive control (NMPC) [24] generalizes this to a nonlinear objective and nonlinear dynamics and path constraints. Originally, NMPC found interest in process control due to the slowly moving dynamics of the corresponding reactions. An overview of the industrial use of MPC can be found in [22]. Recently, both algorithmic and computational advances made NMPC applicable in real time for systems with fast dynamics, e.g., in [2, 23, 27, 29]. For an overview of computationally efficient methods for NMPC, we point the reader to [6, 19].

*Received by the editors June 27, 2016; accepted for publication (in revised form) April 27, 2017; published electronically September 26, 2017.
http://www.siam.org/journals/siopt/27-3/M108154.html
Funding: This research was supported by the EU via ERC-HIGHWIND (259 166), FP7-ITN-TEMPO (607 957), and H2020-ITN-AWESCO (642 682) and by the DFG via project FOR2401. The third author holds a PhD fellowship from the Research Foundation–Flanders (FWO).

†Department of Microsystems Engineering, Albert-Ludwigs-Universität, Georges-Koehler-Allee 102, 79110 Freiburg im Breisgau, Germany (robin.verschueren@imtek.uni-freiburg.de).
‡Department of Signals and Systems, Chalmers University of Technology, Hörsalsvägen 11, SE-41296 Göteborg, Sweden (zmario@chalmers.se).
§Department of Electrical and Electronical Engineering (ESAT), KU Leuven, Kasteelpark Arenberg 10, Leuven, Belgium, and Department of Microsystems Engineering, Albert-Ludwigs-Universität, Georges-Koehler-Allee 102, 79110 Freiburg im Breisgau, Germany (rien.quirynen@esat.kuleuven.be).
¶Department of Microsystems Engineering and Department of Mathematics, Albert-Ludwigs-Universität, Georges-Koehler-Allee 102, 79110 Freiburg im Breisgau, Germany (moritz.diehl@imtek.uni-freiburg.de).

One online algorithm for NMPC is the real-time iteration (RTI) [5] scheme, which is based on sequential quadratic programming (SQP). The RTI scheme is an example of a *direct* optimal control method, as it first discretizes the continuous-time OCP into a finite-dimensional optimization problem (in contrast to *indirect* methods), after which the problem is solved. RTI relies on multiple shooting [3] to discretize the continuous-time OCP, forming a nonlinear program (NLP) which is approximated by a quadratic programming (QP) subproblem with optimal control structure at each time step. In addition, the RTI algorithm is based on a generalized continuation approach to efficiently solve the parametric optimization problem depending on the current state of the system [6].

There exist several approaches to solving the structured QP subproblem. One technique, which is typically referred to as *condensing* [3], is to make use of the dynamic equations to eliminate the state variables, resulting in a smaller condensed problem with only controls as decision variables. The dense subproblem can be passed to a generic QP solver, e.g., QPKWIK [26], QPOPT [14], or qpOASES [8], to obtain the solution and afterward recover all of the state variables and Lagrange multipliers by a so-called expansion step [3]. A drawback of the condensing approach is that the computational complexity scales at best quadratically in the horizon length [1, 12].

Alternatively, one can solve the QP directly using a structure-exploiting QP solver, of which the computational complexity typically scales linearly with the horizon length. This becomes advantageous in OCPs with long horizon lenghts, in which case the condensing approach is less competitive [29]. Examples of structure-exploiting QP solvers tailored to optimal control are qpDUNES [11], FORCES [7], and HPMPC [13]. The software package HQP [10] is a general-purpose sparse QP solver that can readily be used to solve large-scale OCPs. Although the structure-exploiting QP solvers FORCES, HPMPC, and qpDUNES require a positive definite Hessian matrix, the second order sufficient conditions (SOSC) for optimality require positive definiteness only of the reduced Hessian, which is defined to be the Hessian matrix projected onto the null space of the active constraints [21]. We point out that HPMPC and FORCES would in principle be compatible with problems with an indefinite Hessian with positive definite reduced Hessian; however, this would not allow part of the code optimization for the Cholesky factorization and, therefore, indefinite Hessians are not supported. Moreover, qpDUNES does not allow indefinite Hessians, not even ones that are positive definite in the reduced space, as it is based on dual decomposition, which requires strictly convex Hessian matrices.

In NMPC, it often happens that the Hessian of the QP subproblem with OCP structure is indefinite and therefore should be approximated with a positive definite Hessian in order to make sure that the calculated step is a descent direction; in this paper, we refer to such a procedure as *regularization*. The Hessian regularization is performed right before solving the QP. More specifically, for the condensing approach, we could either regularize the full Hessian before performing the condensing step or we could regularize the condensed Hessian afterward. A numerical case study comparing these two alternatives is presented in [28]. On the other hand, for the case of structure-exploiting QP solvers, the solvers mentioned in the previous paragraph all require the full Hessian to be positive definite. The convexification method proposed in this paper is therefore particularly suited for structured QP subproblems.

There are several ways of performing regularization. Levenberg–Marquardt regularization consists in adding a multiple of the identity matrix to the Hessian [21]. In [20], a way of ensuring a positive definite Hessian without checking its eigenvalues, based on differential dynamic programming, is presented. One other method adapts the positive definiteness of the Hessian by directly modifying the factors of the Cholesky factorization or the symmetric indefinite factorization of the Hessian [21]. Quasi-Newton methods can generally be modified to directly provide a positive definite Hessian approximation; see, e.g., [15, 17].

Regularization is also an important algorithmic component for interior point methods. One could, for example, look at the KKT matrix and ensure that it has the correct inertia, as, e.g., in [9], by using an inertia-controlling factorization. A similar idea is used in IPOPT [30], which performs an inertia correction step on the KKT matrix whenever necessary. The inertia information comes from the indefinite symmetric linear system solvers used in that code. An improvement of the IPOPT regularization in case of redundant constraints is presented in [31]. One interesting alternative method of dealing with indefiniteness is presented in [16], which consists of solving QP subproblems with indefinite Hessians that can be proven to be equivalent to strictly convex QPs. Straightforward regularization methods typically modify the reduced Hessian of an optimization problem and therefore also its corresponding optimal solution. Replacing the Hessian with a positive definite one without altering the reduced Hessian is called *convexification* in this paper.

As the main contribution of this paper, we propose a structure-preserving convexification method for indefinite QPs with positive definite reduced Hessian. In case the Hessian is indefinite but the reduced Hessian is positive definite, we prove in this paper that the underlying convexity can be recovered by applying a modification to the original Hessian without altering the reduced Hessian and at the same time preserving the sparsity structure of the problem. The proposed algorithm can readily be extended to the case of indefinite reduced Hessians, resulting in a heuristic regularization approach which will be shown to perform well in practice. Our convexification approach therefore (a) provides a fully positive definite Hessian; (b) can be applied as a separate routine, independent of the QP solver used; and (c) preserves the optimal control sparsity structure and has a computational complexity that is linear in the horizon length.

Our convexification algorithm, which fulfills the above criteria, is a recursive procedure which exploits the block-diagonal structure of the Hessian and the stage-by-stage structure which is typical for direct optimal control. The resulting *convexified* Hessian can then be fed directly to a structure-exploiting QP solver. Note that by doing so, we avoid the potentially costly step of condensing. Instead, we directly solve the structured QP with the additional computational cost resulting from the above convexification procedure.

Our approach is motivated by the convergence of a Newton-type SQP method to a local solution for a nonlinear OCP. When employing the exact Hessian in such a method, convergence to a nearby local minimum is quadratic under mild assumptions [21]. When starting close enough to a local minimum, the convergence of the Newton-type method with convexified Hessian remains quadratic under the same assumptions. This will additionally be illustrated further in a numerical case study. We would like to point out that, e.g., the method in [16] will ultimately also recover quadratic convergence, but indefinite QP subproblems are solved instead of positive definite ones.

The structure of the paper is as follows. The problem setting is introduced in section 2. In section 3 we show how to recover convexity from general QPs and QPs with optimal control structure with positive definite reduced Hessians. This section also presents the structure-preserving convexification algorithm. Section 4 shows how to handle inequality constraints, and section 5 deals with problems with indefinite reduced Hessian. An illustration of our regularization method is given based on a nonlinear OCP example in section 6. The paper is concluded in section 7.

2. Problem formulation.

In this paper, we are interested in NLPs with an OCP structure.

Definition 2.1 (NLP with OCP structure).

$$\underset{\substack{s_0,\dots,s_N,\\ q_0,\dots,q_{N-1}}}{\text{minimize}} \quad \sum_{k=0}^{N-1} f_k(s_k, q_k) + f_N(s_N) \tag{1a}$$

$$\text{subject to} \quad s_{k+1} = \phi_k(s_k, q_k), \qquad k = 0, \dots, N-1, \tag{1b}$$

$$s_0 = \bar{s}_0, \tag{1c}$$

$$0 \geqslant c_k(s_k, q_k), \qquad k = 0, \dots, N-1, \tag{1d}$$

$$0 \geqslant c_N(s_N). \tag{1e}$$

In the above definition, $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ *is the stage cost at each stage* k *of the problem and* $f_N : \mathbb{R}^{n_x} \to \mathbb{R}$ *is the terminal cost. We denote the state vectors with* $s_k \in \mathbb{R}^{n_x}$ *and the controls with* $q_k \in \mathbb{R}^{n_u}$*. Function* $\phi_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ *is a discrete-time representation of the dynamic system which yields the state* s_{k+1} *at the next stage, given the current state and control* s_k, q_k*. The remaining constraints are the path constraints* $c_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_{c,k}}$ *and* $c_N : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{c,N}}$*, and the initial constraint where* $\bar{s}_0 \in \mathbb{R}^{n_x}$ *is fixed.*

Next, we define a QP with OCP structure, which is a more specific form of (1), where the objective is quadratic and the dynamics and inequality constraints are linear. It may also arise as a subproblem in an SQP-type method to solve the structured NLP (1).

Definition 2.2 (QP with OCP structure).

$$\mathrm{QP}(H): \quad \underset{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}}{\text{minimize}} \quad \frac{1}{2}\sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \overbrace{\begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix}}^{H_k} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \frac{1}{2} x_N^\top \widehat{Q}_N x_N \tag{2a}$$

$$\text{subject to} \quad x_{k+1} = A_k x_k + B_k u_k, \qquad k = 0, \dots, N-1, \tag{2b}$$

$$x_0 = \bar{x}_0, \tag{2c}$$

$$0 \geqslant C_{k,x} x_k + C_{k,u} u_k, \quad k = 0, \dots, N-1, \tag{2d}$$

$$0 \geqslant C_N x_N, \tag{2e}$$

where we define the state vectors as $x_k \in \mathbb{R}^{n_x}$*, the controls as* $u_k \in \mathbb{R}^{n_u}$*, and the cost matrices as* $Q_k, \widehat{Q}_N \in \mathbb{R}^{n_x \times n_x}, S_k \in \mathbb{R}^{n_u \times n_x}, R_k \in \mathbb{R}^{n_u \times n_u}$*, and the Hessian matrix* $H := \operatorname{diag}(H_0, \dots, H_{N-1}, \widehat{Q}_N)$*. The constraints denote, respectively, dynamic constraints with matrices* $A_k \in \mathbb{R}^{n_x \times n_x}, B_k \in \mathbb{R}^{n_x \times n_u}$*, inequality constraints with* $C_{k,x} \in \mathbb{R}^{n_{c,k} \times n_x}, C_{k,u} \in \mathbb{R}^{n_{c,k} \times n_u}, C_N \in \mathbb{R}^{n_{c,N} \times n_x}$*, and an initial constraint with* $\bar{x}_0 \in \mathbb{R}^{n_x}$*. Note that this compact notation, as proposed in, e.g.,* [13]*, allows for a more general OCP formulation including linear cost terms and constant terms in the constraints.*

In general, the Hessian H of QP (2) might be indefinite, for example, in case of an exact-Hessian based SQP method to solve NLP (1). However, this does not necessarily prevent the problem from being convex and therefore the solution of QP (2) to be global and unique. In the next section, we present a general framework to recover this underlying convexity.

