A projected gradient and constraint linearization method for nonlinear model predictive control

Torrisi, Giampaolo; Grammatico, Sergio; Smith, Roy S.; Morari, Manfred

DOI
10.1137/16M1098103

Publication date
2018

Document Version
Final published version

Published in
SIAM Journal on Control and Optimization

SIAM J. CONTROL OPTIM.
Vol. 56, No. 3, pp. 1968–1999

A PROJECTED GRADIENT AND CONSTRAINT LINEARIZATION METHOD FOR NONLINEAR MODEL PREDICTIVE CONTROL*

GIAMPAOLO TORRISI†, SERGIO GRAMMATICO‡, ROY S. SMITH†, AND MANFRED MORARI§

Abstract. Projected gradient descent denotes a class of iterative methods for solving optimization programs. In convex optimization, its computational complexity is relatively low whenever the projection onto the feasible set is relatively easy to compute. On the other hand, when the problem is nonconvex, e.g., because of nonlinear equality constraints, the projection becomes hard and thus impractical. In this paper, we propose a projected gradient method for nonlinear programs that only requires projections onto the linearization of the nonlinear constraints around the current iterate, similar to sequential quadratic programming (SQP). The proposed method falls neither into the class of projected gradient descent approaches, because the projection is not performed onto the original nonlinear manifold, nor into that of SQP, since second-order information is not used. For nonlinear smooth optimization problems, we assess local and global convergence to a Karush–Kuhn–Tucker point of the original problem. Further, we show that nonlinear model predictive control is a promising application of the proposed method, due to the sparsity of the resulting optimization problem.

Key words. nonlinear programming, first-order methods, sequential quadratic programming, nonlinear model predictive control

AMS subject classifications. 90C30, 90C55

DOI. 10.1137/16M1098103

1. Introduction.

The projected gradient method is an established approach for solving convex optimization problems. The subject has been extensively investigated over the last decades, developing algorithms that guarantee best performance for convex and strongly convex problems; see [32, 1]. Recently, the Nesterov's accelerated gradient method has been applied to linear model predictive control (MPC), and a priori worst-case bounds for finding a solution with prespecified accuracy have been derived [31, 40]. When the optimization problem is a general nonlinear program, the gradient method can still be used for finding a Karush–Kuhn–Tucker (KKT) point [42]. In particular, for general nonconvex constraints in the form of a nonlinear manifold, the projection onto the feasible set is performed in two stages. First, the projection is derived onto the tangent space to the nonlinear manifold, which in general is a polyhedron. Then, the determined point is projected again onto the original nonlinear manifold, via some strategy guaranteed to determine a feasible point that improves the objective function. While ensuring convergence, this second projection

*Received by the editors October 10, 2016; accepted for publication (in revised form) February 21, 2018; published electronically May 31, 2018. This work has similarities with *A variant to sequential quadratic programming for nonlinear model predictive control*, Proceedings of the IEEE Conference on Decision and Control, 2016, pp. 2814–2819. The technical differences are explained in section 1.
http://www.siam.org/journals/sicon/56-3/M109810.html
Funding: The work of the first author was partially supported by ABB Corporate Research, Switzerland.

†Automatic Control Laboratory, ETH Zurich, Zurich, Switzerland (torrisig@control.ee.ethz.ch, rsmith@control.ee.ethz.ch).

‡Delft Center for Systems and Control, TU Delft, Delft, The Netherlands (s.grammatico@tudelft.nl).

§Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 (morari@seas.upenn.edu).

is in general computationally expensive, hence the method is not recommended in practice for solving nonlinear MPC problems.

In this paper, we analyze a gradient method for nonlinear programs (NLPs) that only requires projections onto the tangent space, obtained by linearization of the nonlinear manifold around the current iterate. Note that this determines a sequence of points that are not necessarily feasible for the original NLP. Thus, standard projected gradient method results do not apply to prove convergence.

Linearized constraints are instead considered in sequential quadratic programming (SQP), which is an established method to determine a local solution to a smooth nonconvex NLP. The solution is determined via a sequence of iterates, each obtained as the solution to a quadratic program (QP), that are usually called the major iterations. In turn, each QP is solved via so-called minor iterations using available convex optimization methods [35, 41, 2]. Typically, each QP has as objective function a second-order approximation of the Lagrangian function of the nonlinear problem and as constraints the linearization of the nonlinear manifold, both computed at the current iterate. The solution of the QP updates the current iterate, and then the next QP is formulated.

When considering programs with only equality constraints, the basic SQP method is equivalent to the Newton's method applied to the KKT conditions of the original nonlinear optimization problem; thus it is locally quadratically convergent [43]. Since the required Hessian of the Lagrangian is expensive to compute and it is not guaranteed to be positive definite on every subspace far from the solution, a suitable approximation of the Hessian of the Lagrangian is typically used, e.g., quasi-Newton or Broyden–Fletcher–Goldfarb–Shanno (BFGS) updates, that guarantee local superlinear convergence [5, 3, 7, 33]. Global convergence is usually obtained via a line-search approach. Merit functions are considered that comprise both the objective and the constraint functions, e.g., in the form of an augmented Lagrangian [19, 22, 38]. Then by appropriate tuning of some penalty parameters, the solution to the QP is proven to be a descent direction for the merit function. A line search then determines a step size for the convergence of the method. Several contributions in the literature have discussed different reformulations that trade off theoretical convergence guarantees and computational complexity; see the review in section 2 [2, 38, 24, 37, 22]. Indeed, commercial numerical solvers use this technique for approaching a KKT point of an NLP [21]. An alternative approach for establishing global convergence is the trust-region method, where additional constraints are included in the optimization program [2, 12, 45].

