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 (SQP) 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 (SOSC) 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. Secondly, 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.

1. Introduction. In the last 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) [25] 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 [23]. Recently, both algorithmic and computational advances made NMPC applicable in real-time for systems with fast dynamics, e.g. in [2, 24, 28, 30]. For an overview of efficient methods for NMPC, we point the reader to [6, 20].

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

---

†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. (amario@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)
approach to efficiently solve the parametric optimization problem depending on the current state of the system [6].

There exist several approaches to solve 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 [27], QPOPT [15], qpOASES [8], to obtain the solution and afterwards 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, 13].

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 lengths, in which case the condensing approach is less competitive [30]. Examples of structure-exploiting QP solvers tailored to optimal control are qpDUNES [12, 11], FORCES [7] and HPMPC [14]. 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, qpDUNES require a positive definite Hessian matrix, the second order sufficient conditions (SOSC) for optimality require positive definite-ness only of the reduced Hessian, which is defined to be the Hessian matrix projected onto the null space of the active constraints [22]. We point out that HPMPC and FORCES would be in principle 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 our 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 afterwards. A numerical case study comparing these two alternatives is presented in [29]. 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 [22]. In [21], 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 [22]. Quasi-Newton methods can generally be modified to directly provide a positive definite Hessian approximation, see e.g. [16, 18].

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 [31], that 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 the case of redundant constraints is presented in [32]. One interesting alternative method of dealing with indefiniteness is presented in [17], 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. 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; (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 [22]. 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 [17] 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 optimal control problem structure.
**Definition 1** (NLP with OCP structure).

\[(1a) \quad \text{minimize} \quad \sum_{k=0}^{N-1} f_k(s_k, q_k) + f_N(s_N)\]

\[(1b) \quad \text{subject to} \quad s_{k+1} = \phi_k(s_k, q_k), \quad k = 0, \ldots, N-1,\]

\[(1c) \quad s_0 = s_0,\]

\[(1d) \quad 0 \geq c_k(s_k, q_k), \quad k = 0, \ldots, N-1,\]

\[(1e) \quad 0 \geq c_N(s_N).\]

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 \(s_0 \in \mathbb{R}^{n_x}\) 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** (QP with OCP structure).

\[(2a) \quad \text{QP}(H) : \quad \text{minimize} \quad \sum_{k=0}^{N-1} \frac{1}{2} \left[ x_k^T H_k \begin{bmatrix} Q_k & S_k & R_k \end{bmatrix} \begin{bmatrix} x_k \ u_k \end{bmatrix} + \frac{1}{2} x_N^T \hat{Q}_N x_N \right] \]

\[(2b) \quad \text{subject to} \quad x_{k+1} = A_k x_k + B_k u_k, \quad k = 0, \ldots, N-1,\]

\[(2c) \quad x_0 = x_0,\]

\[(2d) \quad 0 \geq C_{k,x} x_k + C_{k,u} u_k, \quad k = 0, \ldots, N-1,\]

\[(2e) \quad 0 \geq C_N x_N,\]

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 \(Q_k, \hat{Q}_N \in \mathbb{R}^{n_x \times n_x}, S_k \in \mathbb{R}^{n_u \times n_x}, R_k \in \mathbb{R}^{n_u \times n_u}\) and the Hessian matrix \(H := \text{diag}(H_0, \ldots, H_{N-1}, \hat{Q}_N)\). The constraints denote, respectively, dynamic constraints with matrices \(A_k \in \mathbb{R}^{n_x \times n_x}, B_k \in \mathbb{R}^{n_x \times n_u}\), dynamic constraints with \(C_{k,x} \in \mathbb{R}^{n_c,k \times n_x}, C_{k,u} \in \mathbb{R}^{n_c,k \times n_u}\), and an initial constraint with \(x_0 \in \mathbb{R}^{n_x}\). Note that this compact notation, as proposed in e.g. [14], 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 the 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 the 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 (Equality constrained QP).

\begin{align}
\text{(3a)} & \quad \text{minimize} \quad \frac{1}{2} w^T H w \\
\text{(3b)} & \quad \text{subject to} \quad G w + g = 0,
\end{align}

with constraint matrix $G \in \mathbb{R}^{p \times n}$ of full rank $p$ with $p \leq n$. Note that the linear independence constraint qualification (LICQ) requires full rank of $G$ [22]. 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:

\begin{align}
\mathbb{S}^n := \{ X \in \mathbb{R}^{n \times n} \mid X = X^T \}.
\end{align}

In the case of the 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: $w^T = [x_0^T, u_0^T, \ldots, x_N^T]$.

Definition 4 (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:

\begin{align}
\text{(5a)} & \quad G Z = 0, \\
\text{(5b)} & \quad (Y | Z)^T (Y | Z) = I.
\end{align}

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

3.1. A characterization of convexity. Using Definitions 3-4, 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 [22].

Theorem 5. Consider QP (3) and Definition 4, assuming that LICQ holds. Then QP (3) has a unique global optimum if and only if the reduced Hessian is positive definite, i.e.

\begin{align}
\text{(6)} & \quad Z^T H Z > 0.
\end{align}

We note that the reduced Hessian being positive definite corresponds to the second order sufficient condition (SOSC) for optimality, as defined in [22]. A well-known fact related to this SOSC is stated in the following theorem (for a proof, see e.g. [22]).

Theorem 6. Consider QP (3) with arbitrary $H \in \mathbb{S}^n$, Definition 4 and assume that LICQ holds. Then

\begin{align}
\text{(7)} & \quad Z^T H Z > 0 \iff \exists \gamma \in \mathbb{R} : H + \gamma G^T G > 0.
\end{align}

Theorem 6 is at the basis of the augmented Lagrangian method for optimization, where one typically calls $\gamma > 0$ the quadratic penalty parameter. By choosing $\gamma$ large enough one can always create a positive definite Hessian at points satisfying SOSC. Note that $H + \gamma G^T G$ 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 [17]. Theorem 6 can be generalized as follows:
Theorem 7 (Revealing Convexity). Consider QP (3), Definition 4, and assume LICQ holds. The reduced Hessian satisfies

\[ Z^T H Z > 0 \]

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

\[ Z^T U Z = 0 \]

such that

\[ H + U > 0. \]

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 \( p Y \mid Z q \). Doing so, matrix inequality (8c) is equivalent to \( (Y^T Z^T H + U) (Y^T Z) > 0 \). This again, due to (8b) and the Schur complement lemma [19], is equivalent to

\[ Z^T H Z > 0, \]

\[ Y^T (H + U) Y > Y^T (H + U) Z \cdot (Z^T H Z)^{-1} \cdot Z^T (H + U) Y. \]

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

\[ U = Y K Y^T + Y M Z^T + Z M^T Y^T, \]

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

\[ K > -Y^T H Y + (Y^T Z^T M + M) \cdot (Z^T H Z)^{-1} \cdot (Y^T Z^T M + M)^T. \]

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 6 is obtained by choosing \( M = 0 \), i.e. \( U = Y K Y^T \), and observing that \( Y \) is a basis of the range space of \( G^T \). 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 7, where we impose an OCP structure on \( U \), which cannot be obtained by \( U = \gamma G^T G \) or its generalization \( U = G^T \Gamma G \). 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 \( H \).