3. Equality constrained problems in optimal control. For the sake of clarity of exposition, we omit the inequality constraints from QP (2) and refer to section 4 for a discussion on how to deal with them within the proposed convexification approach. Without inequality constraints, we can write QP (2a)–(2c) in a more compact form, as follows.

DEFINITION 3.1 (equality constrained QP).

$$\underset{w\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}w^\top H w \tag{3a}$$

$$\text{subject to} \quad Gw + g = 0 \tag{3b}$$

with constraint matrix $G \in \mathbb{R}^{p\times n}$ *of full rank* p *with* $p \leqslant n$. *Note that the linear independence constraint qualification (LICQ) requires full rank of* G [21]. *Furthermore, we have a symmetric but possibly indefinite Hessian matrix* $H \in \mathbb{S}^n$, *where we define the space of symmetric matrices of size* n *as follows:*

$$\mathbb{S}^n := \left\{X \in \mathbb{R}^{n\times n} \mid X = X^\top\right\}. \tag{4}$$

In case of a structured equality constrained QP (2a)–(2c), *we have that* $n = (N+1)\cdot n_x + N\cdot n_u$, $p = (N+1)\cdot n_x$ *and we choose the following ordering of the optimization variables:* $w^\top = [x_0^\top, u_0^\top, \dots, x_N^\top]$.

DEFINITION 3.2 (range space and null space basis of G). *Consider QP* (3) *and assume LICQ holds. We let* $Z \in \mathbb{R}^{n\times q}$ *with* $q = n - p$ *denote a null space basis with corresponding range space basis* $Y \in \mathbb{R}^{n\times p}$ *of* G *that satisfy the following:*

$$GZ = 0, \tag{5a}$$

$$(Y|Z)^\top (Y|Z) = I. \tag{5b}$$

Note that for any such Z *holds that* $\text{span}(Z) = \text{null}(G)$.

3.1. A characterization of convexity. Using Definitions 3.1–3.2, we can state some interesting properties of QP (3) with regard to convexity of the reduced Hessian. The following theorem is a well-known result, of which a proof is presented in [21].

THEOREM 3.3. *Consider QP* (3) *and Definition* 3.2, *assuming that LICQ holds. Then QP* (3) *has a unique global optimum if and only if the reduced Hessian is positive definite, i.e.,*

$$Z^\top H Z \succ 0. \tag{6}$$

We note that the reduced Hessian being positive definite corresponds to the SOSC for optimality, as defined in [21]. A well-known fact related to this SOSC is stated in the following theorem (for a proof, see, e.g., [21]).

THEOREM 3.4. *Consider QP* (3) *with arbitrary* $H \in \mathbb{S}^n$, *Definition* 3.2, *and assume that LICQ holds. Then*

$$Z^\top H Z \succ 0 \iff \exists \gamma \in \mathbb{R} : H + \gamma G^\top G \succ 0. \tag{7}$$

Theorem 3.4 is at the basis of the augmented Lagrangian method for optimization, where one typically calls $\gamma > 0$ the *quadratic penalty parameter*. By choosing γ large enough one can always create a positive definite Hessian at points satisfying SOSC. Note that $H + \gamma G^\top G$ destroys the sparsity pattern present in the original Hessian H, but an actual implementation would solve an augmented primal-dual system with the correct sparsity in the Hessian, e.g., as in [16]. Theorem 3.4 can be generalized as follows.

THEOREM 3.5 (revealing convexity). *Consider QP* (3) *and Definition* 3.2*, and assume LICQ holds. The reduced Hessian satisfies*

$$Z^\top H Z \succ 0 \tag{8a}$$

if and only if there exists a symmetric matrix $U \in \mathbb{S}^n$ *with*

$$Z^\top U Z = 0 \tag{8b}$$

such that

$$H + U \succ 0. \tag{8c}$$

Proof. From (8b), (8c), and the fact that Z is of full rank, (8a) directly follows. In order to prove the converse, let us introduce a change of basis, where the new basis is formed by $(Y|Z)$. Doing so, matrix inequality (8c) is equivalent to $(Y|Z)^\top(H + U)(Y|Z) \succ 0$. This again, due to (8b) and the Schur complement lemma [18], is equivalent to

$$Z^\top H Z \succ 0, \tag{9}$$

$$Y^\top(H+U)Y \succ Y^\top(H+U)Z \cdot (Z^\top H Z)^{-1} \cdot Z^\top(H+U)Y. \tag{10}$$

Thus, we need to show that there always exists a matrix U satisfying (8b) and (10).

Using the same change of basis as above for U and using (8b), it holds that

$$U = YKY^\top + YMZ^\top + ZM^\top Y^\top \tag{11}$$

with $K \in \mathbb{S}^p$ and $M \in \mathbb{R}^{p\times q}$. It directly follows that $Z^\top U Z = 0$. Statement (10) can then be written as

$$K \succ -Y^\top H Y + (Y^\top H Z + M) \cdot (Z^\top H Z)^{-1} \cdot (Y^\top H Z + M)^\top. \tag{12}$$

As K appears individually on the left-hand side, for given H, M, there always exists a matrix K such that (12) holds. Thus, there always exists a matrix U satisfying (10). ∎

The proof of Theorem 3.4 is obtained by choosing $M = 0$, i.e., $U = YKY^\top$, and observing that Y is a basis of the range space of $G^\top$. It is interesting to remark that the introduction of matrix U in order to obtain a certificate for second order optimality bears some similarity in spirit to the introduction of Lagrange multipliers in order to obtain a certificate of first order optimality.

The next section regards a different special case of Theorem 3.5, where we impose an OCP structure on U, which cannot be obtained by $U = \gamma G^\top G$ or its generalization $U = G^\top \Gamma G$. *Convexification* is our name for the process of finding such a matrix U, which we call *structure-preserving convexification* if U has the same sparsity structure as the original Hessian matrix H.

3.2. Structure-preserving convexification. The convexification algorithm presented in this section exploits the convexity of the reduced Hessian in order to compute a modified quadratic cost matrix $\widetilde{H} \in \mathbb{S}_{\mathrm{OCP}}^{N,n_x,n_u}$ which is positive definite and has the same sparsity pattern as H; here, we use the following definition for the space of symmetric block-diagonal matrices with OCP structure:

$$\mathbb{S}_{\text{OCP}}^{N,n_x,n_u} := \{X \in \mathbb{S}^{N(n_x+n_u)+n_x} \mid X = \operatorname{diag}(X_0, \ldots, X_N),\\ X_k \in \mathbb{S}^{n_x+n_u}, k = 0, \ldots, N-1, X_N \in \mathbb{S}^{n_x}\}.$$

The convexification can be performed by using the structure of the equality constraints. The resulting modified QP($\widetilde{H}$) has two important properties: (a) the primal solutions of QP(H) and QP($\widetilde{H}$) are equal; (b) $\widetilde{H}$ is positive definite if and only if the reduced Hessian of QP(H) is positive definite.

In the following, we establish property (a) in Theorem 3.6. Afterward, we present the convexification procedure in detail and state (b) in Theorem 3.10, which is the main result of this section. Furthermore, we propose a procedure to recover the dual solution of QP(H) from the solution of QP($\widetilde{H}$) in section 3.6. To conclude this section, we present a tutorial example.

Throughout this section, we use the following definitions (see the equality constrained QP (2a)–(2c)):

$$G := \begin{bmatrix} -I & & & & & \\ A_0 & B_0 & -I & & & \\ & & \ddots & \ddots & \ddots & \\ & & & A_{N-1} & B_{N-1} & -I \end{bmatrix}, \quad g := \begin{bmatrix} \overline{x}_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{13}$$

such that the dynamic equalities can be written as $Gw + g = 0$, where $w^\top = [x_0^\top, u_0^\top, \ldots, x_N^\top]$. Furthermore, we will use matrix Z as in Definition 3.2.

3.3. Transfer of cost between stages. The N stages of QP (2a)–(2c) are coupled by the dynamic constraints $x_{k+1} = A_k x_k + B_k u_k$. These constraints are used to transfer cost between consecutive stages of the problem without changing its primal solution. We introduce the *transferred cost* as follows:

$$\overline{q}_k(x) := x^\top \overline{Q}_k x, \qquad k = 0, \ldots, N, \tag{14}$$

for given matrices $\overline{Q}_k, \in \mathbb{S}^{n_x}, k = 0, \ldots, N$. Then, the stage cost l_k and the modified cost L_k are defined as follows (see (2a)–(2c)):

$$l_k(x_k, u_k) := \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top H_k \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \qquad k = 0, \ldots, N-1, \tag{15a}$$

$$L_k(x_k, u_k) := l_k(x_k, u_k) - \overline{q}_k(x_k) + \overline{q}_{k+1}(A_k x_k + B_k u_k) \tag{15b}$$

$$= \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \widetilde{H}_k \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \qquad k = 0, \ldots, N-1, \tag{15c}$$

in which the modified cost L_k is calculated by adding the transferred cost from the next stage $\overline{q}_{k+1}$ to the stage cost l_k and subtracting the cost $\overline{q}_k$, which is the cost to transfer to the previous stage. This yields a Hessian $\widetilde{H} = \operatorname{diag}(\widetilde{H}_0, \ldots, \widetilde{H}_{N-1}, \widetilde{Q}_N)$, where $\widetilde{Q}_N := \widehat{Q}_N - \overline{Q}_N$, which allows us to state the following theorem.