The proposed gradient algorithm in this paper can be seen as an incomplete SQP where, instead of solving each generated QP, only one gradient step is computed for the QP and then projected onto the linearized constraint. Some literature has proposed solving the QP program inexactly, e.g., by bounding the suboptimality of the estimated solution to the QP to recover some rate of convergence for the SQP [10, 27, 30]. However, our approach does not fall into this class of methods, as only one gradient step for each QP is in general not enough for reaching the desired level of suboptimality. Moreover, for these approaches second-order information on the Lagrangian is necessary, which in contrast is not required for our proposed algorithm.

An approach that is widely considered in the literature is the real-time iteration (RTI), specifically oriented to MPC applications [13]. It basically yields an approximation to the NLP solution based on the SQP method. Instead of iterating the solution to the QPs until a KKT point is encountered, only one QP is solved. Further technical differences concern the computation of the Jacobian matrices—the QPs are

formed using the Jacobian matrices from the previous time step—and how the initial state is embedded into the optimization problem. These refinements allow the next control input to be rapidly calculated before the linearization is found for the upcoming time step. Suboptimality and closed-loop considerations of RTI are discussed in [14].

An approach based on a first-order update can be found in [26]. The update is determined based on a convex combination of all of the previously computed gradient steps. The approach also works for a nondifferentiable objective function, even though this needs to be convex and the constraint function affine, thus its linearization is always feasible. Via duality theory it is also possible to find an equivalence between our projected gradient and constraint linearization step and the update step in [26]. An alternative method is the successive linear programming, which iteratively solves linear programs rather than QPs to determine a critical point of a NLP [36]. For this method though there is not a general convergence theory and the theoretical guarantees are derived only in the case with only linear constraints.

In this paper we show local and global convergence properties of the proposed gradient algorithm, leveraging established SQP results in the literature. Local conditions are derived when each gradient step is directly employed to update the current iterate. As in standard gradient method, ensuring convergence of the algorithm requires that some conditions have to be set on the gradient step size, typically depending on second-order information of the considered problem—in our case, the Lipschitz constant of the Lagrangian function. Under a particular assumption on the Hessian, the algorithm converges with linear rate, as expected for first-order methods for NLPs [20]. This is guaranteed to work close to a local optimum only. For the practical use of the algorithm, global convergence is required instead. Since updating the current iterate with this gradient step might not guarantee convergence, a variable step size is considered. Analogously to SQP, a merit function in the form of an augmented Lagrangian function weights the optimality and unfeasibility of the iterates and is employed in the line search for determining the step size.

Finally, we notice that sparsity considerably reduces the computational complexity of the problem. In particular, we show that nonlinear MPC is particularly well suited for applying the proposed method, due to the structure of the constraints generated by causal model dynamics. Similarly to the gradient method for linear MPC, easy-to-project constraints can be efficiently included in the nonlinear MPC formulation [40]. Furthermore, in the presence of a quadratic terminal cost and constraints that ensure closed-loop stability [29, 6, 9], we show that the projection can be computed in closed form, making the proposed algorithm computationally efficient.

Preliminary results have been published in [49], where the global convergence of a similar algorithm employing only primal iterates is proven. In this work, instead, the combined use of primal and dual variable iterates and other technical improvements in the considered augmented Lagrangian function reduce considerably the resulting computational complexity. In [49] the effect of some heuristics is also analyzed, that yield an interesting speed-up in the computational time specifically for MPC problems.

The remainder of this paper is organized as follows. Given the similarity of the proposed method to SQP, the standard SQP method is reviewed in section 2. Then, the proposed algorithm is presented in section 3. In section 4 we show the convergence of the algorithm, and then in section 5 we discuss the practical implementation for general problems and nonlinear MPC. Section 6 shows numerical experiments on a benchmark example.

2. Iterative methods for nonlinear optimization problems. We consider the constrained nonlinear optimization problem (NLP)

$$\begin{aligned} \min_{z\in\mathbb{R}^n} \quad & J(z) \\ \text{s.t.} \quad & g(z) \leq 0, \\ & h(z) = 0, \end{aligned} \tag{2.1}$$

where the functions $J : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$, and $h : \mathbb{R}^n \to \mathbb{R}^p$ are twice continuously differentiable functions, possibly nonconvex.

Let us define the Lagrangian function of the NLP in (2.1) as

$$\mathcal{L}(z, \lambda, \nu) := J(z) + g(z)^\top \lambda + h(z)^\top \nu,$$

with Lagrange multiplier vector $\lambda \in \mathbb{R}^m_{\geq 0}$ and $\nu \in \mathbb{R}^p$.

We call $z^\star$ a critical point of (2.1) if it satisfies the first-order conditions with strict complementarity [4], i.e., there exist $\lambda^\star \in \mathbb{R}^m_{\geq 0}$ and $\nu^\star \in \mathbb{R}^p$ such that

$$\begin{aligned} & \nabla\mathcal{L}(z^\star, \lambda^\star, \nu^\star) = \nabla J(z^\star) + \nabla g(z^\star)\lambda^\star + \nabla h(z^\star)\nu^\star = 0, \\ & \operatorname{diag}(\lambda^\star) g(z^\star) = 0, \\ & \lambda^\star_j > 0 \text{ if } g_j(z^\star) = 0, \\ & h(z^\star) = 0. \end{aligned} \tag{2.2}$$

Let us assume that the NLP in (2.1) has a finite number of critical points.

Iterative methods generate a sequence $\left(z^{(i)}\right)_{i=1}^{\infty}$ (major iterations) to determine a critical point $z^\star$. In subsection 2.1, the state-of-the-art SQP is reviewed [22]. To generate each $z^{(i)}$, a sequence of minor iterations is required, and each involves solving a QP. In section 3, we propose an alternative method to determine a critical point $z^\star$ by computing major iterations only. At the ith major iteration, $z^{(i+1)}$ is directly derived via a projected gradient step onto a linearization of the constraint around the current iterate $z^{(i)}$.