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 \( \hat{H} \in \mathbb{S}^{N_{x},n_{x}} \) 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:

\[ \mathbb{S}_{OCP}^{N_{x},n_{x}} := \left\{ X \in \mathbb{S}^{N_{x},n_{x}} \mid X = \text{diag}(X_0, \ldots, X_N), X_k \in \mathbb{S}^{n_{x}}, k = 0, \ldots, N-1, X_N \in \mathbb{S}^{n_{x}} \right\}. \]
The convexification can be performed by using the structure of the equality constraints. The resulting modified QP(\(\tilde{H}\)) has two important properties: (a) the primal solutions of QP(\(H\)) and QP(\(\tilde{H}\)) are equal; (b) \(\tilde{H}\) 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 8. Afterwards, we present the convexification procedure in detail and state (b) in Theorem 12, 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(\(\tilde{H}\)) 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)):

\[
G := \begin{bmatrix} -I & A_0 & B_0 & -I \\ & \ddots & \ddots & \ddots \\ & & -I & A_{N-1} & B_{N-1} \end{bmatrix}, \quad g := \begin{bmatrix} \tau_0 \\ \vdots \\ 0 \end{bmatrix},
\]

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 4.

### 3.3. Transfer of cost between stages

The \(N\) stages of QP (2a)-(2c) are coupled by the dynamic constraints \(x_{k+1} = A_kx_k + B_ku_k\). 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:

\[
\bar{\eta}_k(x) := x^\top \bar{Q}_k x, \quad k = 0, \ldots, N,
\]

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

\[
\begin{align}
l_k(x_k, u_k) & := \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top H_k \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \quad k = 0, \ldots, N-1, \\
L_k(x_k, u_k) & := l_k(x_k, u_k) - \bar{\eta}_k(x_k) + \bar{\eta}_{k+1}(A_kx_k + B_ku_k), \\
L_k(x_k, u_k) & := \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \tilde{H}_k \begin{bmatrix} x_k \\ u_k \end{bmatrix}, \quad k = 0, \ldots, N-1,
\end{align}
\]

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

**Theorem 8** (Equality of primal QP solutions). *Consider the equality constrained QP (2a)-(2c) and assume that \(Z^\top HZ > 0\). Then, the primal solutions of QP(\(H\)) and QP(\(\tilde{H}\)) are equal.*

*Proof.* By assumption \(Z^\top HZ > 0\), therefore QP(\(H\)) has a unique global mini-
mum, by Theorem 5. The cost function of QP(\(\tilde{H}\)) satisfies
\[
\sum_{k=0}^{N-1} L_k(x_k, u_k) + x_N^T \tilde{Q}_N x_N
\]
\[
= \sum_{k=0}^{N-1} I_k(x_k, u_k) - \varpi_k(x_k) + \varpi_{k+1}(A_k x_k + B_k u_k) + x_N^T \tilde{Q}_N x_N,
\]
\[
= \sum_{k=0}^{N-1} I_k(x_k, u_k) - \varpi_0(x_0) + \varpi_N(x_N) + x_N^T \tilde{Q}_N x_N
\]
\[
= \sum_{k=0}^{N-1} I_k(x_k, u_k) + x_N^T \hat{Q}_N x_N - \varpi_0(x_0),
\]
which is equal to the cost function of QP(\(H\)), up to the constant term \(-\varpi_0(x_0)\). It follows, because the constraints of QP(\(H\)) and QP(\(\tilde{H}\)) are identical, that the primal solutions of both problems coincide. \(\square\)

Note that we can write \(\tilde{H} = H + U(\overline{Q})\) where we let \(U(\overline{Q}) := \text{diag}(U_0, \ldots, U_N)\), with \(Q := \text{diag}(Q_0, \ldots, Q_N)\) and the quantities \(U_k, k = 0, \ldots, N\) are defined as follows:

\[
U_k := \begin{bmatrix}
A_k^T \overline{Q}_{k+1} A_k - \overline{Q}_k & A_k^T \overline{Q}_{k+1} B_k \\
B_k^T \overline{Q}_{k+1} A_k & B_k^T \overline{Q}_{k+1} B_k
\end{bmatrix}, \quad U_N := -Q_N.
\]

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

**Algorithm 1** \(U(\overline{Q})\)

**Input:** \(\overline{Q}\)

**Output:** \(U\)

1. \(U_N := -Q_N\)
2. for \(k = N - 1, \ldots, 0\) do
3. \(U_k := \begin{bmatrix}
A_k^T \overline{Q}_{k+1} A_k - \overline{Q}_k & A_k^T \overline{Q}_{k+1} B_k \\
B_k^T \overline{Q}_{k+1} A_k & B_k^T \overline{Q}_{k+1} B_k
\end{bmatrix}\)
4. end for
5. \(U := \text{diag}(U_0, \ldots, U_N)\)

Theorem 8 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 9.** Consider the equality constrained QP (2a)-(2c), \(Z\) from Definition 4 and \(U\) from (16) and assume LICQ is satisfied. Then, for any \(\overline{Q}\), it holds that \(Z^T U(\overline{Q}) Z = 0\).

**Proof.** Using Equations (16) and (13), we can rewrite \(U\) as
\[
U(\overline{Q}) = (G + \Sigma)^T \overline{Q} (G + \Sigma) - \Sigma^T \overline{Q} \Sigma,
\]
\[
= G^T \overline{Q} G + G^T \overline{Q} \Sigma + \Sigma^T \overline{Q} G,
\]
\[
= G^T \overline{Q} G + \Sigma^T \overline{Q} (G + \Sigma) = G^T \overline{Q} G + \Sigma^T \overline{Q} \Sigma.
\]
We introduce

\[ \Sigma := \begin{bmatrix} I & 0 & \cdots & 0 \\ 0 & I & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & I \end{bmatrix}, \tag{18} \]

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 \( Z^\top UZ = 0 \).