THEOREM 3.6 (equality of primal QP solutions). *Consider the equality constrained QP* (2a)–(2c) *and assume that $Z^\top H Z \succ 0$. Then, the primal solutions of* QP(H) *and* QP($\widetilde{H}$) *are equal.*

Proof. By assumption $Z^\top H Z \succ 0$; therefore QP(H) has a unique global minimum, by Theorem 3.3. The cost function of QP$(\widetilde{H})$ satisfies

$$\begin{aligned}
&\sum_{k=0}^{N-1} L_k(x_k, u_k) + x_N^\top \widetilde{Q}_N x_N \\
&\quad = \sum_{k=0}^{N-1} l_k(x_k, u_k) - \overline{q}_k(x_k) + \overline{q}_{k+1}(A_k x_k + B_k u_k) + x_N^\top \widetilde{Q}_N x_N \\
&\quad = \sum_{k=0}^{N-1} l_k(x_k, u_k) - \overline{q}_0(x_0) + \overline{q}_N(x_N) + x_N^\top \widetilde{Q}_N x_N \\
&\quad = \sum_{k=0}^{N-1} l_k(x_k, u_k) + x_N^\top \widehat{Q}_N x_N - \overline{q}_0(\overline{x}_0),
\end{aligned}$$

which is equal to the cost function of QP(H), up to the constant term $-\overline{q}_0(\overline{x}_0)$. It follows, because the constraints of QP(H) and QP$(\widetilde{H})$ are identical, that the primal solutions of both problems coincide. ∎

Note that we can write $\widetilde{H} = H + U(\overline{Q})$, where we let $U(\overline{Q}) := \operatorname{diag}(U_0, \ldots, U_N)$, with $\overline{Q} := \operatorname{diag}(\overline{Q}_0, \ldots, \overline{Q}_N)$ and the quantities $U_k, k = 0, \ldots, N$, are defined as follows:

$$U_k := \begin{bmatrix} A_k^\top \overline{Q}_{k+1} A_k - \overline{Q}_k & A_k^\top \overline{Q}_{k+1} B_k \\ B_k^\top \overline{Q}_{k+1} A_k & B_k^\top \overline{Q}_{k+1} B_k \end{bmatrix}, \quad U_N := -\overline{Q}_N. \tag{16}$$

Matrix U can be computed by Algorithm 1. Note that $H, U \in \mathbb{S}_{\text{OCP}}^{N,n_x,n_u}$ so that $\widetilde{H} \in \mathbb{S}_{\text{OCP}}^{N,n_x,n_u}$ also.

Theorem 3.6 states that transferring cost as in (15) does not alter the primal solution. In the following lemma we prove that also the reduced Hessian is invariant under cost transfer (15).

LEMMA 3.7. *Consider the equality constrained QP* (2a)–(2c), *Z from Definition* 3.2, *and U from* (16) *and assume LICQ is satisfied. Then, for any $\overline{Q}$, it holds that $Z^\top U(\overline{Q}) Z = 0$.*

Proof. Using (16) and (13), we can rewrite U as

$$\begin{aligned}
U(\overline{Q}) &= (G + \Sigma)^\top \overline{Q} (G + \Sigma) - \Sigma^\top \overline{Q} \Sigma, \\
&= G^\top \overline{Q} G + G^\top \overline{Q} \Sigma + \Sigma^\top \overline{Q} G,
\end{aligned} \tag{17}$$

Algorithm 1. $U(\overline{Q})$.

Input: $\overline{Q}$

Output: U

1: $U_N := -\overline{Q}_N$
2: **for** $k = N-1, \ldots, 0$ **do**
3: $\quad U_k := \begin{bmatrix} A_k^\top \overline{Q}_{k+1} A_k - \overline{Q}_k & A_k^\top \overline{Q}_{k+1} B_k \\ B_k^\top \overline{Q}_{k+1} A_k & B_k^\top \overline{Q}_{k+1} B_k \end{bmatrix}$
4: **end for**
5: $U := \operatorname{diag}(U_0, \ldots, U_N)$

---

where we introduced

$$\Sigma := \begin{bmatrix} I & 0 & & & \\ & & I & 0 & \\ & & & & \ddots & \\ & & & & & I \end{bmatrix}, \tag{18}$$

with $I \in \mathbb{R}^{n_x \times n_x}$ so that $\Sigma \in \mathbb{R}^{p \times n}$, where $n = (N+1) \cdot n_x + N \cdot n_u$, $p = (N+1) \cdot n_x$, as before. By definition $GZ = 0$; therefore it follows directly that $Z^\top U Z = 0$. ∎

To summarize, we give an explicit form for the blocks of the diagonal block matrix $\widetilde{H}$. For the last stage it holds that

$$\widetilde{Q}_N = \widehat{Q}_N + U_N = \widehat{Q}_N - \overline{Q}_N. \tag{19}$$

For the rest of the stages, we go in reverse order from $k = N-1$ to $k = 0$ and first compute the intermediate quantities

$$\begin{bmatrix} \widehat{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix} := \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix} + \begin{bmatrix} A_k^\top \overline{Q}_{k+1} A_k & A_k^\top \overline{Q}_{k+1} B_k \\ B_k^\top \overline{Q}_{k+1} A_k & B_k^\top \overline{Q}_{k+1} B_k \end{bmatrix}, \tag{20}$$

and then set

$$\widetilde{Q}_k = \widehat{Q}_k - \overline{Q}_k, \tag{21}$$

$$\widetilde{H}_k = H_k + U_k = \begin{bmatrix} \widetilde{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix}, \qquad k = 0, \dots, N-1. \tag{22}$$

Moreover, we remark on the resemblance of (17) and (11), so that we can write the following expressions for K, M:

$$K = Y^\top U Y = Y^\top \big(G^\top \overline{Q} G + G^\top \overline{Q} \Sigma + \Sigma^\top \overline{Q} G \big) Y,$$
$$M = Z^\top U Y = Z^\top \Sigma^\top \overline{Q} G Y.$$

Please note that for arbitrary $\overline{Q}$, the cost transfer operation presented in this section generally results in indefinite $\widetilde{H}$. In the next section, we will establish Theorem 3.10, which states that we can find a $U \in \mathbb{S}_{\mathrm{OCP}}^{N, n_x, n_u}$ with corresponding positive definite $\widetilde{H}$, and we present an algorithm to compute it.

3.4. The structure-preserving convexification algorithm. We propose a structure-preserving convexification procedure, which is built on (19)–(22). It computes $\widetilde{H} = H + U(\overline{Q})$, which can be shown to be positive definite, where $U(\overline{Q})$ is defined as in (16) based on a careful choice of $\overline{Q}$.

The procedure, shown in Algorithm 2, proceeds as follows: starting from the last stage, we choose a positive definite matrix $\widetilde{Q}_N = \delta I$, with $\delta > 0$ a small constant, such that $\overline{Q}_N = \widehat{Q}_N - \delta I$ (lines 1 and 2), and we use this matrix to transfer the cost $x_N^\top \overline{Q}_N x_N$ to the previous stage. The updated quantities $\widehat{Q}_{N-1}, \widehat{S}_{N-1}, \widehat{R}_{N-1}$ are calculated according to (20) in line 4. We use the Schur complement lemma [18] in line 5 of the algorithm to ensure that $\widetilde{H}_{N-1} \succ 0$. Next, we compute $\overline{Q}_{N-1} = \widehat{Q}_{N-1} - \widetilde{Q}_{N-1}$ and we repeat steps 4–7 until we arrive at the first stage of the problem.

By inspection of Algorithm 2, we have that the computational complexity scales linearly with the horizon length, i.e., it is $\mathcal{O}(N)$. We include $\overline{Q}(\delta)$ and $\widehat{R}(\delta) := \mathrm{diag}(\widehat{R}_0, \dots, \widehat{R}_{N-1})$ in the output of the algorithm for convenience.

Algorithm 2. Structure-preserving convexification: Equality constrained case.

Input: H, δ

Output: $\overline{Q}(\delta), \widetilde{H}(\delta), \widehat{R}(\delta)$

$\widetilde{Q}_N = \delta I$

$\overline{Q}_N = \widehat{Q}_N - \widetilde{Q}_N$

for $k = N-1, \dots, 0$ **do**

$$\begin{bmatrix} \widehat{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix} = \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix} + \begin{bmatrix} A_k^\top \overline{Q}_{k+1} A_k & A_k^\top \overline{Q}_{k+1} B_k \\ B_k^\top \overline{Q}_{k+1} A_k & B_k^\top \overline{Q}_{k+1} B_k \end{bmatrix}$$

$\widetilde{Q}_k := \widehat{S}_k^\top \widehat{R}_k^{-1} \widehat{S}_k + \delta I$

$$\widetilde{H}_k := \begin{bmatrix} \widetilde{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix}$$

$\overline{Q}_k = \widehat{Q}_k - \widetilde{Q}_k$

end for

$\widetilde{H} := \mathrm{diag}(\widetilde{H}_0, \dots, \widetilde{Q}_N)$

In Theorem 3.10, we show that Algorithm 2 indeed produces a positive definite $\widetilde{H}$, given a sufficiently small value for δ if and only if the reduced Hessian is positive definite. Lemmas 3.8 and 3.9, presented next, help us to prove this result.

LEMMA 3.8. *Consider the equality constrained QP* (2a)–(2c) *with OCP structure and Definition* 3.2*, assuming LICQ, $Z^\top H Z \succ 0$ hold. Then, Algorithm* 2 *with $\delta = 0$ delivers positive definite $\widehat{R}(0) \succ 0$ and positive semidefinite $\widetilde{H}(0) \succeq 0$.*

Proof. Since $\delta = 0$, Algorithm 2 starts with $\overline{Q}_N = \widehat{Q}_N$. Following a dynamic programming argument in order to solve QP (2a)–(2c), we have at each stage the following problem, with x_k fixed:

$$\text{(23a)} \qquad \underset{u_k}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix} + \frac{1}{2} x_{k+1}^\top \overline{Q}_{k+1} x_{k+1}$$

$$\text{(23b)} \qquad \text{subject to} \quad x_{k+1} = A_k x_k + B_k u_k,$$

which is equivalent to

$$\text{(24)} \qquad \underset{u_k}{\text{minimize}} \quad \frac{1}{2} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} Q_k + A_k^\top \overline{Q}_{k+1} A_k & S_k^\top + A_k^\top \overline{Q}_{k+1} B_k \\ S_k + B_k^\top \overline{Q}_{k+1} A_k & R_k + B_k^\top \overline{Q}_{k+1} B_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}.$$

By assumption, the reduced Hessian of the full QP (2a)–(2c) is positive definite, which, by Theorem 3.3, entails that the minimum of the QP is unique. Dynamic programming yields the same *unique* solution for each u_k, which in turn implies that the Hessian of (24) must be positive definite as well. This amounts to $(R_k + B_k^\top \overline{Q}_{k+1} B_k) = \widehat{R}_k \succ 0$ for $k = 0, \dots, N-1$. From the Schur complement lemma, we have that if $\widehat{R}_k \succ 0$,

$$\text{(25)} \qquad \widetilde{Q}_k - \widehat{S}_k^\top \widehat{R}_k^{-1} \widehat{S}_k \succeq 0 \iff \begin{bmatrix} \widetilde{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix} \succeq 0.$$

With $\delta = 0$, from Algorithm 2 it follows that $\widetilde{Q}_k = \widehat{S}_k^\top \widehat{R}_k^{-1} \widehat{S}_k$, such that the left-hand side of (25) holds. This entails that the Hessian blocks $\widetilde{H}_k \succeq 0$, such that, together with $\widetilde{Q}_N = 0$, it holds that $\widetilde{H} \succeq 0$. ∎

In the following, we regard matrices $\hat{R}_k$ from Algorithm 2 as the map $\hat{R}(\delta)$ and we recall that $\widehat{R}(\delta) := \operatorname{diag}(\widehat{R}_0, \ldots, \widehat{R}_{N-1})$. We use this map in Lemma 3.9, which will help us prove Theorem 3.10.