2.1. Sequential quadratic programming. The SQP method updates the sequence $\left(z^{(i)}\right)_{i\in\mathbb{N}}$ via the solution of a sequence of QPs. Given the current $z^{(i)}$, the method generates the QP

$$\begin{aligned} d_{\mathrm{z}}^{(i)} := \arg\min_{d_{\mathrm{z}}} \quad & \tfrac{1}{2} {d_{\mathrm{z}}}^\top H^{(i)} d_{\mathrm{z}} + \nabla J(z^{(i)})^\top d_{\mathrm{z}} \\ \text{s.t.} \quad & g(z^{(i)}) + \nabla g(z^{(i)})^\top d_{\mathrm{z}} \leq 0, \\ & h(z^{(i)}) + \nabla h(z^{(i)})^\top d_{\mathrm{z}} = 0, \end{aligned} \tag{2.3}$$

where $H^{(i)}$ is either the exact Hessian of the Lagrangian $\mathcal{L}$ of the NLP in (2.1) or an appropriate approximation. The dual variables $\lambda^{(i)}_{\mathrm{QP}} \in \mathbb{R}^m_{\geq 0}$ and $\nu^{(i)}_{\mathrm{QP}} \in \mathbb{R}^p$ are associated with the inequality and equality constraints, respectively. The resulting KKT conditions for the QP in (2.3) are

$$\begin{aligned} & H^{(i)} d_{\mathrm{z}}^{(i)} + \nabla J(z^{(i)}) + \nabla g(z^{(i)})\ \lambda^{(i)}_{\mathrm{QP}} + \nabla h(z^{(i)})\ \nu^{(i)}_{\mathrm{QP}} = 0, \\ & \operatorname{diag}(\lambda^{(i)}_{\mathrm{QP}}) \left(g(z^{(i)}) + \nabla g(z^{(i)})^\top d_{\mathrm{z}}^{(i)} \right) = 0, \\ & h(z^{(i)}) + \nabla h(z^{(i)})^\top d_{\mathrm{z}}^{(i)} = 0. \end{aligned} \tag{2.4}$$

Based on the solution $d_{\mathrm{z}}^{(i)}$ to (2.3), the sequence is updated as

$$z^{(i+1)} := z^{(i)} + t^{(i)} d_{\mathrm{z}}^{(i)}, \tag{2.5}$$

where $t^{(i)} \in (0, 1]$ is a step size to be determined.

To prove convergence results for the SQP methods, some regularity and boundedness assumptions are typically considered [22, Assumptions (i)–(iii)]. Some of these assumptions will additionally hold throughout the paper, and they will be denoted as standing assumptions.

Consider a generic optimization problem, indicated in the form of (2.1), a feasible solution z, and the set of the active inequality constraints as

$$\mathcal{I}_{\mathrm{NL}}(z) := \{\, j \in \{1, \dots, m\} \mid g_j(z) = 0 \,\}.$$

Then, the vector z is said to be regular if the equality constraint gradients $\nabla h_i(z)$, for all $i \in 1, \dots, p$, and the active inequality constraints $\nabla g_j(z)$, for all $j \in \mathcal{I}_{\mathrm{NL}}(z)$, are linearly independent.

Assumption 2.1. The matrices $\{H^{(i)}\}_{i=1}^{\infty}$ are positive definite, with bounded condition number, and smallest eigenvalue uniformly bounded away from zero, i.e., $\exists \gamma > 0$ such that, for all $i \in \mathbb{N}$, $d^\top H^{(i)} d \geq \gamma \|d\|_2^2$ for all $d \in \mathbb{R}^n$.

Standing Assumption 2.2. For all $i \in \mathbb{N}$, the QP in (2.3) is feasible.

Assumption 2.3. For all $i \in \mathbb{N}$, let $\mathcal{I}_{\mathrm{QP}}(d_{\mathrm{z}}^{(i)}, z^{(i)})$ denote the index set of the active inequality constraints in (2.3) parametric in $z^{(i)}$, i.e.,

$$\mathcal{I}_{\mathrm{QP}}(d_{\mathrm{z}}^{(i)}, z^{(i)}) := \left\{ j \in \{1, \dots, m\} \;\middle|\; g_j(z^{(i)}) + \nabla g_j(z^{(i)})^\top d_{\mathrm{z}}^{(i)} = 0 \right\}.$$

$d_{\mathrm{z}}^{(i)}$ is regular, i.e., the matrix made up of $\nabla h(z^{(i)})$ along with the columns $\nabla g_j(z^{(i)})$, $j \in \mathcal{I}_{\mathrm{QP}}(d_{\mathrm{z}}^{(i)}, z^{(i)})$, has full column rank. Strict complementarity holds.

Boundedness and uniqueness of the dual variables $\lambda_{\mathrm{QP}}^{(i)}, \nu_{\mathrm{QP}}^{(i)}$ in (2.4) then follow.

Standing Assumption 2.4. For all $i \in \mathbb{N}$, $z^{(i)}$, $z^{(i)} + d_{\mathrm{z}}^{(i)} \in \Omega$, for some compact set $\Omega \subset \mathbb{R}^n$.

Standing Assumption 2.5. The functions J, g, h and their first and second derivatives are uniformly bounded in norm in Ω.

Several choices for the Hessian $H^{(i)}$ have been considered in the literature. By setting $H^{(i)}$ as the Hessian of the Lagrangian of (2.1) and unit step size, local convergence to the desired $z^\star$ is achieved with a quadratic rate [23, 43, 44]. Other choices make the computation of $H^{(i)}$ less expensive, but deteriorate the convergence speed—see [7] for an overview of superlinear convergence theorems for SQP methods.