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

\[ \tilde{Q}_N = \hat{Q}_N + U_N = \hat{Q}_N - \overline{Q}_N. \tag{19} \]

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

\[ \begin{bmatrix} \hat{Q}_k \\ \hat{S}_k \\ \hat{R}_k \end{bmatrix} := \begin{bmatrix} Q_k \\ S_k \\ R_k \end{bmatrix} + \begin{bmatrix} A_k^\top \overline{Q}_{k+1} A_k \\ B_k^\top \overline{Q}_{k+1} B_k \\ 0 & 0 & 0 \end{bmatrix}, \tag{20} \]

and then set

\[ \hat{Q}_k = \hat{Q}_k - \overline{Q}_k, \tag{21} \]

\[ \tilde{H}_k = H_k + U_k = \begin{bmatrix} \hat{Q}_k & \hat{S}_k \\ \hat{S}_k & \hat{R}_k \end{bmatrix}, \quad k = 0, \ldots, N - 1. \tag{22} \]

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

\[
K = Y^\top UY = Y^\top (G^\top \overline{Q}G + G^\top \overline{Q} \Sigma + \Sigma^\top \overline{Q} G) Y,
\]

\[
M = Z^\top UY = Z^\top \Sigma^\top \overline{Q} G Y.
\]

Please note that for arbitrary \( \overline{Q} \), the cost transfer operation presented in this section generally results in indefinite \( \tilde{H} \). In the next section, we will establish Theorem 12, which states that we can find a \( U \in \mathbb{S}^{N,n_x \times n_u}_C \) and corresponding positive definite \( \tilde{H} \), 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 Equations (19)-(22). It computes \( \tilde{H} = H + U(\overline{Q}) \) 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 \( \tilde{Q}_N = \delta I \), with \( \delta > 0 \) a small constant, such that \( \overline{Q}_N = \hat{Q}_N - \delta I \) (lines 1 and 2), and we use this matrix to transfer the cost \( x_K^\top \overline{Q}_N x_N \) to the previous stage. The updated quantities \( \hat{Q}_{N-1}, \hat{S}_{N-1}, \hat{R}_{N-1} \) are calculated according to (20) in line 4. We use the Schur complement lemma [19] in line 5 of the algorithm to ensure that \( \hat{H}_{N-1} > 0 \). Next, we compute \( \overline{Q}_{N-1} = \hat{Q}_{N-1} - \hat{Q}_N \) 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 \( \hat{R}(\delta) := \text{diag}(\hat{R}_0, \ldots, \hat{R}_{N-1}) \) in the output of the algorithm for convenience.
Algorithm 2 Structure-Preserving Convexification: equality constrained case

**Input:** $H, \delta$

**Output:** $\overline{Q}(\delta), \overline{H}(\delta), \tilde{R}(\delta)$

1. $\overline{Q}_N = \delta I$
2. $\overline{Q}_N = \overline{Q}_N - \tilde{Q}_N$
3. for $k = N - 1, \ldots, 0$ do 
   4. $\begin{bmatrix} \tilde{Q}_k \\ \tilde{S}_k \\ \tilde{R}_k \end{bmatrix} = \begin{bmatrix} Q_k \\ S_k \\ R_k \end{bmatrix} + \begin{bmatrix} A_k^T \overline{Q}_{k+1} A_k \\ B_k^T \overline{Q}_{k+1} B_k \\ A_k^T \overline{Q}_{k+1} B_k + B_k^T \overline{Q}_{k+1} A_k \end{bmatrix}$
   5. $\tilde{Q}_k := \tilde{S}_k^T \tilde{R}_k^{-1} \tilde{S}_k + \delta I$
   6. $\tilde{H}_k := \begin{bmatrix} \tilde{Q}_k \\ \tilde{S}_k \\ \tilde{R}_k \end{bmatrix}$
   7. $\overline{Q}_k = \overline{Q}_k - \tilde{Q}_k$
8. end for
9. $\tilde{H} := \text{diag}(\tilde{R}_0, \ldots, \tilde{Q}_N)$

In Theorem 12, we show that Algorithm 2 indeed produces a positive definite $\tilde{H}$, given a sufficiently small value for $\delta$, if and only if the reduced Hessian is positive definite. Lemmas 10 and 11, presented next, help us to prove this result.

**Lemma 10.** Consider the equality constrained QP (2a)-(2c) with OCP structure and Definition 4, assuming LICQ, $Z^T H Z > 0$ hold. Then, Algorithm (2) with $\delta = 0$ delivers positive definite $\tilde{R}(0) > 0$ and positive semi-definite $\tilde{H}(0) \geq 0$.

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

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

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

which is equivalent to

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

By assumption, the reduced Hessian of the full QP (2a)-(2c) is positive definite, which, by Theorem 5, entails that the minimum of the QP is unique. Dynamic programming yields the same solution for each $u_k$, which in turn implies that the Hessian of (24) must be positive definite as well. This amounts to $(R_k + B_k^T \overline{Q}_{k+1} B_k) = \tilde{R}_k > 0$, for $k = 0, \ldots, N - 1$. From the Schur complement lemma, we have that if $\tilde{R}_k > 0$,

(25) \[
\tilde{Q}_k - \tilde{S}_k^T \tilde{R}_k^{-1} \tilde{S}_k \geq 0 \iff \begin{bmatrix} \tilde{Q}_k \\ \tilde{S}_k \\ \tilde{R}_k \end{bmatrix} \geq 0.
\]

With $\delta = 0$, from Algorithm 2 it follows that $\tilde{Q}_k = \tilde{S}_k^T \tilde{R}_k^{-1} \tilde{S}_k$, such that the left hand side of (25) holds. This entails that the Hessian blocks $\tilde{H}_k \geq 0$, such that, together with $\tilde{Q}_N = 0$, it holds that $\tilde{H} \geq 0$. □
In the following, we regard matrices \( \hat{R}_k \) from Algorithm 2 as the map \( \hat{R}(\delta) \) and we recall that \( \hat{R}(\delta) := \text{diag}(\hat{R}_0, \ldots, \hat{R}_{N-1}) \). We use this map in Lemma 11, which will help us prove Theorem 12.

**Lemma 11.** Consider QP (2a)-(2c) and assume that \( Z^\top HZ > 0 \) holds. Then there exists a value \( \delta > 0 \) such that Algorithm 2 computes a positive definite matrix \( \hat{R}(\delta) > 0 \).