Lemma 3.9. *Consider QP* (2a)–(2c) *and assume that* $Z^\top HZ \succ 0$ *holds. Then there exists a value* $\delta > 0$ *such that Algorithm* 2 *computes a positive definite matrix* $\widehat{R}(\delta) \succ 0$.

Proof. Consider the map $\widehat{R}(\delta)$, implicitly defined by Algorithm 2. By Lemma 3.8 it holds that $\widehat{R}(0) \succ 0$. Furthermore, δ enters linearly in the equations of Algorithm 2 and each step of the algorithm is continuous. This includes line 5, where the inverse of $\widehat{R}_k$ appears, which is well-defined and continuous as long as $\widehat{R}_k(\delta)$ remains positive definite, which is true at $\delta = 0$. As a consequence, the map $\widehat{R}(\delta)$ is continuous at the origin. It follows that there exists a value $\delta > 0$ such that $\widehat{R}_k \succ 0, k = 0, \ldots, N-1$. ∎

We conclude this section by establishing our main result.

Theorem 3.10. *Consider QP* (2a)–(2c) *and Definition* 3.2 *and assume LICQ holds. Then* $Z^\top HZ \succ 0 \iff \exists \delta > 0$ *such that* $\widetilde{H}(\delta) \succ 0$ *as defined by Algorithm* 2.

Proof. Assume there exists some $\delta > 0$ such that $\widetilde{H} \succ 0$. Then $Z^\top HZ \succ 0$ follows from $\widetilde{H} = H + U$ and Lemma 3.7. The converse is proven as follows. From Algorithm 2 we have that $\widetilde{Q}_k - \widehat{S}_k^\top \widehat{R}_k^{-1} \widehat{S}_k = \delta I \succ 0$. By Lemma 3.9, $\exists \delta > 0$ such that $\widehat{R}_k \succ 0$. Then, from the Schur complement lemma, it follows that

$$\widetilde{H}_k := \begin{bmatrix} \widetilde{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix} \succ 0. \tag{26}$$

As $\widetilde{Q}_N = \delta I \succ 0$, it follows that $\widetilde{H} = \operatorname{diag}(\widetilde{H}_0, \ldots, \widetilde{H}_{N-1}, \widetilde{Q}_N) \succ 0$. ∎

3.5. Connection with the discrete Riccati equation. We establish next an interesting relation between Algorithm 2 and the discrete-time Riccati equation. This result is not needed in the remainder of this paper but is given for completeness.

Definition 3.11 (discrete-time Riccati equation). *For a discrete linear time varying system* $x_{k+1} = A_k x_k + B_k u_k$, *the discrete-time Riccati equation starting with* $X_N := \overline{Q}_N$ *iterates backward from* $k = N-1$ *to* $k = 0$ *by computing*

$$X_k = Q_k + A_k^\top X_{k+1} A_k - (S_k^\top + A_k^\top X_{k+1} B_k)(R_k + B_k^\top X_{k+1} B_k)^{-1}(S_k + B_k^\top X_{k+1} A_k). \tag{27}$$

We call matrices X_k *the* cost-to-go *matrices for* $k = 0, \ldots, N$.

Lemma 3.12. *Consider QP* (2a)–(2c) *and Algorithm* 2 *and assume* $Z^\top HZ \succ 0$, *with* Z *as in Definition* 3.2. *If* $\delta = 0$, *then the matrices* $\overline{Q}_0, \ldots, \overline{Q}_N$ *in the output of Algorithm* 2 *are equal to the cost-to-go matrices* $X_0, \ldots, X_N$ *computed with the discrete-time Riccati equation as defined above.*

Proof. For stage N, $X_N = \overline{Q}_N$ by definition. For $k = 0, \ldots, N-1$, from Definition 3.11, we have that

$$X_k = Q_k + A_k^\top X_{k+1} A_k - (S_k^\top + A_k^\top X_{k+1} B_k)(R_k + B_k^\top X_{k+1} B_k)^{-1}(S_k + B_k^\top X_{k+1} A_k), \tag{28}$$

and, if we replace X_{k+1} by $\overline{Q}_{k+1}$,

$$X_k = \widehat{Q}_k - \widehat{S}_k^\top \widehat{R}_k^{-1} \widehat{S}_k, \tag{29}$$

$$= \overline{Q}_k, \tag{30}$$

where we used $\widehat{Q}_k, \widehat{S}_k, \widehat{R}_k$ as in (20) for ease of notation, and (30) follows from Algorithm 2, where $\overline{Q}_k = \widehat{Q}_k - \widetilde{Q}_k$, which is equivalent to the right-hand side of (29) for $\delta = 0$. ∎

3.6. Recovering the dual solution of the original QP. We propose a procedure to recover the dual solution of QP(H) from its primal solution. We define the Lagrangian of the equality constrained QP (2a)–(2c) as follows:

$$\begin{aligned}\mathcal{L}(w,\lambda) = \frac{1}{2}\sum_{k=0}^{N-1} x_k^\top Q_k x_k + 2x_k^\top S_k^\top u_k + u_k^\top R_k u_k + \frac{1}{2} x_N^\top \widehat{Q}_N x_N \\ + \sum_{k=0}^{N-1} \lambda_{k+1}^\top \big(A_k x_k + B_k u_k - x_{k+1}\big) + \lambda_0^\top (\overline{x}_0 - x_0)\end{aligned} \tag{31}$$

with $\lambda^\top = [\lambda_0^\top, \dots, \lambda_N^\top], \lambda_k \in \mathbb{R}^{n_x}$. We can obtain the Lagrange multipliers by computing the partial derivatives of the Lagrangian with respect to x_k, which should equal zero by the necessary conditions for optimality [21]. The derivation is shown below, and a procedure to compute λ is stated in Algorithm 3.

$$0 = \frac{\partial \mathcal{L}(w,\lambda)}{\partial x_k}^\top, \qquad k = 0, \dots, N-1, \tag{32}$$

$$= Q_k x_k + S_k^\top u_k + A_k^\top \lambda_{k+1} - \lambda_k, \tag{33}$$

where additionally

$$0 = \frac{\partial \mathcal{L}(w,\lambda)}{\partial x_N}^\top = \widehat{Q}_N x_N - \lambda_N. \tag{34}$$

3.7. A tutorial example. To illustrate our convexification method with a simple example, we regard the following one-stage OCP:

$$\underset{s_0,q_0,s_1}{\text{minimize}} \quad s_0^2 - \frac{1}{2}q_0^2 + s_1^2 + s_1^4 \tag{35a}$$

$$\text{subject to} \quad s_1 = s_0 + q_0, \tag{35b}$$

$$s_0 = \overline{s}_0. \tag{35c}$$

As the term s_1^4 makes this problem an NLP, we solve it by using an SQP method with exact Hessian. We set $\overline{s}_0 = 1$.

Algorithm 3. Recovery of Lagrange multipliers: Equality constrained case.

Input: w

Output: λ

```
λ_N = Q̂_N x_N
for k = N − 1, ..., 0 do
    λ_k ← Q_k x_k + S_k^⊤ u_k + A_k^⊤ λ_{k+1}
end for
```

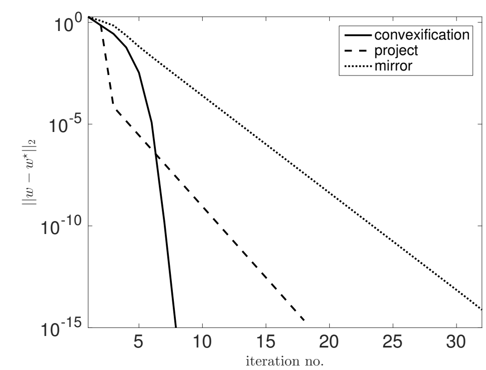

FIG. 1. *Convergence for SQP algorithm with three different regularization methods. On the vertical axis we plot the distance to the global solution $y^\star$. The Hessian obtained by Algorithm 2 with $\delta = 10^{-4}$ enables quadratic convergence of the exact Newton method, in contrast to the alternative regularization methods with $\epsilon = 10^{-4}$.*

We define $y := [s_0, q_0, s_1]^\top$. Optimization problem (35) has a global minimizer at $y^\star = [1, -3/2, -1/2]^\top$. The Hessian of the Lagrangian at the solution is equal to

$$\nabla^2_{yy}\mathcal{L}(y^\star, \lambda^\star) = \nabla^2 f(y^\star) = \operatorname{diag}(2, -1, 5) \nsucc 0. \tag{36}$$

We define

$$G := \begin{bmatrix} -1 & 0 & 0 \\ 1 & 1 & -1 \end{bmatrix}, \quad Z := \begin{bmatrix} 0 \\ \sqrt{2}/2 \\ \sqrt{2}/2 \end{bmatrix}, \tag{37}$$

such that $GZ = 0$ and $Z^\top Z = I$. The reduced Hessian at the solution $y^\star$ then reads as $Z^\top \nabla^2_{yy}\mathcal{L}(y^\star, \lambda^\star) Z = 2 > 0$.

Applying our convexification with sufficiently small δ therefore results in strictly convex QP subproblems, when the SQP method is sufficiently close to the minimizer.

We compare the convergence of Newton's method which employs either the proposed convexification method or the regularization methods defined in (38), called *project* and *mirror*, which do, instead, modify the reduced Hessian. Note that these regularizations fulfill all three properties of the regularizations that we desire, as mentioned in the introduction: they yield a fully positive definite Hessian, they are independent of the QP solver, and they preserve the OCP structure.

With $V_k D_k V_k^{-1}$ the eigenvalue decomposition of the Hessian block H_k, $k = 0, \dots, N$, these two regularizations are defined as follows:

$$\operatorname{project}(H_k, \epsilon) := V_k \left[\max(\epsilon, D_k) \right] V_k^{-1}, \tag{38a}$$

$$\operatorname{mirror}(H_k, \epsilon) := V_k \left[\max(\epsilon, \operatorname{abs}(D_k)) \right] V_k^{-1}, \tag{38b}$$

where $\operatorname{abs}(\cdot)$ is the operator which takes the elementwise absolute value and $\epsilon > 0$.

In Figure 1, we compare convergence of the SQP method for the convexification scheme and the two alternative regularizations in (38). We can observe from the figure that Newton's method which employs the proposed convexification procedure exhibits locally quadratic convergence to the solution, as it finds the correct primal and dual solution of the QP in each iteration. The other regularization methods result in linear convergence.

4. Inequality constrained optimization. In the previous section, we analyzed the case without inequality constraints. In this section, we analyze the general case and present a generalized version of Algorithm 2.