To ensure global convergence to a critical point, step sizes $t^{(i)} \neq 1$ are employed in SQP, together with a merit function as an augmented Lagrangian:

$$\begin{aligned} \mathcal{L}_{\mathrm{aug}}(z, \lambda, \nu, s, \rho) := \; & J(z) + (g(z) + s)^\top \lambda + h(z)^\top \nu \\ & + \frac{\rho}{2} \|g(z) + s\|_2^2 + \frac{\rho}{2} \|h(z)\|_2^2, \end{aligned} \tag{2.6}$$

where $\rho \geq 0$ is a penalty parameter to be determined and $s \in \mathbb{R}^m_{\geq 0}$ is a vector of slack variables, defined at the beginning of each iteration i such that its jth component

satisfies the following equation [22, equation (2.8)]:

$$s_j^{(i)} := \begin{cases} \max\left\{0, -g_j(z^{(i)})\right\} & \text{if } \rho = 0, \\ \max\left\{0, -g_j(z^{(i)}) - \frac{\lambda_j}{\rho}\right\} & \text{otherwise.} \end{cases} \tag{2.7}$$

Whenever $\rho \neq 0$, the vector s in (2.7) yields the value of $\mathcal{L}_{\text{aug}}$ minimized with respect to the slack variables only, subject to the nonnegativity constraint, $s \geq 0$.

For the design of the dual variables, diverse options are possible, e.g., least square estimate, dependent on z and on the Jacobian matrices of objective and constraints [2, 38], or constant estimates [24, 37]. The latter reduces the computational burden but leads to technical difficulties for proving global convergence.

In this paper, we build upon the approach in [22], where λ and ν are considered as additional variables, updated with step size $t^{(i)}$ along with the primal sequence $\left(z^{(i)}\right)_{i\in\mathbb{N}}$. Specifically, in view of [22], we consider the iterative update

$$\begin{bmatrix} z^{(i+1)} \\ \lambda^{(i+1)} \\ \nu^{(i+1)} \\ s^{(i+1)} \end{bmatrix} := \begin{bmatrix} z^{(i)} \\ \lambda^{(i)} \\ \nu^{(i)} \\ s^{(i)} \end{bmatrix} + t^{(i)} \begin{bmatrix} d_{\text{z}}^{(i)} \\ d_{\lambda}^{(i)} \\ d_{\nu}^{(i)} \\ d_{\text{s}}^{(i)} \end{bmatrix}, \tag{2.8}$$

where $d_{\text{z}}^{(i+1)}$ is from (2.3), $d_{\lambda}^{(i)} := \lambda_{\text{QP}}^{(i)} - \lambda^{(i)}$, $d_{\nu}^{(i)} := \nu_{\text{QP}}^{(i)} - \nu^{(i)}$, and the slack variation $d_{\text{s}}^{(i)}$ satisfies

$$g(z^{(i)}) + \nabla g(z^{(i)})^{\top} d_{\text{z}}^{(i)} + s^{(i)} + d_{\text{s}}^{(i)} = 0. \tag{2.9}$$

Then, we define the function $\phi : \mathbb{R} \to \mathbb{R}$ as

$$\phi(t) := \mathcal{L}_{\text{aug}}\left(z + t d_{\text{z}}, \lambda + t d_{\lambda}, \nu + t d_{\nu}, s + t d_{\text{s}}, \rho\right) \tag{2.10}$$

to determine the step size t, e.g., via a backtracking line search starting from $t = 1$, that satisfies the Wolfe conditions [12, 34]:

$$\phi(t) - \phi(0) \leq \sigma_1\, t\, \phi'(0), \tag{2.11a}$$

$$|\phi'(t)| \leq -\sigma_2\, \phi'(0) \quad \text{or} \quad (t = 1 \text{ and } \phi'(1) \leq -\sigma_2 \phi'(0)) \tag{2.11b}$$

for some $0 < \sigma_1 \leq \sigma_2 < \frac{1}{2}$. For ease of notation, we avoid making explicit the dependence of ϕ on the arguments z, λ, ν, s of the augmented Lagrangian $\mathcal{L}_{\text{aug}}$.

Note that if the derivative $\phi'(0)$ is negative, then there exists a step size $t^{(i)} \in (0, 1]$ such that the conditions in (2.11) hold. The condition on the derivative $\phi'(0)$ is checked numerically at every iteration $i \in \mathbb{N}$ via the inequality condition

$$\phi'(0) \leq -\frac{1}{2}(d_{\text{z}}^{(i)})^{\top} H^{(i)} d_{\text{z}}^{(i)}. \tag{2.12}$$

If this latter inequality does not hold true, then the parameter ρ is adjusted. In particular, there exists a lower bound $\hat{\rho} \in \mathbb{R}_{\geq 0}$ such that the inequality in (2.12) holds for all $\rho \geq \hat{\rho}$ [22, Lemma 4.3].

For the practical implementation, the line search in (2.11) is typically simplified in order to check only the first condition in (2.11a) [11, 35]. This has the effect of reducing the computational burden required to compute the derivative $\phi'(t)$, and it does not impede convergence of the algorithm in practice. To derive the step size t, a backtracking line search is employed with safeguarded polynomial interpolation [28].

The SQP steps for the NLP in (2.1) are summarized in Algorithm 1.

Algorithm 1. sequential quadratic programming.

INITIALIZE $i \leftarrow 0$, $z^{(0)} \in \mathbb{R}^n$ and $\rho^{(0)} = 0$
repeat
  COMPUTE $d_{\mathrm{z}}^{(i)}$ as in (2.3) and $\lambda_{\mathrm{QP}}^{(i)}$, $\nu_{\mathrm{QP}}^{(i)}$ such that (2.4) holds
  if $d_{\mathrm{z}}^{(i)} = 0$ **then**
    SET $z^\star = z^{(i)}$, $\lambda^\star = \lambda_{\mathrm{QP}}^{(i)}$, $\nu^\star = \nu_{\mathrm{QP}}^{(i)}$ and STOP
  else
    if $i = 0$ **then**
      SET $\lambda^{(0)} = \lambda_{\mathrm{QP}}^{(0)}$, $\nu^{(0)} = \nu_{\mathrm{QP}}^{(0)}$
    end if
    SET $d_\lambda^{(i)} = \lambda_{\mathrm{QP}}^{(i)} - \lambda^{(i)}$, $d_\nu^{(i)} = \nu_{\mathrm{QP}}^{(i)} - \nu^{(i)}$
  end if
  DETERMINE $s^{(i)}$ and $d_{\mathrm{s}}^{(i)}$ from (2.7) and (2.9)
  SET $\rho^{(i)} \geq 0$ such that (2.12) holds
  DETERMINE the step size $t^{(i)}$ that satisfies (2.11), e.g., via line search
  UPDATE $z^{(i+1)}, \lambda^{(i+1)}, \nu^{(i+1)}, s^{(i+1)}$ as in (2.8)
  $i \leftarrow i + 1$
until *Convergence*
return $z^\star$, $\lambda^\star$ and $\nu^\star$