**Proof.** Consider the map \( \hat{R}(\delta) \), implicitly defined by Algorithm 2. By Lemma 10 it holds that \( \hat{R}(0) > 0 \). Furthermore, \( \delta \) enters linearly in the equations of Algorithm 2 and each step of the algorithm is continuous. This includes line 5, where the inverse of \( \hat{R}_k \) appears, which is well-defined and continuous as long as \( \hat{R}_k(\delta) \) remains positive definite, which is true at \( \delta = 0 \). As a consequence, the map \( \hat{R}(\delta) \) is continuous at the origin. It follows that there exists a value \( \delta > 0 \) such that \( \hat{R}_k > 0 \), for \( k = 0, \ldots, N-1 \).

We conclude this section by establishing our main result.

**Theorem 12.** Consider QP (2a)-(2c), Definition 4 and assume LICQ holds. Then \( Z^\top HZ > 0 \iff 3\delta > 0 \), such that \( \bar{H}(\delta) > 0 \) as defined by Algorithm 2.

**Proof.** Assume there exists some \( \delta > 0 \) such that \( \bar{H} > 0 \). Then \( Z^\top HZ > 0 \) follows from \( \bar{H} = H + U \) and Lemma 9. The converse is proven as follows. From Algorithm 2 we have that \( \bar{Q}_k - \bar{S}_k^\top \bar{R}_k^{-1} \bar{S}_k = 3I > 0 \). By Lemma 11, \( 3\delta > 0 \) such that \( \hat{R}_k > 0 \). Then, from the Schur complement lemma, it follows that

\[
(26) \quad \bar{H}_k := \begin{bmatrix} \bar{Q}_k & \bar{S}_k \\ \bar{S}_k & \bar{R}_k \end{bmatrix} > 0.
\]

As \( \bar{Q}_N = 3I > 0 \), it follows that \( \bar{H} = \text{diag}(\bar{R}_0, \ldots, \bar{R}_{N-1}, \bar{Q}_N) > 0 \).

### 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 13 (DRE).** 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 \( X_N := \bar{Q}_N \) iterates backwards from \( k = N-1 \) to \( k = 0 \) by computing

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

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

**Lemma 14.** Consider QP (2a)-(2c), Algorithm 2 and assume \( Z^\top HZ > 0 \), with \( Z \) as in Definition 4. If \( \delta = 0 \), then the matrices \( \bar{Q}_0, \ldots, \bar{Q}_N \) in the output of Algorithm 2 are equal to the cost-to-go matrices \( X_0, \ldots, X_N \) computed with the DRE as defined above.

**Proof.** For stage \( N \), \( X_N = \bar{Q}_N \) by definition. For \( k = 0, \ldots, N-1 \), from Definition 13, we have that

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

and, if we replace \( X_{k+1} \) by \( \bar{Q}_{k+1} \),

\[
(29) \quad X_k = \hat{Q}_k - \hat{S}_k^\top \hat{R}_k^{-1} \hat{S}_k, \\
(30) \quad = \bar{Q}_k,
\]
where we used $\hat{Q}_k, \hat{S}_k, \hat{R}_k$ as in (20) for ease of notation, and (30) follows from Algorithm 2, where $\bar{Q}_k = Q_k - \hat{Q}_k$, 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:

$$L(w, \lambda) = \frac{1}{2} \sum_{k=0}^{N-1} x_k^T Q_k x_k + 2x_k^T S_k^T u_k + u_k^T R_k u_k + \frac{1}{2} x_N^T \hat{Q}_N x_N$$  

$$+ \sum_{k=0}^{N-1} \lambda_{k+1} (A_k x_k + B_k u_k - x_{k+1}) + \lambda_0^T (x_0 - x_0),$$

(31)

with $\lambda^T = [\lambda_0^T, \ldots, \lambda_N^T], \lambda_k \in \mathbb{R}^{nx}$. 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 [22]. The derivation is shown below, a procedure to compute $\lambda$ is stated in Algorithm 3.

$$0 = \frac{\partial L(w, \lambda)}{\partial x_k}^\top, \quad k = 0, \ldots, N - 1$$  

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

(32)

where additionally

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

(33)

(34)

Algorithm 3 Recovery of Lagrange multipliers: equality constrained case

Input: $w$

Output: $\lambda$

1: $\lambda_N = \hat{Q}_N x_N$
2: for $k = N - 1, \ldots, 0$ do
3: $\lambda_k \leftarrow Q_k x_k + S_k^T u_k + A_k^T \lambda_{k+1}$
4: end for

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

$$\begin{align*}
\text{minimize} & \quad x_0^2 - \frac{1}{2} u_0^2 + x_1^2 + x_1^4 \\
\text{subject to} & \quad x_1 = x_0 + u_0 \\
& \quad x_0 = \bar{x}_0.
\end{align*}$$

As the term $x_1^4$ makes this problem an NLP, we solve it by using an SQP method with exact Hessian. We set $x_0 = 1$.

We define $w := [x_0, u_0, x_1]^\top$. Optimization problem (35) has a global minimizer at $w^* = [1, -3/2, -1/2]^\top$. The Hessian of the Lagrangian at the solution is equal to

$$\nabla_w^2 L(w^*, \lambda^*) = \nabla^2 f(w^*) = \text{diag}(2, -1, 5) \neq 0.$$  

(36)
Fig. 1: Convergence for SQP algorithm with three different regularization methods. On the vertical axis we plot the distance to the global solution $w^*$. 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

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

such that $GZ = 0$ and $Z^TZ = I$. The reduced Hessian at the solution $w^*$ then reads as $Z^T\nabla^2_{ww} \mathcal{L}(w^*, \lambda^*)Z = 2 > 0$.

Applying our convexification with sufficiently small $\delta$ 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 convexification with Newton’s method using 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 $V_kD_kV_k^{-1}$ the eigenvalue decomposition of the Hessian block $H_k$, $k = 0, \ldots, N$, these two regularizations are defined as follows:

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

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

where $\text{abs}(\cdot)$ is the operator which takes the element-wise absolute value and $\epsilon > 0$.