4.1. Revealing convexity under active inequalities. We regard the following compact formulation of NLP (1):

$$\underset{y\in\mathbb{R}^n}{\text{minimize}} \quad f(y) \quad \text{subject to} \quad g(y)=0, \quad c(y)\leqslant 0, \tag{39}$$

with equality constraints $g:\mathbb{R}^n\to\mathbb{R}^{n_g}$ and inequality constraints $c:\mathbb{R}^n\to\mathbb{R}^{n_c}$ with index set $\mathcal{I}=\{1,2,\dots,n_c\}$. We choose the following order for the optimization variables (see (1)): $y^\top := [s_0^\top, q_0^\top, \dots, s_N^\top]$. We define the Lagrangian of NLP (39) as

$$\mathcal{L}(y,\lambda,\mu) = f(y) + \lambda^\top g(y) + \mu^\top c(y), \tag{40}$$

with Lagrange multipliers $\lambda\in\mathbb{R}^{n_g}, \mu\in\mathbb{R}^{n_c}$. A KKT-point $z^\star := (y^\star,\lambda^\star,\mu^\star)$ at which LICQ, SOSC, and strict complementarity hold, is called a *regular solution* of NLP (39). We define the active set at a feasible point y as follows:

$$\mathcal{A}(y) := \{i\in\mathcal{I} \mid c_i(y)=0\}. \tag{41}$$

Solving NLP (39) with an exact-Hessian SQP method results, at iterate $(\overline{y},\overline{\lambda},\overline{\mu})$, in QP subproblems of the form

$$\mathrm{QP}_{\mathrm{SQP}}(\overline{y},H) := \underset{w\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}w^\top H\, w + \nabla f(\overline{y})^\top w \tag{42a}$$

$$\text{subject to} \quad 0 = g(\overline{y}) + \frac{\partial g}{\partial y}(\overline{y})w, \tag{42b}$$

$$0 \geqslant c(\overline{y}) + \frac{\partial c}{\partial y}(\overline{y})w, \tag{42c}$$

with optimal active set $\mathcal{A}^\star_{\mathrm{QP}}(\overline{y},w^\star,H)$.

Remark 4.1. Since $\nabla^2_{yy}\mathcal{L}(\overline{z})$ might be indefinite, there could be multiple solutions to $\mathrm{QP}_{\mathrm{SQP}}(\overline{y},\nabla^2_{yy}\mathcal{L}(\overline{z}))$. In the following, we will assume, as, e.g., in [17, 25], that $w^\star$, the solution of $\mathrm{QP}_{\mathrm{SQP}}(\overline{y},\nabla^2_{yy}\mathcal{L}(\overline{z}))$, is the *minimum-norm* solution.

For a *minimum-norm* solution of $\mathrm{QP}_{\mathrm{SQP}}(\overline{y},\nabla^2_{yy}\mathcal{L}(\overline{z}))$, the following lemma from [21] holds.

LEMMA 4.2. *Suppose that $z^\star$ is a regular solution of* (39). *Then if $\overline{z} := (\overline{y},\overline{\lambda},\overline{\mu})$ is sufficiently close to $z^\star$, the minimum-norm solution of $\mathrm{QP}_{\mathrm{SQP}}(\overline{y},\nabla^2_{yy}\mathcal{L}(\overline{z}))$ has an active set $\mathcal{A}^\star_{\mathrm{QP}}(\overline{y},w^\star,\nabla^2_{yy}\mathcal{L}(\overline{z}))$ that is the same as the active set $\mathcal{A}(y^\star)$ of NLP* (39) *at $z^\star$.*

Suppose we are at a point $(\overline{y},\overline{\lambda},\overline{\mu})$ sufficiently close to a solution of the NLP. Then Lemma 4.2 serves as a motivation to define a QP, which is the same as QP (42), but with the active inequalities replaced by equalities. We will use the following shorthand: $G := \frac{\partial g}{\partial y}(\overline{y}), G_{\mathrm{act}} := \frac{\partial c_i}{\partial y}(\overline{y}), i\in\mathcal{A}(y^\star)$, and n_a denotes the number of active constraints. For ease of notation, we omit the dependence of G, G_{act} on $\overline{y}$, as it is constant within one QP subproblem.

DEFINITION 4.3 (QP with fixed active set).

$$\underset{w\in\mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}w^\top H w \tag{43a}$$

$$\text{subject to} \quad \widetilde{G}w + \widetilde{g} = 0 \tag{43b}$$

with $\widetilde{G} := \left[\begin{smallmatrix} G \\ G_{\mathrm{act}} \end{smallmatrix}\right]$, $G \in \mathbb{R}^{p\times n}, G_{\mathrm{act}} \in \mathbb{R}^{n_a\times n}$, *and* $\widetilde{g} := \left[\begin{smallmatrix} g \\ g_{\mathrm{act}} \end{smallmatrix}\right]$, $g \in \mathbb{R}^p$, $g_{\mathrm{act}} \in \mathbb{R}^{n_a}$. *Note that we omitted the gradient term in the objective from* (42)*, for ease of notation, by performing the same transformation of variables as in* [13]*. Additionally, we introduce the following matrices.*

Definition 4.4 (null space of equalities and active inequalities). *Consider* $\widetilde{G}$ *as in QP* (43)*. We define* $\widetilde{Z} \in \mathbb{R}^{n\times(n-p-n_a)}$ *such that the following hold:*

$$\widetilde{G}\widetilde{Z} = 0, \quad \widetilde{Z}^\top \widetilde{Z} = I. \tag{44}$$

Matrix $\widetilde{Z}$ *is a basis for the null space of* $\widetilde{G}$*. The null space of* G *comprises* $\widetilde{Z}$ *but is possibly larger. We complement* $\widetilde{Z}$ *with* $Z_c \in \mathbb{R}^{n\times n_a}$ *and introduce* $Z \in \mathbb{R}^{n\times(n-p)}$ *as a basis for the null space of* G*, as follows:*

$$Z = (Z_c|\widetilde{Z}), \quad GZ = 0, \quad Z^\top Z = I. \tag{45}$$

From now on we call $\widetilde{Z}^\top H\widetilde{Z}$ the reduced Hessian. Furthermore, note that the above definition of Z is compatible with Definition 3.2.

Using Definition 4.4, we establish an extension of Theorem 3.5 for the case of active inequality constraints in Theorem 4.6, after proving the following lemma.

Lemma 4.5. *Consider QP* (43) *and Definition* 4.4 *and assume LICQ holds. We then have the following equivalence:*

$$\widetilde{Z}^\top H\widetilde{Z} \succ 0 \iff \exists\Gamma \in \mathbb{S}^{n_a} : Z^\top(H + G_{\mathrm{act}}^\top\Gamma G_{\mathrm{act}})Z \succ 0. \tag{46}$$

Proof. The proof follows a similar argument as in the proof of Theorem 3.5. We consider $H + G_{\mathrm{act}}^\top\Gamma G_{\mathrm{act}}$ in the basis $Z = (Z_c|\widetilde{Z})$; see (45). Applying the Schur complement lemma yields the following conditions:

$$\widetilde{Z}^\top H\widetilde{Z} \succ 0, \tag{47}$$

$$Z_c^\top(H + G_{\mathrm{act}}^\top\Gamma G_{\mathrm{act}})Z_c \succ Z_c^\top H\widetilde{Z}\cdot(\widetilde{Z}^\top H\widetilde{Z})^{-1}\cdot\widetilde{Z}^\top HZ_c, \tag{48}$$

where we used the fact that $G_{\mathrm{act}}\widetilde{Z} = 0$. The first inequality is the same as the left-hand side of (46). Using (45), and due to LICQ and the fact that Z_c is orthogonal to the null space of $\widetilde{G}$ and part of the null space of G, it holds that $G_{\mathrm{act}}Z_c$ is of full rank and therefore invertible. Thus, (48) becomes

$$\Gamma \succ (Z_c^\top G_{\mathrm{act}}^\top)^{-1}(Z_c^\top H\widetilde{Z}\cdot(\widetilde{Z}^\top H\widetilde{Z})^{-1}\cdot\widetilde{Z}^\top HZ_c - Z_c^\top HZ_c)(G_{\mathrm{act}}Z_c)^{-1}. \tag{49}$$

As Γ appears solely on the left side of the inequality, there always exists a Γ such that condition (48) is met. ∎

Theorem 4.6. *Consider QP* (43) *and Definition* 4.4 *and assume LICQ holds. It then holds that*

$$\widetilde{Z}^\top H\widetilde{Z} \succ 0 \iff \exists\Gamma \in \mathbb{S}^{n_a}, \exists U \in \mathbb{S}^n : Z^\top UZ = 0, H + G_{\mathrm{act}}^\top\Gamma G_{\mathrm{act}} + U \succ 0. \tag{50}$$

Proof. The proof of the theorem follows from Theorem 3.5 and Lemma 4.5.

4.2. Preserving the OCP structure. Theorem 4.6 can be specialized for the case of an OCP structure. To this end, let us introduce the following notation. We consider again problems with OCP structure as in (1). For such a problem we have a stagewise active set as follows:

$$\mathcal{A}_k(\overline{y}) := \{ i \in \mathcal{I}_k \mid \mathrm{row}_i(c_k(\overline{y})) = 0 \} \tag{51}$$

with $\mathcal{I}_k$ the index set corresponding to the inequalities in each stage $k = 0, \ldots, N$, respectively. We can now define $G_{\mathrm{act},k}$ at some feasible point $\overline{y}$, as follows:

$$G_{\mathrm{act},k} := \frac{\partial \mathrm{row}_i(c_k)}{\partial y}(\overline{y}) \quad : i \in \mathcal{A}_k(\overline{y}) \tag{52}$$

for $k = 0, \ldots, N$. Again, we omit $\overline{y}$ in the notation for $G_{\mathrm{act},k}$ for improved readability of the equations. Furthermore, we define $G_{\mathrm{act}}^\top := (G_{\mathrm{act},0}^\top \mid \cdots \mid G_{\mathrm{act},N}^\top)$. We remark that $G_{\mathrm{act}}^\top G_{\mathrm{act}} \in \mathbb{S}_{\mathrm{OCP}}^{N,n_x,n_u}$. Using these definitions, we can establish the following theorem.

THEOREM 4.7. *Consider QP* (2) *and Definition* 4.4 *and assume that LICQ holds. Then, it holds that*

$$\widetilde{Z}^\top H \widetilde{Z} \succ 0$$
$$\Longleftrightarrow$$
$$\exists \gamma \in \mathbb{R}, \exists U \in \mathbb{S}_{\mathrm{OCP}}^{N,n_x,n_u} : Z^\top U Z = 0,\ H + \gamma G_{\mathrm{act}}^\top G_{\mathrm{act}} + U \succ 0.$$

Proof. The proof follows from Theorem 4.6 with matrix $\Gamma = \gamma I$ and Theorem 3.10. Note that the sparsity structure of H is preserved in $\widetilde{H} := H + \gamma G_{\mathrm{act}}^\top G_{\mathrm{act}} + U$, as $U \in \mathbb{S}_{\mathrm{OCP}}^{N,n_x,n_u}$ and $G_{\mathrm{act}}^\top G_{\mathrm{act}} \in \mathbb{S}_{\mathrm{OCP}}^{N,n_x,n_u}$. ∎

We now present the structure-preserving convexification algorithm, for problems with inequalities, in Algorithm 4. It works along the same lines as Algorithm 2, with the difference that we add $\gamma G_{\mathrm{act},k}^\top G_{\mathrm{act},k}$ to the original Hessian blocks. For clarity, we introduce an operator $\mathcal{H}(H, \delta, \gamma, \mathcal{A})$ that computes the convexified Hessian $\widetilde{H}$.