3. Proposed variant to SQP. The SQP method presented in section 2 requires the computation of the primal and dual optimal solutions to each of the QPs in (2.3). Therefore, for all $i \in \mathbb{N}$, each QP has to be exactly solved to determine the updates $d_{\mathrm{z}}^{(i)}$, $d_\lambda^{(i)}$, and $d_\nu^{(i)}$.

In this paper, we propose determining $d_{\mathrm{z}}^{(i)}$ through one projected gradient step onto the linearization of the constraints around $z^{(i)}$, i.e., the feasible set of the QP in (2.3). This is formalized by

$$d_{\mathrm{z}}^{(i)} := \Pi_{\mathcal{C}^{(i)}} \left(-\alpha^{(i)} \nabla J(z^{(i)}) \right), \tag{3.1}$$

with bounded gradient step size $\alpha^{(i)} \in \mathbb{R}_{>0}$, and $\Pi_{\mathcal{C}^{(i)}}(\cdot) : \mathbb{R}^n \to \mathcal{C}^{(i)} \subseteq \mathbb{R}^n$ being the Euclidean projection onto the set

$$\mathcal{C}^{(i)} := \left\{ d_{\mathrm{z}} \in \mathbb{R}^n \;\middle|\; g(z^{(i)}) + \nabla g(z^{(i)})^\top d_{\mathrm{z}} \leq 0,\, h(z^{(i)}) + \nabla h(z^{(i)})^\top d_{\mathrm{z}} = 0 \right\}. \tag{3.2}$$

Note that the algorithm step in (3.1) is equivalent to computing only one gradient step of the QP in (2.3) with null initialization $d_{\mathrm{z}}^{\mathrm{ini}}$:

$$d_{\mathrm{z}}^{(i)} = \Pi_{\mathcal{C}^{(i)}} \left(\underbrace{d_{\mathrm{z}}^{\mathrm{ini}}}_{=0} - \alpha^{(i)} \left(\underbrace{H^{(i)} d_{\mathrm{z}}^{\mathrm{ini}}}_{=0} + \nabla J(z^{(i)}) \right) \right) = \Pi_{\mathcal{C}^{(i)}} \left(-\alpha^{(i)} \nabla J(z^{(i)}) \right).$$

In section 4, more detail is given about the choice of α for the convergence of the algorithm. Note that the projection in (3.1) is equivalent to solving the following

optimization problem.

$$
\begin{aligned}
d_{\mathrm{z}}^{(i)} = \arg\min_{d_{\mathrm{z}} \in \mathbb{R}^n} \quad & \frac{1}{2\alpha^{(i)}} \left\| d_{\mathrm{z}} + \alpha^{(i)} \nabla J(z^{(i)}) \right\|_2^2 \\
\text{s.t.} \quad & g(z^{(i)}) + \nabla g(z^{(i)})^\top d_{\mathrm{z}} \leq 0, \\
& h(z^{(i)}) + \nabla h(z^{(i)})^\top d_{\mathrm{z}} = 0.
\end{aligned} \tag{3.3}
$$

Therefore, there exist dual multipliers $\lambda_{\mathrm{G}}^{(i)} \in \mathbb{R}_{\geq 0}^m$, $\nu_{\mathrm{G}}^{(i)} \in \mathbb{R}^p$ such that

$$
\begin{aligned}
& \frac{1}{\alpha^{(i)}} d_{\mathrm{z}}^{(i)} + \nabla J(z^{(i)}) + \nabla g(z^{(i)}) \lambda_{\mathrm{G}}^{(i)} + \nabla h(z^{(i)}) \nu_{\mathrm{G}}^{(i)} = 0, \\
& \operatorname{diag}(\lambda_{\mathrm{G}}^{(i)}) \left(g(z^{(i)}) + \nabla g(z^{(i)})^\top d_{\mathrm{z}}^{(i)} \right) = 0, \\
& h(z^{(i)}) + \nabla h(z^{(i)})^\top d_{\mathrm{z}}^{(i)} = 0.
\end{aligned} \tag{3.4}
$$

The dual variables increments then are

$$
\begin{aligned}
d_{\lambda}^{(i)} &:= \lambda_{\mathrm{G}}^{(i)} - \lambda^{(i)}, \\
d_{\nu}^{(i)} &:= \nu_{\mathrm{G}}^{(i)} - \nu^{(i)},
\end{aligned} \tag{3.5}
$$

while we define the slack variable $s^{(i)}$ and variation $d_{\mathrm{s}}^{(i)}$ as in the SQP method, i.e., from (2.7) and (2.9), respectively.

Analogously to the SQP method reviewed in subsection 2.1, the augmented Lagrangian function in the form (2.6) is considered. The dual variables are updated along with the primal variables according to the update equation in (2.8), with step size $t^{(i)} \in (0, 1]$ defined such that the conditions in (2.11) hold.

We choose the penalty parameter $\rho \in \mathbb{R}_{\geq 0}$ such that the condition

$$
\phi'(0) \leq -\frac{1}{2\alpha^{(i)}} \left\| d_{\mathrm{z}}^{(i)} \right\|_2^2 \tag{3.6}
$$

is satisfied with $\phi(\cdot)$ defined as in (2.10). Later, in Lemma 4.4, we provide a lower bound $\hat{\rho}$ such that (3.6) holds for all $\rho \geq \hat{\rho}$.