In Figure 1, we compare convergence of the SQP method for the convexification 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 of no 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):

\[(39) \quad \text{minimize} \quad f(y) \quad \text{subject to} \quad g(y) = 0, \quad c(y) \leq 0,\]

with equality constraints \(g : \mathbb{R}^n \rightarrow \mathbb{R}^{n_e}\) and inequality constraints \(c : \mathbb{R}^n \rightarrow \mathbb{R}^{n_c}\) with index set \(I = \{1, 2, \ldots, n_c\}\). We define the Lagrangian of NLP (39) as

\[(40) \quad L(y, \lambda, \mu) = f(y) + \lambda^T g(y) + \mu^T c(y),\]

with Lagrange multipliers \(\lambda \in \mathbb{R}^{n_e}, \mu \in \mathbb{R}^{n_c}\). A KKT-point \(z^* := (y^*, \lambda^*, \mu^*)\) 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:

\[(41) \quad A(y) := \{i \in I \mid c_i(y) = 0\}.\]

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

\[(42a) \quad \text{minimize} \quad \frac{1}{2} w^T \nabla^2 f(y) w + \nabla f(y)^T w\]

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

\[(42c) \quad 0 \geq c(y) + \frac{\partial c}{\partial y}(y)w.\]

We repeat the following lemma from [22].

**Lemma 15.** Suppose that \(z^*\) is a regular solution of (39). Then if \((y, \lambda, \mu)\) is sufficiently close to \(z^*\), there is a regular solution of (42) whose active set is the same as the active set \(A(y^*)\) of NLP (39) at \(z^*\).

Suppose we are at a point \((\bar{y}, \bar{\lambda}, \bar{\mu})\) sufficiently close to a solution of the NLP. Then Lemma 15 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 shorthands \(G := \frac{\partial f}{\partial y}(\bar{y}), G_{\text{act}} := \frac{\partial c}{\partial y}(\bar{y})\), \(i \in A(\bar{y})\), and \(n_a\) denotes the number of active constraints. For ease of notation, we omit the dependence of \(G, G_{\text{act}}\) on \(\bar{y}\), as it is constant within one QP subproblem.

**Definition 16 (QP with fixed active set).**

\[(43a) \quad \text{minimize} \quad \frac{1}{2} w^T H w\]

\[(43b) \quad \text{subject to} \quad \tilde{G} w + \tilde{g} = 0,\]

with \(\tilde{G} := \begin{bmatrix} G \\ G_{\text{act}} \end{bmatrix}, G \in \mathbb{R}^{p \times n}, G_{\text{act}} \in \mathbb{R}^{n_a \times n}\) and \(\tilde{g} := \begin{bmatrix} g \\ g_{\text{act}} \end{bmatrix}, g \in \mathbb{R}^p, g_{\text{act}} \in \mathbb{R}^{n_a}\). 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 [14]. Additionally, we introduce the following matrices:
Consider $\tilde{G}$ as in QP (43). We define $\tilde{Z} \in \mathbb{R}^{n \times (n-p-n_a)}$, such that the following hold:

$$\tilde{G}\tilde{Z} = 0, \quad \tilde{Z}^\top \tilde{Z} = I.$$  

Matrix $\tilde{Z}$ is a basis for the null space of $\tilde{G}$. The null space of $G$ comprises $\tilde{Z}$ but is possibly larger. We complement $\tilde{Z}$ 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:

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

From now on we call $\tilde{Z}^\top H\tilde{Z}$ the reduced Hessian. Furthermore, note that the above definition of $Z$ is compatible with Definition 4.

Using Definition 17, we establish an extension of Theorem 7 in the case of active inequality constraints in Theorem 19, after proving the following lemma.

**Lemma 18.** Consider QP (43), Definition 17 and assume LICQ holds. We then have the following equivalence:

$$\tilde{Z}^\top H\tilde{Z} > 0 \iff \exists \Gamma \in \mathbb{S}^{n_a} : Z^\top (H + G_{act}^\top \Gamma G_{act}) Z > 0.$$  

**Proof.** The proof follows a similar argument as the proof of Theorem 7. We consider $H + G_{act}^\top \Gamma G_{act}$ in the basis $Z = (Z_c|\tilde{Z})$, see (45). Applying the Schur complement lemma yields the following conditions:

$$\tilde{Z}^\top H\tilde{Z} > 0,$$

$$Z_c^\top (H + G_{act}^\top \Gamma G_{act}) Z_c > Z_c^\top H\tilde{Z} \cdot (\tilde{Z}^\top H\tilde{Z})^{-1} \cdot \tilde{Z}^\top Hc,$$

where we used the fact that $G_{act}\tilde{Z} = 0$. 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 $\tilde{G}$ and part of the null space of $G$, it holds that $G_{act}Z_c$ is of full rank and therefore invertible. Thus, (48) becomes

$$\Gamma > (Z_c^\top G_{act}^\top)^{-1}(Z_c^\top H\tilde{Z} \cdot (\tilde{Z}^\top H\tilde{Z})^{-1} \cdot \tilde{Z}^\top HZ_c - Z_c^\top HZc)(G_{act}Z_c)^{-1}.$$  

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

**Theorem 19.** Consider QP (43), Definition 17 and assume LICQ holds. It then holds that

$$\tilde{Z}^\top H\tilde{Z} > 0 \iff \exists \Gamma \in \mathbb{S}^{n_a}, \exists U \in \mathbb{S}^n : Z^\top UZ = 0, H + G_{act}^\top \Gamma G_{act} + U > 0.$$  

**Proof.** The proof of the theorem follows from Theorem 7 and Lemma 18.

**4.2. Preserving the OCP structure.** Theorem 19 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 stage-wise active set as follows:

$$A_k(w) := \{i \in I_k \mid \text{row}_i(c_k(w)) = 0\},$$  

with $I_k$ the index set corresponding to the inequalities in each stage $k = 0, \ldots, N$, respectively. We can now define $G_{act,k}$ at some feasible point $\tilde{w}$, as follows:

$$G_{act,k} := \left( \frac{\partial \text{row}_i(c_k)}{\partial w} \right)_{\tilde{w}} : i \in A_k(\tilde{w}).$$
for $k = 0, \ldots, N$. Again, we omit $\mathbf{w}$ in the notation for $G_{\text{act}, k}$ for improved readability of the equations. Furthermore, we define $(G^{\text{OCP}}_{\text{act}})^\top := (G^T_{\text{act}, 0} | \cdots | G^T_{\text{act}, N})$. We remark that $(G^{\text{OCP}}_{\text{act}})^\top G^{\text{OCP}}_{\text{act}} \in \mathbb{S}^{N, n_x, n_u}_\text{OCP}$. Using these definitions, we can establish the following theorem:

**Theorem 20.** Consider QP (2), Definition 17 and assume that LICQ holds. Then, it holds that

\[
\tilde{Z}^\top H \tilde{Z} > 0
\]

\[
\iff \exists \gamma \in \mathbb{R}, \exists U \in \mathbb{S}^{N, n_x, n_u}_\text{OCP} : Z^\top U Z = 0, H + \gamma (G^{\text{OCP}}_{\text{act}})^\top G^{\text{OCP}}_{\text{act}} + U > 0.
\]

**Proof.** The proof follows from Theorem 19 with matrix $\Gamma = \gamma I$ and Theorem 12. Note that the sparsity structure of $H$ is preserved in $\tilde{H} := H + \gamma (G^{\text{OCP}}_{\text{act}})^\top G^{\text{OCP}}_{\text{act}} + U$, as $U \in \mathbb{S}^{N, n_x, n_u}_\text{OCP}$ and $(G^{\text{OCP}}_{\text{act}})^\top G^{\text{OCP}}_{\text{act}} \in \mathbb{S}^{N, n_x, n_u}_\text{OCP}$.

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 $\gamma G^T_{\text{act}, k} G_{\text{act}, k}$ to the original Hessian blocks.

**Algorithm 4** Structure-Preserving Convexification: inequality constrained case

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

**Output:** $\tilde{H}$

1: $\tilde{Q}_N = \delta I$
2: $\tilde{Q}_N = \hat{Q}_N + \gamma G^T_{\text{act}, N} G_{\text{act}, N} - \hat{Q}_N$
3: for $k = N - 1, \ldots, 0$ do
4: \[
\begin{bmatrix}
\tilde{Q}_k \\
\tilde{S}_k \\
\tilde{R}_k
\end{bmatrix}
= \begin{bmatrix}
Q_k \\
S_k \\
R_k
\end{bmatrix}
+ \begin{bmatrix}
A_k^T \hat{Q}_{k+1} A_k \\
B_k^T \hat{Q}_{k+1} A_k \\
B_k^T \hat{Q}_{k+1} B_k
\end{bmatrix}
+ \gamma G^T_{\text{act}, k} G_{\text{act}, k}
\]
5: $\tilde{Q}_k := \tilde{S}_k^T \tilde{R}_k^{-1} \tilde{S}_k + \delta I$
6: $\tilde{H}_k := \begin{bmatrix}
\tilde{Q}_k \\
\tilde{S}_k \\
\tilde{R}_k
\end{bmatrix}$
7: $\tilde{Q}_k = \hat{Q}_k - \tilde{Q}_k$
8: end for
9: $\tilde{H} := \text{diag}(\tilde{H}_0, \ldots, \tilde{Q}_N)$

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

**Theorem 21.** Consider QP (2), Definition 16, Definition 17, assuming LICQ and $\tilde{Z}^\top H \tilde{Z} > 0$ hold. Then, the primal solutions of QP($H$) and QP($\tilde{H}$) are identical, with $\tilde{H}$ defined as in Algorithm 4.

**Proof.** The proof is based on the null space method for solving equality constrained QPs, as presented in [22]. We decompose the primal solution vector $w$ as follows:

\[
w = \tilde{Z} w_z + \tilde{Y} w_y,
\]

with $\tilde{Z}$ as in Definition 17, and we complement the basis of the null space of $\tilde{G}$ with a basis of its range space $\tilde{Y}$, such that $(\tilde{Z} \tilde{Y})^\top (\tilde{Z} \tilde{Y}) = I$. We can compute $w_y$ from
the constraints:

\begin{align}
G\tilde{Y}w_y &= -g, \\
G_{\text{act}}^{\text{OCP}}\tilde{Y}w_y &= -g_{\text{act}}^{\text{OCP}}.
\end{align}

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

\begin{equation}
H\tilde{Z}w_z + HY\tilde{y}w_y + G^\top \lambda + (G_{\text{act}}^{\text{OCP}})^\top \mu = 0.
\end{equation}

Multiplying from the left with \( \tilde{Z}^\top \) gives

\begin{equation}
\tilde{Z}^\top H\tilde{Z}w_z = -\tilde{Z}^\top HY\tilde{y}w_y,
\end{equation}

where we used the fact that \( \tilde{Z} \) forms a basis for the null space of \( G_z \). Since \( \tilde{Z}^\top H\tilde{Z} > 0 \) by assumption, the solution is well-defined. Substituting \( H \) by \( \tilde{Z}^\top G\tilde{Z} \) gives

\begin{equation}
\tilde{Z}^\top U\tilde{Y}w_y = \tilde{Z}^\top G^\top (G + \Sigma)\tilde{Y}w_y + \tilde{Z}^\top \Sigma^\top QG\tilde{Y}w_y,
\end{equation}

and we can show that \( \tilde{Z}^\top \Sigma^\top QG\tilde{Y}w_y = 0 \), as follows:

\begin{equation}
\tilde{Z}^\top \Sigma^\top QG\tilde{Y}w_y = \tilde{Z}^\top \Sigma^\top Q(-g), \quad \text{from (54)}
\end{equation}

where the last step follows from the fact that only the first \( n_z \) rows of \( g \) are non-zero (see (13)), and the first \( n_x \) columns of \( Z_c^\top \Sigma^\top Q \) are zero (see (18) and (45)). Thus, (10) and (57) 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 in 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 NLP, see Lemma 15. Supposing we have identified the correct active set, we can write the QP as in (43). The corresponding Lagrangian function and its gradient are

\begin{align}
\mathcal{L}(w, \lambda, \mu) &:= \frac{1}{2} w^\top Hw + \lambda^\top (Gw + g) + \mu_{\text{act}}^\top (G_{\text{act}}^{\text{OCP}} w + g_{\text{act}}), \\
\nabla_w \mathcal{L}(w, \lambda, \mu) &= Hw + G^\top \lambda + (G_{\text{act}}^{\text{OCP}})^\top \mu_{\text{act}},
\end{align}

where \( g_{\text{act}} := c_i(w), i \in A(w) \) and \( \mu_{\text{act}} \) are the multipliers corresponding to the active inequalities.
Multipliers of active inequality constraints. Using the definitions of $\tilde{G}, \tilde{Z}, Z_c$ as in (45), and stating $\nabla_w \mathcal{L}(w^*, \lambda^*, \mu^*) = 0$, we can write

\[(61) \quad (G_{\text{act}}^\text{OCP} Z_c) \transp \mu^\text{act} = -Z_c \transp (H w^*),\]

where we multiplied $\nabla_w \mathcal{L}(w^*, \lambda^*, \mu^*)$ from the left with $Z_c$. Note that the matrix $G_{\text{act}}^\text{OCP} Z_c$ is invertible, for the same reasons as given in the proof of Lemma 18. Substituting the original Hessian $H$ by the convexified Hessian $\gamma G_{\text{act}}^\text{OCP}$ gives us an expression for the multipliers corresponding to the active inequalities of the convexified problem:

\[(62) \quad (G_{\text{act}}^\text{OCP} Z_c) \transp \mu^\text{conv, act} = -Z_c \transp (H w^* + U w^* + \gamma (G_{\text{act}}^\text{OCP}) \transp G_{\text{act}}^\text{OCP} w^*).\]