Moreover, in the OCP case, we can again show that the primal solutions of QP(H) and QP$(\widetilde{H})$ are equal.

Algorithm 4. Structure-preserving convexification: Inequality constrained case.

Input: H, δ, γ, current active set $\mathcal{A}$

Output: $\widetilde{H} := \mathcal{H}(H, \delta, \gamma, \mathcal{A})$

$\widetilde{Q}_N = \delta I$

$\overline{Q}_N = \widehat{Q}_N + \gamma G_{\mathrm{act},N}^\top G_{\mathrm{act},N} - \widetilde{Q}_N$

for $k = N-1, \ldots, 0$ **do**

$\quad \begin{bmatrix} \widehat{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix} = \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix} + \begin{bmatrix} A_k^\top \overline{Q}_{k+1} A_k & A_k^\top \overline{Q}_{k+1} B_k \\ B_k^\top \overline{Q}_{k+1} A_k & B_k^\top \overline{Q}_{k+1} B_k \end{bmatrix} + \gamma G_{\mathrm{act},k}^\top G_{\mathrm{act},k}$

$\quad \widetilde{Q}_k := \widehat{S}_k^\top \widehat{R}_k^{-1} \widehat{S}_k + \delta I$

$\quad \widetilde{H}_k := \begin{bmatrix} \widetilde{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix}$

$\quad \overline{Q}_k = \widehat{Q}_k - \widetilde{Q}_k$

end for

$\widetilde{H} := \mathrm{diag}(\widetilde{H}_0, \ldots, \widetilde{Q}_N)$

THEOREM 4.8. *Consider a primal solution $w^\star$ of QP(H) as defined in* (2) *with optimal active set $\mathcal{A}^\star_{\mathrm{QP}}$. Furthermore, consider Definitions* 4.3 *and* 4.4*, and we assume that LICQ and $\widetilde{Z}^\top H \widetilde{Z} \succ 0$ hold at $w^\star$. Then there exist δ, γ such that $w^\star$ is the unique primal solution of QP($\widetilde{H}$) with $\widetilde{H} = \mathcal{H}(H, \delta, \gamma, \mathcal{A}^\star_{\mathrm{QP}})$.*

Proof. The proof is based on the null space method for solving equality constrained QPs, as presented in [21]. We decompose the primal solution vector w as follows:

$$w = \widetilde{Z} w_z + \widetilde{Y} w_y \tag{53}$$

with $\widetilde{Z}$ as in Definition 4.4, and we complement the basis of the null space of $\widetilde{G}$ with a basis of its range space $\widetilde{Y}$, such that $(\widetilde{Z}|\widetilde{Y})^\top(\widetilde{Z}|\widetilde{Y}) = I$. We can compute w_y from the constraints:

$$G\widetilde{Y} w_y = -g, \tag{54}$$

$$G_{\mathrm{act}}\widetilde{Y} w_y = -g_{\mathrm{act}}. \tag{55}$$

We can obtain w_z as follows. From the first order optimality conditions for QP (2) we have that

$$H\widetilde{Z} w_z + H\widetilde{Y} w_y + G^\top \lambda + G_{\mathrm{act}}^\top \mu = 0. \tag{56}$$

Multiplying from the left with $\widetilde{Z}^\top$ gives

$$\widetilde{Z}^\top H \widetilde{Z} w_z = -\widetilde{Z}^\top H \widetilde{Y} w_y, \tag{57}$$

where we used the fact that $\widetilde{Z}$ forms a basis for the null space of $\widetilde{G}$. Since $\widetilde{Z}^\top H \widetilde{Z} \succ 0$ by assumption, the solution is well-defined. Substituting H by $\widetilde{H} = H + U + \gamma G_{\mathrm{act}}^\top G_{\mathrm{act}}$, we have that

$$\begin{aligned} &\left(\widetilde{Z}^\top H \widetilde{Z} + \underbrace{\widetilde{Z}^\top U \widetilde{Z}}_{=0} + \gamma \underbrace{\widetilde{Z}^\top G_{\mathrm{act}}^\top}_{=0} G_{\mathrm{act}} \widetilde{Z} \right) w_z \\ &\quad = -\widetilde{Z}^\top H \widetilde{Y} w_y - \widetilde{Z}^\top U \widetilde{Y} w_y - \underbrace{\widetilde{Z}^\top G_{\mathrm{act}}^\top}_{=0} G_{\mathrm{act}} \widetilde{Y} w_y. \end{aligned} \tag{58}$$

The left-hand side of (58) is equal to the left-hand side of (57), due to Theorem 4.7 and Definition 4.4. In order to prove equality of the right-hand side, we use Definition 4.4 and (17) to obtain

$$\widetilde{Z}^\top U \widetilde{Y} w_y = \underbrace{\widetilde{Z}^\top G^\top}_{=0} \overline{Q}(G + \Sigma)\widetilde{Y} w_y + \widetilde{Z}^\top \Sigma^\top \overline{Q} G \widetilde{Y} w_y, \tag{59}$$

and we can show that $\widetilde{Z}^\top \Sigma^\top \overline{Q} G \widetilde{Y} w_y = 0$ as follows:

$$\begin{aligned} \widetilde{Z}^\top \Sigma^\top \overline{Q} G \widetilde{Y} w_y &= \widetilde{Z}^\top \Sigma^\top \overline{Q}(-g) \quad \text{from (54)} \\ &= 0, \end{aligned}$$

where the last step follows from the fact that only the first n_x rows of g are nonzero (see (13)), and the first n_x columns of $Z_c^\top \Sigma^\top \overline{Q}$ are zero (see (18) and (45)). Thus, (57) and (58) are identical and, together with (54)–(55), yield the same solution for w_y, w_z, from which we can compute the primal solution w of the QP. ∎

4.3. Recovering the dual solution. Recovering the Lagrange multipliers of the original problem is possible also for the case of active inequalities. Suppose we are sufficiently close to a regular solution of NLP (39) such that QP (42) has a regular solution whose active set is the same as the one from the solution $y^\star$ of the NLP, as in Lemma 4.2. Supposing we have identified the correct active set, we can write the QP as in (43). The corresponding Lagrangian function and its gradient are

$$\mathcal{L}(w,\lambda,\mu) := \frac{1}{2} w^\top H w + \lambda^\top (Gw + g) + \mu_{\text{act}}^\top (G_{\text{act}} w + g_{\text{act}}), \tag{60}$$

$$\nabla_w \mathcal{L}(w,\lambda,\mu) = Hw + G^\top \lambda + G_{\text{act}}^\top \mu_{\text{act}}, \tag{61}$$

where we define $g_{\text{act}} := c_i(\overline{y}), i \in \mathcal{A}(y^\star)$, and μ_{act} are the multipliers corresponding to the active inequalities.

Multipliers of active inequality constraints. Using the definitions of $\widetilde{G}, \widetilde{Z}, Z_c$ as in (45), and stating $\nabla_w \mathcal{L}(w^\star, \lambda^\star, \mu^\star) = 0$, we can write

$$(G_{\text{act}} Z_c)^\top \mu_{\text{act}}^\star = -Z_c^\top (Hw^\star), \tag{62}$$

where we multiplied $\nabla_w \mathcal{L}(w^\star, \lambda^\star, \mu^\star) = 0$ from the left with $Z_c^\top$. Note that the matrix $G_{\text{act}} Z_c$ is invertible, for the same reasons as given in the proof of Lemma 4.5. Substituting the original Hessian H by the convexified Hessian $\widetilde{H} = H + U + \gamma G_{\text{act}}^\top G_{\text{act}}$ gives us an expression for the multipliers corresponding to the active inequalities of the convexified problem:

$$(G_{\text{act}} Z_c)^\top \mu_{\text{conv,act}}^\star = -Z_c^\top (Hw^\star + Uw^\star + \gamma G_{\text{act}}^\top G_{\text{act}} w^\star). \tag{63}$$

For QPs with OCP structure, as in (2), it holds that

$$\begin{aligned} Z_c^\top U w^\star &= \underbrace{Z_c^\top G^\top}_{=0} \overline{Q}(G + \Sigma) w^\star + Z_c^\top \Sigma^\top \overline{Q} G w^\star \\ &= Z_c^\top \Sigma^\top \overline{Q}(-g) \\ &= 0, \end{aligned}$$

where we used a similar argument as in the proof of Theorem 4.8.

Comparing (62) and (63), we can recover the correct multipliers of the original problem as

$$\mu_{\text{act}}^\star = \mu_{\text{conv,act}}^\star + \gamma G_{\text{act}} w^\star. \tag{64}$$

We remark that the multipliers of the inequalities can be recovered in a stagewise fashion, as with the multipliers corresponding to the equality constraints, as shown in Algorithm 5.

Algorithm 5. Recovery of Lagrange multipliers: Inequality constrained case.

Input: w, $\mu_{\text{conv,act}}$

Output: λ, μ_{act}

1: $\mu_{\text{act},N} \leftarrow \mu_{\text{conv,act},N} + \gamma G_{\text{act},N}\, x_N$
2: $\lambda_N \leftarrow \widehat{Q}_N x_N + C_N^\top \mu_{\text{act},N}$
3: **for** $k = N-1, \ldots, 0$ **do**
4: $\quad \mu_{\text{act},k} \leftarrow \mu_{\text{conv,act},k} + \gamma G_{\text{act},k}\, w_k$
5: $\quad \lambda_k \leftarrow Q_k x_k + S_k^\top u_k + A_k^\top \lambda_{k+1} + C_{k,x}^\top \mu_{\text{act},k}$
6: **end for**

Multipliers of equality constraints. We need to make a small modification to Algorithm 3 in order to recover the correct multipliers of the equality constraints, because they depend on the multipliers of the active inequality constraints. With the notation of QP (2), the procedure is shown in Algorithm 5.

To summarize, we first compute the primal solution; the QP solver provides us with $\mu_{\text{conv,act}}$, with which we compute the multipliers corresponding to the active inequalities and the equality constraints.

4.4. Local convergence of SQP method with structure-preserving convexification. Since our convexification method locally does not alter the primal solution, and we can correctly recover the dual solution, a full step SQP algorithm that employs our structure-preserving convexification algorithm converges quadratically under some assumptions, as is established in the next theorem. Note that this is a local convergence result in a neighborhood of a minimizer, while global convergence results require additional globalization strategies as discussed in [21].