Our proposed approach is summarized in Algorithm 2.

4. Proof of convergence of the proposed algorithm. In this section, we show the convergence properties of the proposed approach in Algorithm 2, under the standing assumptions of the SQP method in subsection 2.1 and the following assumption that replaces Assumption 2.3.

Assumption 4.1. For all $i \in \mathbb{N}$ and $z^{(i)}$, let $\mathcal{I}_{\mathrm{G}}(d_{\mathrm{z}}^{(i)}, z^{(i)})$ denote the index set of the active constraints in (3.3), i.e.,

$$
\mathcal{I}_{\mathrm{G}}(d_{\mathrm{z}}^{(i)}, z^{(i)}) = \left\{ j \in \{1, \ldots, m\} \;\middle|\; g_j(z^{(i)}) + \nabla g_j(z^{(i)})^\top d_{\mathrm{z}}^{(i)} = 0 \right\}. \tag{4.1}
$$

Then $d_{\mathrm{z}}^{(i)}$ is regular, i.e., the matrix made up of $\nabla h(z^{(i)})$ along with the columns $\nabla g_j(z^{(i)})$, $\forall j \in \mathcal{I}_{\mathrm{G}}(d_{\mathrm{z}}^{(i)}, z^{(i)})$, has full column rank. Furthermore, strict complementarity holds. □

Note that Assumption 4.1 implies that the dual variables $(\lambda_{\mathrm{G}}^{(i)}, \nu_{\mathrm{G}}^{(i)})$ in (3.4) are bounded and unique. Also note that the problem in (3.3) has the same feasible set as (2.3), thus by Standing Assumption 2.2, the projection in (3.1) is always feasible.

Algorithm 2. Variant to SQP.

INITIALIZE $i \leftarrow 0$, $z^{(0)} \in \mathbb{R}^n$, and $\rho^{(0)} = 0$
repeat
  COMPUTE $d_{\mathrm{z}}^{(i)}$ with step size $\alpha^{(i)}$ as in (3.1)
  DETERMINE $\lambda_{\mathrm{G}}^{(i)}$, $\nu_{\mathrm{G}}^{(i)}$ such that (3.4) holds
  if $d_{\mathrm{z}}^{(i)} = 0$ **then**
    SET $z^\star = z^{(i)}$, $\lambda^\star = \lambda_{\mathrm{G}}^{(i)}$, $\nu^\star = \nu_{\mathrm{G}}^{(i)}$ and STOP
  else
    if $i = 0$ **then**
      SET $\lambda^{(0)} = \lambda_{\mathrm{G}}^{(0)}$, $\nu^{(0)} = \nu_{\mathrm{G}}^{(0)}$
    end if
    SET $d_\lambda^{(i)} = \lambda_{\mathrm{G}}^{(i)} - \lambda^{(i)}$, $d_\nu^{(i)} = \nu_{\mathrm{G}}^{(i)} - \nu^{(i)}$
  end if
  DETERMINE $s^{(i)}$ and $d_{\mathrm{s}}^{(i)}$ from (2.7) and (2.9)
  SET $\rho^{(i)} \geq 0$ such that (3.6) holds
  DETERMINE the step size $t^{(i)}$ that satisfies (2.11), e.g., via line search
  UPDATE $z^{(i+1)}, \lambda^{(i+1)}, \nu^{(i+1)}, s^{(i+1)}$ as in (2.8)
  $i \leftarrow i + 1$
until *Convergence*
return $z^\star$, $\lambda^\star$ and $\nu^\star$

This section is organized as follows: in subsection 4.1, we derive conditions for local linear convergence (setting $t = 1$ in (2.11)) under additional assumptions on the Hessian at the critical point and on the step size α. These conditions are not required to prove the global convergence of the algorithm in subsection 4.2, albeit for $t \leq 1$ the convergence can be theoretically slower than linear. We finally show that the linear convergence rate is not precluded close to the solution, since $t = 1$ is admissible by the line search in subsection 4.3.

4.1. Local convergence. According to [8, 41], we define the general recursive algorithm as a method to determine a critical point for the NLP in (2.1) via intermediate iterates $w^{(i)} := \left(z^{(i)}, \lambda^{(i)}, \nu^{(i)}\right)$, whose update $w^{(i+1)}$ is determined as the KKT triple of a specific optimization problem $\mathrm{P}(w^{(i)})$. Given the generic optimization problem,

$$\mathrm{P}(w^{(i)}): \quad z^{(i+1)} = \arg\min_{z} \ \mathbb{J}(z, w^{(i)}) \quad \text{s.t. } \mathrm{g}(z, w^{(i)}) \leq 0, \ \mathbb{h}(z, w^{(i)}) = 0, \tag{4.2}$$

the updates $\lambda^{(i+1)}$ and $\nu^{(i+1)}$ are the dual variables associated with the KKT conditions for $\mathrm{P}(w^{(i)})$. As in [41], and in line with the original NLP in (2.1), we assume that the functions $\mathbb{J}$, g, and $\mathbb{h}$ are twice continuously differentiable in their first argument. Let us define a KKT triple as $w := (z, \lambda, \nu)$ and the function

$$\mathcal{U}\left(w,\, w^{(i)}\right) := \Big[\nabla_w \mathbb{J}(z, w^{(i)}) + \lambda^\top \nabla_w \mathbb{g}(z, w^{(i)}) + \nu^\top \nabla_w \mathbb{h}(z, w^{(i)}); \\ \lambda_1 \mathbb{g}_1(z, w^{(i)}); \ldots; \lambda_m \mathbb{g}_m(z, w^{(i)}); \mathbb{h}_1(z, w^{(i)}); \ldots; \mathbb{h}_p(z, w^{(i)})\Big]. \tag{4.3}$$

The SQP methods and the proposed algorithm can be recast as general recursive algorithms of the form in (4.2). Let us first consider the SQP in Algorithm 1. A local version of this algorithm, which is not guaranteed to converge for any initialization $z^{(0)}$, takes step $t^{(i)} = 1$ for all $i \in \mathbb{N}$.