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

\[Z_c \transp U w^* = Z_c \transp (G + \Sigma) w^* + Z_c \transp \Sigma \transp \bar{Q} G w^* = Z_c \transp \Sigma \transp \bar{Q} (-g), = 0,\]

where we used a similar argument as in the proof of Theorem 21.

Comparing (61) and (62), we can recover the correct multipliers of the original problem as

\[(63) \quad \mu^\text{act} = \mu^\text{conv, act} + \gamma G_{\text{act}}^\text{OCP} w^*.\]

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

**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.

**Algorithm 5** Recovery of Lagrange multipliers: inequality constrained case

**Input:** $w$, $\mu^\text{conv, act}$

**Output:** $\lambda$, $\mu^\text{act}$

1. $\mu^\text{act, N} \leftarrow \mu^\text{conv, act, N} + \gamma G_{\text{act, N}} x_N$
2. $\lambda_N \leftarrow \bar{Q}_N x_N + C_N \mu^\text{act, N}$
3. for $k = N - 1, \ldots, 0$ do
4. $\mu^\text{act, k} \leftarrow \mu^\text{conv, act, k} + \gamma G_{\text{act, k}} w_k$
5. $\lambda_k \leftarrow \bar{Q}_k x_k + S_k \transp u_k + A_k \transp \mu^\text{act, N} + C_k, x \mu^\text{act, k}$
6. end for

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 [22].

**Theorem 22.** Regard NLP (1) with a regular solution \( z^* = (w^*, \lambda^*, \mu^*) \). Then there exist \( \delta > 0 \), \( \gamma > 0 \), and \( \epsilon > 0 \) so that for all \( z_0 = (w_0, \lambda_0, \mu_0) \) with \( \|z_0 - z^*\| < \epsilon \), a full step SQP algorithm with Hessian matrices convexified by using Algorithm 4 converges, the QP subproblems are convex, have the same active set as NLP (1), and the convergence rate is asymptotically quadratic.

**Proof.** The asymptotically quadratic convergence rate follows from a standard convergence proof of Newton’s method, e.g. Theorem 3.5 in [22]. If the Hessian is exact and not convexified, the optimal active set is identified at \( z_0 \). From Theorem 2.1 in [26], we have the existence of an open neighborhood \( \mathcal{N} \) around the regular solution \( z^* \) such that all \( z \in \mathcal{N} \) have the same active set as the optimal active set \( A(z^*) \). As such, there always exists an \( \epsilon \) such that \( z_0 \in \mathcal{N} \). A similar statement can be found in [4].

In Theorem 21, we state that by using our convexification procedure the primal step is not altered. Moreover, by using Algorithm 5 we also recover the correct dual step. Finally, convexity of the QP subproblems follows from Theorem 20. Therefore, by using our convexification approach we take the same primal-dual steps as the original problem while solving convex QP subproblems. \( \square \)

5. **Dealing with indefinite reduced problems.** In the previous sections, we assumed a positive reduced Hessian. We call problems with an indefinite reduced Hessian \( \tilde{Z}^T H \tilde{Z} \prec 0 \) 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 \( \tilde{R}_k \succ 0 \) holds in line 5 of the algorithm, such that the Schur complement lemma no longer holds. Instead, we still compute \( \tilde{H}_k \) 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 \( \tilde{R}_k \).

**Remark 23.** In Theorem 20, 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 identical and in principle, we can choose \( \gamma \) 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 \( \gamma \) might result in poor convergence. A possible heuristic as an alternative to a fixed value \( \gamma \) is motivated by referring to Algorithm 6. In line 5 of the algorithm, we check for positive definiteness of \( \tilde{R}_k \). In line 4, we will try to make this matrix positive definite, by adding a matrix \( G_{\text{act},k}^\top \Gamma G_{\text{act},k} \), where we compute \( \Gamma \) as follows. Consider a decomposition of matrix \( \tilde{R}_k + \tilde{B}_k^\top \tilde{Q}_{k+1} B_k \) in a basis for the range space \( Y_{\text{act},k} \) and null space \( Z_{\text{act},k} \) of \( G_{\text{act},k} \). A necessary condition for the positive definiteness of \( \tilde{R}_k \),
for some \( \gamma \) (65) \( \Gamma \) such that we could propose the following expression for \( \text{Hessian} \):

\[
H = H + \Delta H = H + \text{diag}(\Delta H_0, \ldots, \Delta H_{N-1}, 0),
\]

by the Schur complement lemma, is then

\[
Y_{\text{act}, k}^T (R_k + B_k^T Q_{k+1} B_k + G_{\text{act}, k}^T \gamma G_{\text{act}, k}) Y_{\text{act}, k} > 0,
\]

such that we could propose the following expression for \( \Gamma \):

\[
\Gamma = -(Y_{\text{act}, k}^T G_{\text{act}, k})^{-1} Y_{\text{act}, k}^T (R_k + B_k^T Q_{k+1} B_k) Y_{\text{act}, k} (G_{\text{act}, k} Y_{\text{act}, k})^{-1} + \gamma I,
\]

for some \( \gamma > 0 \), where we used the fact that \( G_{\text{act}, k} Y_{\text{act}, k} \) is invertible by construction. Note that (65) is a heuristic choice for \( \Gamma \) in the sense that the above condition is only necessary, i.e. you still might have to apply regularization on \( \tilde{H}_k \), 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 \( \text{QP}(\tilde{H}) \) 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 backwards 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, as it employs the original Hessian matrix. 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:

\[
\text{Algorithm 6 Structure-Preserving Convexification for inequality constrained optimization, including the regularization of Hessian blocks}
\]