THEOREM 4.9. *Regard NLP* (1) *with a regular solution* $z^\star = (y^\star, \lambda^\star, \mu^\star)$. *Then there exist* $\delta > 0$, $\gamma > 0$, *and* $\epsilon > 0$ *so that for all* $z_0 = (y_0, \lambda_0, \mu_0)$ *with* $\|z_0 - z^\star\| < \epsilon$, *a full step SQP algorithm with Hessian approximation* $\widetilde{H} = \mathcal{H}(\nabla^2_{yy}\mathcal{L}(z), \delta, \gamma, \mathcal{A}(y^\star))$ *converges, the QP subproblems are convex, and the convergence rate is quadratic.*

Proof. We start the iterations with the optimal active set $\mathcal{A}(y^\star)$ and the exact Hessian at z_0. By convexifying the exact Hessian, we do not alter the primal solution, as established in Theorem 4.8. Moreover, by using Algorithm 5 we also recover the correct dual step. Therefore, by using our convexification approach we take the same primal-dual steps as the exact-Hessian SQP method while solving convex QP subproblems. The quadratic convergence then follows from a standard convergence proof of Newton's method, e.g., Theorem 3.5 in [21]. ◻

Remark 4.10. In Theorem 4.9 we assume that the SQP iterations start by using the optimal active set of the NLP solution. We remark that this assumption is not very restrictive: it is a standard property (see, e.g., [4]) that if z_0 is close enough to $z^\star$, the QP subproblem even with inexact positive definite Hessian matrix will identify the correct active set. This is further illustrated by the numerical experiments in section 6.

5. Dealing with indefinite reduced problems. In the previous sections, we assumed a positive reduced Hessian. We call problems with an indefinite reduced Hessian $\widetilde{Z}^\top H \widetilde{Z} \nsucc 0$ *indefinite reduced* problems. In order to compute a positive definite Hessian approximation, we need to introduce some regularization. There are different heuristics of doing this which modify the problem in different ways to allow a unique global solution. We present one alternative here.

5.1. Regularization of the Hessian. Consider Algorithm 4. For an indefinite reduced problem, it is possible that $\widehat{R}_k \nsucc 0$ holds in line 5 of the algorithm, such that the Schur complement lemma no longer holds. Instead, we still compute $\widehat{H}_k$ but regularize this Hessian block by removing all negative eigenvalues and replacing them with slightly positive eigenvalues. We call this action the "projection" of the eigenvalues as in (38), where $\epsilon > 0$ is some small positive number. Applying this regularization results in Algorithm 6, which is based on Algorithm 4 but includes an if-clause checking for positive definiteness of $\widehat{R}_k$.

Remark 5.1. In Theorem 4.7, we establish that there exists some $\gamma > 0$, such that there exists a positive definite convexified Hessian. In an SQP setting, when we are close to a local solution of NLP (39), the active set of NLP (39) and QP (42) are

Algorithm 6. Structure-preserving convexification for inequality constrained optimization, including the regularization of Hessian blocks.

Input: $H, \delta, \gamma, \epsilon$, current active set $\mathcal{A}$

Output: $\widetilde{H}$

1: $\widetilde{Q}_N = \delta I$

2: $\overline{Q}_N = \widehat{Q}_N + \gamma G_{\mathrm{act},N}^\top G_{\mathrm{act},N} - \widetilde{Q}_N$

3: **for** $k = N-1, \ldots, 0$ **do**

4: $\quad \begin{bmatrix} \widehat{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix} := \begin{bmatrix} Q_k & S_k^\top \\ S_k & R_k \end{bmatrix} + \begin{bmatrix} A_k^\top \overline{Q}_{k+1} A_k & A_k^\top \overline{Q}_{k+1} B_k \\ B_k^\top \overline{Q}_{k+1} A_k & B_k^\top \overline{Q}_{k+1} B_k \end{bmatrix} + \gamma G_{\mathrm{act},k}^\top G_{\mathrm{act},k}$

5: $\quad$ **if** $\widehat{R}_k \nsucc 0$ **then**

6: $\qquad \breve{H}_k := \begin{bmatrix} \breve{Q}_k & \breve{S}_k^\top \\ \breve{S}_k & \breve{R}_k \end{bmatrix} = \mathrm{project}(\widehat{H}_k, \epsilon)$

7: $\quad$ **else**

8: $\qquad \breve{H}_k := \begin{bmatrix} \breve{Q}_k & \breve{S}_k^\top \\ \breve{S}_k & \breve{R}_k \end{bmatrix} = \begin{bmatrix} \widehat{Q}_k & \widehat{S}_k^\top \\ \widehat{S}_k & \widehat{R}_k \end{bmatrix}$

9: $\quad$ **end if**

10: $\quad \widetilde{Q}_k := \breve{S}_k^\top \breve{R}_k^{-1} \breve{S}_k + \delta I$

11: $\quad \widetilde{H}_k := \begin{bmatrix} \widetilde{Q}_k & \breve{S}_k^\top \\ \breve{S}_k & \breve{R}_k \end{bmatrix}$

12: $\quad \overline{Q}_k = \breve{Q}_k - \widetilde{Q}_k$

13: **end for**

14: $\widetilde{H} := \mathrm{diag}(\widetilde{H}_0, \ldots, \widetilde{Q}_N)$

identical and, in principle, we can choose γ arbitrarily big. However, when we are at a point that is far from a local solution and the active set is not stable yet, large values for γ might result in poor convergence. A possible heuristic as an alternative to a fixed value γ is motivated by referring to Algorithm 6. In line 5 of the algorithm, we check for positive definiteness of $\widehat{R}_k$. In line 4, we will try to make this matrix positive definite by adding a matrix $G_{\mathrm{act},k}^\top \Gamma G_{\mathrm{act},k}$, where we compute Γ as follows. Consider a decomposition of matrix $R_k + B_k^\top \overline{Q}_{k+1} B_k$ in a basis for the range space $Y_{\mathrm{act},k}$ and null space $Z_{\mathrm{act},k}$ of $G_{\mathrm{act},k}$. A necessary condition for the positive definiteness of $\widehat{R}_k$, by the Schur complement lemma, is then

$$Y_{\mathrm{act},k}^\top (R_k + B_k^\top \overline{Q}_{k+1} B_k + G_{\mathrm{act},k}^\top \Gamma G_{\mathrm{act},k}) Y_{\mathrm{act},k} \succ 0, \tag{65}$$

such that we could propose the expression for Γ

$$\Gamma = -(Y_{\mathrm{act},k}^\top G_{\mathrm{act},k}^\top)^{-1} Y_{\mathrm{act},k}^\top (R_k + B_k^\top \overline{Q}_{k+1} B_k) Y_{\mathrm{act},k} (G_{\mathrm{act},k} Y_{\mathrm{act},k})^{-1} + \gamma I \tag{66}$$

for some $\gamma > 0$, where we used the fact that $G_{\mathrm{act},k} Y_{\mathrm{act},k}$ is invertible by construction. Note that (66) is a heuristic choice for Γ in the sense that the above condition is only necessary, i.e., you still might have to apply regularization on $\widehat{H}_k$, as in line 6 of Algorithm 6.

5.2. Recovering Lagrange multipliers. By applying our structure-preserving regularization method that is based on the convexification method of section 3.2, we have $\mathrm{QP}(\widetilde{H})$ resulting in a different primal and dual solution than the original problem. We need to recover the dual solution with respect to the modified problem, i.e., without the backward transfer of cost but including the extra convexity introduced by the "project" operation in line 6 of Algorithm 6. In order to do so, we do not start from the original Hessian as in Algorithm 5. Instead, we make use of the Hessian with the regularization terms added, but without the cost transfer terms. In other words, we keep a separate *modified* Hessian, which consists of the following blocks:

$$H_{\mathrm{mod}} = H + \Delta H = H + \mathrm{diag}(\Delta H_0, \ldots, \Delta H_{N-1}, 0), \tag{67}$$

where $\Delta H_k = 0$ when there was no regularization and $\Delta H_k = \check{H}_k - \widehat{H}_k$ otherwise. We then apply Algorithm 5 to H_{mod} instead of H.

6. Numerical example. In this section, we offer a numerical example as an illustration of the practical use of our convexification method. We will solve a nonlinear OCP on an inverted pendulum. This system, depicted in Figure 2, consists of a rod of length l making an angle θ with the vertical axis, attached to a cart with mass M that can move horizontally only, driven by a force F. At the end of the rod is a ball of mass m.

The dynamics of the inverted pendulum are described by the following ODE, where p, v are the horizontal displacement and horizontal velocity, respectively, θ is the angle with the vertical (see Figure 2), and ω is the corresponding angular velocity:

$$\dot{p} = v, \tag{68a}$$

$$\dot{\theta} = \omega, \tag{68b}$$

$$\dot{v} = \frac{-ml\sin(\theta)\dot{\theta}^2 + mg\cos(\theta)\sin(\theta) + F}{M + m - m(\cos(\theta))^2}, \tag{68c}$$

$$\dot{\omega} = \frac{-ml\cos(\theta)\sin(\theta)\dot{\theta}^2 + F\cos(\theta) + (M+m)g\sin(\theta)}{l(M + m - m(\cos(\theta))^2)}. \tag{68d}$$

The control objective is to swing up the ball ($\theta = 0$), starting with the rod hanging vertically down, $\theta = \pi$. We collect the states in the state vector $s := [p, \theta, v, \omega]^\top$. A multiple-shooting discretization of the control problem corresponds to the following OCP formulation:

$$\underset{\substack{s_0,\ldots,s_N,\\ F_0,\ldots,F_{N-1}}}{\text{minimize}} \quad \sum_{k=0}^{N-1} \begin{bmatrix} s_k \\ F_k \end{bmatrix}^\top \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} s_k \\ F_k \end{bmatrix} + s_N^\top Q s_N \tag{69a}$$

$$\text{subject to} \quad s_{k+1} = \phi_k(s_k, F_k), \qquad k = 0, \ldots, N-1, \tag{69b}$$

$$-80 \leqslant F_k \leqslant 80, \qquad k = 0, \ldots, N-1, \tag{69c}$$

$$s_0 = \bar{s}_0, \tag{69d}$$

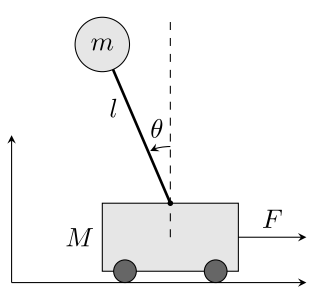

FIG. 2. *Schematic illustrating the inverted pendulum on top of a cart.*

TABLE 1
Exact-Hessian SQP iterations for the pendulum example (N=100), using the structure-preserving convexification method from Algorithm 6.

It.	KKT norm	Step size	$H \succ 0$	$\widetilde{Z}^\top H \widetilde{Z} \succ 0$	$Z^\top HZ \succ 0$	Regs.	Act. set chgs.
1	1.52e+02	7.39e+02	True	True	True	0	74
2	5.33e+06	9.63e+02				9	10
3	2.02e+06	6.01e+02				9	1
4	1.47e+06	3.27e+02				5	4
5	9.44e+05	3.22e+02				4	0
6	3.75e+05	4.87e+02				3	7
7	1.04e+05	4.93e+02				1	22
8	9.13e+03	3.23e+02				2	14
9	3.90e+02	4.14e+02		True		0	2
10	1.25e+02	9.53e+01		True		0	5
11	6.66e+00	3.21e+00		True		0	0
12	1.38e-03	2.88e-03		True		0	0
13	2.02e-08	5.33e-09		True		0	0
14	1.42e-10	6.99e-11		True		0	0

where ϕ denotes a numerical integration method (explicit Runge–Kutta method of order 4) to simulate the continuous-time dynamics in (68) over one shooting interval, the weight matrices are chosen as $Q = \mathrm{diag}([1000, 1000, 0.01, 0.01]), R = 0.01$. Because our aim is to swing the pendulum up, we selected strong weights on the position and angle. The other states and the control are assigned a weak penalty in order to avoid too-abrupt swing-ups and to favor smooth trajectories. Note that the weighting matrices Q and R are tuning parameters used by the control engineer in the design process in order to obtain a desired behavior. Different choices are therefore equally valid. The initial value is $\bar{s}_0 = [0, \pi, 0, 0]^\top$. We choose $N = 100$ shooting intervals of length 0.01 s.