It follows from (2.3) that the update $w^{(i+1)} = (z^{(i+1)}, \lambda^{(i+1)}, \nu^{(i+1)})$ associated with (4.2) is the KKT triple of the problem $\mathrm{P}_{\mathrm{QP}}(w^{(i)})$:

$$\mathrm{P}_{\mathrm{QP}}(w^{(i)}): \quad z^{(i+1)} = \arg\min_z \frac{1}{2}(z - z^{(i)})^\top H^{(i)}(z - z^{(i)}) + \nabla J(z^{(i)})^\top (z - z^{(i)}) \\ \text{s.t.} \quad g(z^{(i)}) + \nabla g(z^{(i)})^\top (z - z^{(i)}) \le 0, \\ h(z^{(i)}) + \nabla h(z^{(i)})^\top (z - z^{(i)}) = 0. \tag{4.4}$$

In fact, since $t^{(i)} = 1$, by (2.8) and (3.5), the update of the dual variables is given by $\left(\lambda^{(i+1)}, \nu^{(i+1)}\right) = (\lambda^{(i)}_{\mathrm{QP}}, \nu^{(i)}_{\mathrm{QP}})$.

Analogously, if we fix $t^{(i)} = 1$ for all $i \in \mathbb{N}$, it follows from (3.3) that the proposed algorithm determines the update $w^{(i+1)}$ as the KKT triple of the problem $\mathrm{P}_{\mathrm{G}}(w^{(i)})$:

$$\mathrm{P}_{\mathrm{G}}(w^{(i)}): \quad z^{(i+1)} = \arg\min_z \frac{1}{2\alpha^{(i)}}(z - z^{(i)})^\top (z - z^{(i)}) + \nabla J(z^{(i)})^\top (z - z^{(i)}) \\ \text{s.t.} \quad g(z^{(i)}) + \nabla g(z^{(i)})^\top (z - z^{(i)}) \le 0, \\ h(z^{(i)}) + \nabla h(z^{(i)})^\top (z - z^{(i)}) = 0. \tag{4.5}$$

Again, by (2.8) and (3.5), the update of the dual variables is $\left(\lambda^{(i+1)}, \nu^{(i+1)}\right) = (\lambda^{(i)}_{\mathrm{G}}, \nu^{(i)}_{\mathrm{G}})$.

The following result establishes some basic properties of the general recursive algorithm in (4.2) that will be necessary to establish local and global convergence of the proposed algorithm.

LEMMA 4.1 (see [41, Theorem 2.1]). *Let $\bar{w} \in \mathbb{R}^{n+m+p}$, and suppose that $\left(\bar{z}, \bar{\lambda}, \bar{\nu}\right) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p$ is a KKT triple of $P(\bar{w})$ from (4.2), at which the first-order conditions with strict complementarity slackness and linear independence of the gradients to the active constraints hold.*

Then, there exist open neighborhoods $W = W(\bar{w})$ and $V = V(\bar{z}, \bar{\lambda}, \bar{\nu})$, and a continuous function $Z : W \to V$, such that $Z(\bar{w}) = (\bar{z}, \bar{\lambda}, \bar{\nu})$ for all $w \in W$, $Z(w)$ is the unique KKT triple in V of $P(w)$ and the unique zero in V of the function $\mathcal{U}((\cdot), w)$ in (4.3). Furthermore, if $Z(w) =: (z(w), \lambda(w), \nu(w))$, then for each $w \in W$, $z(w)$ is a critical point of $P(w)$ at which the first-order KKT conditions are satisfied with strict complementarity slackness and linear independence of the gradients to the active constraints. □

Specifically, the same inequality constraints active at $z(\bar{w})$ will be active at $z(w)$, which is in accordance with [41, proof of Theorem 2.1]. However, since we assume *first-order* conditions with slack complementarity at $z(\bar{w})$, then at $z(w)$ the *first-order* conditions will hold with slack complementarity. The existence of the continuous function Z is guaranteed in our case by the implicit-function theorem, given our

assumptions that the functions $\mathbb{J}, \mathbb{g}$, and $\mathbb{h}$ are continuously differentiable and the fact that the matrix $\frac{\partial \mathcal{U}(\bar{w},\bar{w})}{\partial w}$ is invertible under the considered linear independence and strict complementarity assumptions [18, p. 292]. We refer to [48] for a weaker technical result without the strict complementarity assumption.

As in [2], for the analysis of local convergence, we assume that the correct active set at $z^\star$ is known. This is justified by Lemma 4.1, since the proposed algorithm will eventually identify the active inequality constraint for (2.1). Therefore, we define as $h_{\mathrm{a}}(z) = [g_{\mathrm{a}}(z)\,;\, h(z)]$ the set of active inequality and equality constraints at $z^\star$ and indicate with $\xi := [\lambda_{\mathrm{a}}\,;\,\nu]$ the corresponding dual variables.

The following result establishes the local convergence properties of the algorithm and applies SQP arguments to establish linear convergence to a critical point. In fact, the problem in (4.5) can be seen as an SQP program with Hessian $\frac{1}{\alpha} I \succ 0$.

THEOREM 4.1. *Assume that $(z^\star, \xi^\star)$ is a critical point such that $\nabla^2 \mathcal{L}(z^\star, \xi^\star)$ is positive definite, and let the initialization $z^{(0)}$ be close enough to $z^\star$. Then, there exist positive step sizes $(\alpha^{(i)})_{i\in\mathbb{N}}$ such that the sequence $\left(z^{(i)}\right)_{i\in\mathbb{N}}$ defined as in (4.5) converges to $z^\star$ with linear rate.*

Proof. The proof is similar to [3, proof of Theorem 3.3], albeit the result obtained is different. For ease of notation, we use no superscript for the iteration i and the superscript "+" for the iteration $i+1$.