**Input:** \( H, \delta, \gamma, \epsilon, \) current active set \( \mathcal{A} \)

**Output:** \( \tilde{H} \)

1: \( \tilde{Q}_N = \delta I \)
2: \( \tilde{Q}_N = \tilde{Q}_N + \gamma G_{\text{act}, N}^T G_{\text{act}, N} - \tilde{Q}_N \)
3: for \( k = N-1, \ldots, 0 \) do
4: \[
\begin{bmatrix}
\tilde{Q}_k \\
\tilde{S}_k \\
\tilde{R}_k
\end{bmatrix}
= \begin{bmatrix}
Q_k \\
S_k \\
R_k
\end{bmatrix}
+ \begin{bmatrix}
A_k^T Q_{k+1} A_k \\
B_k^T Q_{k+1} A_k \\
B_k^T Q_{k+1} B_k
\end{bmatrix}
+ \gamma G_{\text{act}, k}^T G_{\text{act}, k}^T
\]
5: if \( \tilde{R}_k \neq 0 \) then
6: \( \tilde{H}_k := \begin{bmatrix}
\tilde{Q}_k \\
\tilde{S}_k \\
\tilde{R}_k
\end{bmatrix}
= \text{project}(\tilde{H}_k, \epsilon) \)
7: else
8: \( \tilde{H}_k := \begin{bmatrix}
\begin{bmatrix}
\tilde{Q}_k \\
\tilde{S}_k \\
\tilde{R}_k
\end{bmatrix}
\end{bmatrix}
= \begin{bmatrix}
\tilde{Q}_k \\
\tilde{S}_k \\
\tilde{R}_k
\end{bmatrix}
\]
9: end if
10: \( \tilde{Q}_k := S_k R_k^{-1} \tilde{S}_k + \delta I \)
11: \( \tilde{H}_k := \begin{bmatrix}
\tilde{Q}_k \\
\tilde{S}_k \\
\tilde{R}_k
\end{bmatrix}
\)
12: \( \tilde{Q}_k = \tilde{Q}_k - \tilde{Q}_k \)
13: end for
14: \( \tilde{H} := \text{diag}(\tilde{H}_0, \ldots, \tilde{Q}_N) \)
where $\Delta H_k = 0$ when there was no regularization and $\Delta H_k = \tilde{H}_k - \hat{H}_k$ otherwise. We then apply Algorithm 5 to $H_{\text{mod}}$ 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 optimal control problem on an inverted pendulum. This system, depicted in Figure 2, consists of a rod of length $l$ making an angle $\theta$ 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:

\[
\begin{align*}
\dot{p} &= v, \\
\dot{\theta} &= \omega, \\
\dot{v} &= \frac{-ml \sin(\theta) \dot{\theta}^2 + mg \cos(\theta) \sin(\theta) + F}{M + m - m \cos(\theta)^2}, \\
\dot{\omega} &= \frac{-ml \cos(\theta) \sin(\theta) \dot{\theta}^2 + F \cos(\theta) + (M + m)g \sin(\theta)}{l(M + m - m \cos(\theta)^2)}.
\end{align*}
\]

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:

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=0}^{N-1} \begin{bmatrix} s_k \\ F_k \end{bmatrix}^\top \begin{bmatrix} Q & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} s_k \\ F_k \end{bmatrix} + s_N^\top Q s_N \\
\text{subject to} & \quad s_{k+1} = \phi_k(s_k, F_k), \quad k = 0, \ldots, N-1, \\
& \quad -80 \leq F_k \leq 80, \quad k = 0, \ldots, N-1, \\
& \quad s_0 = s_0,
\end{align*}
\]

where $\phi$ denotes a numerical integration method (explicit Runge-Kutta method of order 4) to simulate the continuous-time dynamics in (67) over one shooting interval, the weight matrices are chosen as $Q = \text{diag}([1000, 1000, 0.01, 0.01])$, $R = 0.01$ and the initial value is $s_0 = [0, \pi, 0, 0]^\top$. We choose $N = 100$ shooting intervals of length 0.01 s.
We solve NLP (68) with a full-step SQP method. 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 $\tilde{Z}^\top H \tilde{Z}$ is positive definite, but $Z^\top H Z \not\succ 0$, we do not need to regularize thanks to the terms $\gamma G_{\text{act},k}'G_{\text{act},k}$ 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 iteration 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, 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 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 22, when using the convexification method once the active set is fixed (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 very similar to the one reported in Table 1 and we obtain similar convergence profiles to

Table 1: Exact-Hessian SQP iterations for the pendulum example, using the structure-preserving convexification algorithm from Algorithm 6.

<table>
<thead>
<tr>
<th>it.</th>
<th>KKT norm</th>
<th>step size</th>
<th>$H &gt; 0$</th>
<th>$\tilde{Z}^\top H \tilde{Z} &gt; 0$</th>
<th>$Z^\top H Z &gt; 0$</th>
<th>regs.</th>
<th>act. set chgs.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.52e+02</td>
<td>7.39e+02</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>0</td>
<td>74</td>
</tr>
<tr>
<td>2</td>
<td>5.33e+06</td>
<td>9.63e+02</td>
<td></td>
<td></td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2.02e+06</td>
<td>6.01e+02</td>
<td></td>
<td></td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.47e+06</td>
<td>3.27e+02</td>
<td></td>
<td></td>
<td>5</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>9.44e+05</td>
<td>3.22e+02</td>
<td></td>
<td></td>
<td>4</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3.75e+05</td>
<td>4.87e+02</td>
<td></td>
<td></td>
<td>3</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1.04e+05</td>
<td>4.93e+02</td>
<td></td>
<td></td>
<td>1</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9.13e+03</td>
<td>3.23e+02</td>
<td></td>
<td></td>
<td>2</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3.90e+02</td>
<td>4.14e+02</td>
<td>True</td>
<td></td>
<td>0</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1.25e+02</td>
<td>9.53e+01</td>
<td>True</td>
<td></td>
<td>0</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>6.66e+00</td>
<td>3.21e+00</td>
<td>True</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>1.38e-03</td>
<td>2.88e-03</td>
<td>True</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>2.02e-08</td>
<td>5.33e-09</td>
<td>True</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>1.42e-10</td>
<td>6.99e-11</td>
<td>True</td>
<td></td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3 for all methods.

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 [17] or in [31]. 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 [17] 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 optimal control problems using SQP. We prove 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 offer 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 propose a regularization method based on the convexification. We illustrate 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 only work with positive definite block diagonal Hessians. Finally, the automatic selection of parameters $\delta, \gamma, \epsilon$ will be investigated in future research.
REFERENCES


24