We solve NLP (69) with a full-step SQP method, starting from $z_0 = (0, 0, 0)$, i.e., we assume all inequality constraints *inactive*. In each iteration, we apply our convexification method. We choose the following values for the parameters: $\delta = 1 \cdot 10^{-4}, \gamma = 1$. In Table 1, the iterations are given. The SQP scheme converges in 14 steps given a tolerance of 10^{-8}. Only in the first iteration the Hessian matrix is positive definite. At the solution, only the reduced Hessian is positive definite. Whenever the reduced Hessian is not positive definite, we need to apply regularization as in Algorithm 6. This is denoted in Table 1 with the amount of shooting intervals in which we needed to regularize in the next to last column. The number of active set changes in each iteration is listed in the rightmost column.

It is interesting to remark that whenever the reduced Hessian $\widetilde{Z}^\top H \widetilde{Z}$ is positive definite, but $Z^\top HZ \not\succ 0$, we do not need to regularize thanks to the terms $\gamma G_{\mathrm{act},k}^\top G_{\mathrm{act},k}$ coming from the active inequality constraints in each stage k. However, please note that adding this term when we are still far from the NLP solution adds extra regularization, as the correct active set has not been identified yet. In the case the reduced Hessian is not positive definite and as a consequence we have to regularize, we only need to do so at maximum 9 intervals of the 100 control intervals (in iterations 2 and 3; see Table 1).

We compare these results obtained with the structure-preserving convexification against two other regularization methods, namely, *project* and *mirror* as described in (38), applied directly to each Hessian block in order to preserve the OCP structure,

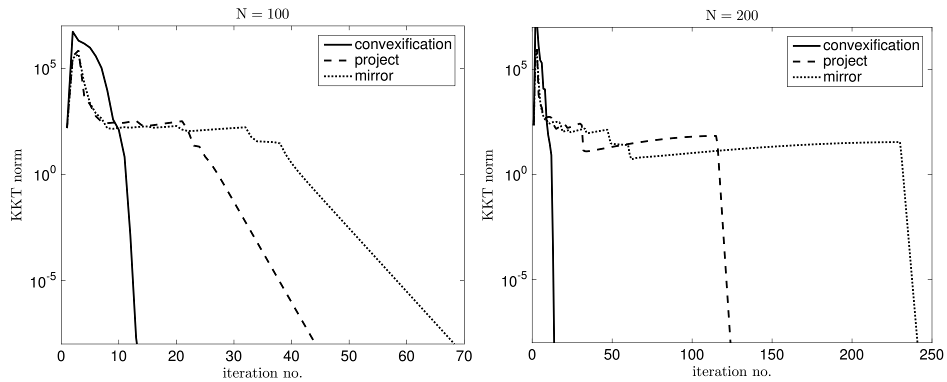

FIG. 3. *Comparison of the convergence of an SQP-type method applied to obtain the solution to NLP* (69)*, with three different regularization methods. We compare two OCP instances: on the left* $N = 100$*, and on the right* $N = 200$.

where we choose $\epsilon = \delta = 1 \cdot 10^{-4}$. The comparison in convergence is made in Figure 3. As can be seen, using the convexification as a regularization method yields faster convergence, namely, convergence in less than half the number of iterations of regularization by projection, using a tolerance of 10^{-8}. Moreover, we obtain quadratic convergence, as we established in Theorem 4.9, when using the convexification method once the optimal active set is found (see Table 1). By contrast, the other regularization methods result in linear convergence. For different horizon lengths, e.g., $N = 50, 150, 200$, the convergence behavior of the structure-preserving convexification method is similar to the one reported in Table 1 and we obtain similar convergence profiles for the other methods (see Figure 3, right side).

We remark that more advanced regularization schemes than the two that we are comparing to would yield similar convergence rates, e.g., the methods in [16] or in [30]. Those methods, however, do not fulfill the desired properties (as mentioned in the introduction and section 3.7) of the regularization schemes. The possibility of combining our approach with the one of [16] is the subject of ongoing research.

7. Conclusions. In this paper, we presented a structure-preserving convexification procedure for indefinite QPs arising from solving nonlinear OCPs using SQP. We proved that there is an equivalence between the existence of a convexified Hessian and the reduced Hessian being positive definite, which result in equal primal solutions. Furthermore, we offered an algorithm that constructs such a convexified Hessian with the same structure as the original Hessian and recovers the dual solution of the original problem. Doing so, we retain a locally quadratic rate of convergence using full steps in the SQP algorithm.

In the case the reduced Hessian is not positive definite, we proposed a regularization method based on the convexification. We illustrated our findings with a numerical example, which consists of solving a nonlinear OCP with an SQP-type method. Possible regularization methods were compared for the indefinite reduced case.

Further research will aim at comparing the computational complexity of condensing based methods against the convexification method presented here. Furthermore, we aim at an efficient implementation of the convexification algorithm, coupled with existing structure-exploiting QP solvers that work only with positive definite block diagonal Hessians. Finally, the automatic selection of parameters δ, γ, ϵ will be investigated in future research.

REFERENCES

[1] D. Axehill and M. Morari, *An alternative use of the Riccati recursion for efficient optimization*, Systems Control Lett., 61 (2012), pp. 37–40.

[2] T. J. Besselmann, S. V. de Moortel, S. Almér, P. Jörg, and H. J. Ferreau, *Model predictive control in the multi-megawatt range*, IEEE Trans. Industrial Electronics, 63 (2015), pp. 4641–4648.

[3] H. G. Bock and K. J. Plitt, *A multiple shooting algorithm for direct solution of optimal control problems*, in Proceedings of the IFAC World Congress, Pergamon Press, Elmsford, NY, 1984, pp. 242–247.

[4] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, Acta Numer., 4 (1995), pp. 1–51.

[5] M. Diehl, H. G. Bock, J. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer, *Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations*, J. Process Control, 12 (2002), pp. 577–585.

[6] M. Diehl, H. J. Ferreau, and N. Haverbeke, *Efficient numerical methods for nonlinear MPC and moving horizon estimation*, in Nonlinear Model Predictive Control, L. Magni, M. Raimondo, and F. Allgöwer, eds., Lecture Notes in Control and Inform. Sci. 384, Springer, New York, 2009, pp. 391–417.

[7] A. Domahidi, A. Zgraggen, M. Zeilinger, M. Morari, and C. Jones, *Efficient interior point methods for multistage problems arising in receding horizon control*, in Proceedings of the IEEE Conference on Decision and Control, Maui, HI, 2012, pp. 668–674.

[8] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, *qpOASES: A parametric active-set algorithm for quadratic programming*, Math. Program. Comput., 6 (2014), pp. 327–363.

[9] A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152.

[10] R. Franke and E. Arnold, *Applying new numerical algorithims to the solution of discrete-time optimal control problems*, in Computer Intensive Methods in Control and Signal Processing: The Curse of Dimensionality, K. Warwick and M. Kárný, eds., Springer, New York, 1997, pp. 105–117.

[11] J. V. Frasch, S. Sager, and M. Diehl, *A parallel quadratic programming method for dynamic optimization problems*, Math. Program. Comput., 7 (2015), pp. 289–329.

[12] G. Frison and J. B. Jørgensen, *A fast condensing method for solution of linear-quadratic control problems*, in Proceedings of the IEEE Conference on Decision and Control, 2013, pp. 7715–7720.

[13] G. Frison, H. B. Sorensen, B. Dammann, and J. B. Jørgensen, *High-performance small-scale solvers for linear model predictive control*, in Proceedings of the European Control Conference, 2014, pp. 128–133.

[14] P. Gill, W. Murray, and M. Saunders, *User's Guide for QPOPT* 1.0*: A Fortran Package for Quadratic Programming*, 1995, http://www.cam.ucsd.edu/~peg/papers/qpopt.pdf.

[15] P. Gill, W. Murray, and M. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Rev., 47 (2005), pp. 99–131, https://doi.org/10.1137/S0036144504446096.

[16] P. E. Gill and D. P. Robinson, *A globally convergent stabilized sqp method*, SIAM J. Optim., 23 (2013), pp. 1983–2010.

[17] N. Gould and D. Robinson, *A second derivative sqp method: Local convergence and practical issues*, SIAM J. Optim., 20 (2010), pp. 2049–2079, https://doi.org/10.1137/080744554.

[18] E. V. Haynsworth, *Determination of the inertia of a partitioned hermitian matrix*, Linear Algebra Appl., 1 (1968), pp. 73–81.

[19] C. Kirches, L. Wirsching, S. Sager, and H. Bock, *Efficient numerics for nonlinear model predictive control*, in Recent Advances in Optimization and its Applications in Engineering, Springer, New York, 2010, pp. 339–357, https://doi.org/10.1007/978-3-642-12598-0_30.

[20] D. Murray and S. J. Yakowitz, *Differential dynamic programming and Newton's method for discrete optimal control problems*, J. Optim. Theory Appl., (1984), pp. 395–414.

[21] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer Ser. Oper. Res. Financ. Eng., 2nd ed., Springer, New York, 2006.

[22] S. Qin and T. Badgwell, *A survey of industrial model predictive control technology*, Control Eng. Practice, 11 (2003), pp. 733–764.

[23] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl, *Autogenerating microsecond solvers for nonlinear MPC: a tutorial using ACADO integrators*, Optim. Control Appl. Methods, 36 (2014), pp. 685–704.

[24] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*, Nob Hill, Madison, WI, 2009.

[25] S. Robinson, *Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms*, Math. Program., 7 (1974), pp. 1–16.

[26] C. Schmid and L. T. Biegler, *Quadratic programming methods for reduced Hessian SQP*, Comput. Chem. Eng., 18 (1994), pp. 817–832.

[27] N. van Duijkeren, R. Verschueren, G. Pipeleers, M. Diehl, and J. Swevers, *Path-following nmpc for serial-link robot manipulators using a path-parametric system reformulation*, in Proceedings of the European Control Conference, 2016.

[28] R. Verschueren, M. Zanon, R. Quirynen, and M. Diehl, *Time-optimal race car driving using an online exact Hessian based nonlinear MPC algorithm*, in Proceedings of the European Control Conference, 2016.

[29] 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 IEEE Conference on Decision and Control, 2013, pp. 5113–5118.

[30] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), pp. 25–57.

[31] W. Wan and L. T. Biegler, *Structured regularization for barrier NLP solvers*, Comput. Optim. Appl., 66 (2017), pp. 401–424.