Then the proposed algorithm has the following update when $t = 1$:

$$\begin{aligned}\xi^+ &= \xi + d_\xi = \xi + (\xi_{\mathrm{G}}(z) - \xi) = \xi_{\mathrm{G}}(z)\\ &= \left(\alpha \nabla h_{\mathrm{a}}(z)^\top \nabla h_{\mathrm{a}}(z)\right)^{-1} \left(h_{\mathrm{a}}(z) - \alpha \nabla h_{\mathrm{a}}(z) \nabla J(z)\right).\end{aligned}$$

From (2.2), we have the optimal dual variable

$$\xi^\star = -\left(\nabla h_{\mathrm{a}}(z^\star)^\top A\, \nabla h_{\mathrm{a}}(z^\star)\right)^{-1} \left(\nabla h_{\mathrm{a}}(z^\star)\, A\, \nabla J(z^\star)\right),$$

where A is any nonsingular matrix that is positive definite on the null space of $\nabla h_{\mathrm{a}}(z^\star)^\top$. In particular, with $A = \alpha I$, the following holds:

$$\begin{aligned}\nabla \xi_{\mathrm{G}}(z^\star) =& \left(\alpha \nabla h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star)\right)^{-1} \Big(\nabla h_{\mathrm{a}}(z^\star)^\top \left(I - \alpha \nabla^2 J(z^\star)\right) - \alpha \nabla^2 h_{\mathrm{a}}(z^\star)^\top \nabla J(z^\star)\\ &- \alpha \left(\nabla h_{\mathrm{a}}(z^\star)^\top \nabla^2 h_{\mathrm{a}}(z^\star) + \nabla^2 h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star)\right) \xi^\star\Big)\\ =& \left(\alpha \nabla h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star)\right)^{-1} \Big(\nabla h_{\mathrm{a}}(z^\star)^\top \left(I - \alpha \nabla^2 J(z^\star)\right) + \alpha \nabla^2 h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star) \xi^\star\\ &- \alpha \left(\nabla h_{\mathrm{a}}(z^\star)^\top \nabla^2 h_{\mathrm{a}}(z^\star) + \nabla^2 h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star)\right) \xi^\star\Big)\\ =& \left(\alpha \nabla h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star)\right)^{-1} \nabla h_{\mathrm{a}}(z^\star)^\top \left(I - \alpha \nabla^2 J(z^\star) - \alpha \nabla^2 h_{\mathrm{a}}(z^\star) \xi^\star\right)\\ =& \left(\alpha \nabla h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star)\right)^{-1} \nabla h_{\mathrm{a}}(z^\star)^\top \left(I - \alpha \nabla^2 \mathcal{L}(z^\star, \xi^\star)\right),\end{aligned}$$

where the first equality follows from (2.2). Thus

(4.6)

$$\begin{aligned}&\xi^+ - \xi^\star = \xi_{\mathrm{G}}(z) - \xi_{\mathrm{G}}(z^\star) = \nabla \xi_{\mathrm{G}}(z^\star)(z - z^\star) + \mathcal{O}\left(\|z - z^\star\|_2^2\right)\\ &= \left(\alpha \nabla h_{\mathrm{a}}(z^\star)^\top \nabla h_{\mathrm{a}}(z^\star)\right)^{-1} \alpha \nabla h_{\mathrm{a}}(z^\star)^\top \left(\frac{1}{\alpha} I - \nabla^2 \mathcal{L}(z^\star, \xi^\star)\right)(z - z^\star) + \mathcal{O}\left(\|z - z^\star\|_2^2\right).\end{aligned}$$

From (3.4), we have the update $d_z = -\alpha \nabla \mathcal{L}(z, \xi_G(z))$; therefore,

(4.7)
$$\begin{aligned} z^+ - z^\star &= z + d_z - z^\star = z - z^\star - \alpha(\nabla \mathcal{L}(z, \xi_G(z)) - \underbrace{\nabla \mathcal{L}(z^\star, \xi^\star)}_{=0}) \\ &= z - z^\star - \alpha \left(\nabla^2 \mathcal{L}(z^\star, \xi^\star)(z - z^\star) + \frac{\partial \nabla \mathcal{L}(z^\star, \xi^\star)}{\partial \xi} (\xi_G(z) - \xi^\star) \right) + \mathcal{O}\left(\|z - z^\star\|_2^2 \right) \\ &= z - z^\star - \alpha \left(\nabla^2 \mathcal{L}(z^\star, \xi^\star)(z - z^\star) + \nabla h_a(z^\star)(\xi_G(z) - \xi^\star) \right) + \mathcal{O}\left(\|z - z^\star\|_2^2 \right) \\ &= \alpha \left(\left(\frac{1}{\alpha} I - \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right)(z - z^\star) - \nabla h_a(z^\star)(\xi_G(z) - \xi^\star) \right) + \mathcal{O}\left(\|z - z^\star\|_2^2 \right). \end{aligned}$$

Now, by substituting (4.6) into (4.7),

$$\begin{aligned} z^+ - z^\star = \alpha \Bigg(& \left(\frac{1}{\alpha} I - \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right)(z - z^\star) - \nabla h_a(z^\star) \left(\alpha \nabla h_a(z^\star)^\top \nabla h_a(z^\star) \right)^{-1} \\ & \cdot \alpha \nabla h_a(z^\star)^\top \left(\frac{1}{\alpha} I - \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right)(z - z^\star) \Bigg) + \mathcal{O}\left(\|z - z^\star\|_2^2 \right) \\ = \alpha T(z^\star) & \left(\frac{1}{\alpha} I - \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right)(z - z^\star) + \mathcal{O}\left(\|z - z^\star\|_2^2 \right) \\ = T(z^\star) & \left(I - \alpha \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right)(z - z^\star) + \mathcal{O}\left(\|z - z^\star\|_2^2 \right), \end{aligned}$$

with $T(z^\star) := I - \nabla h_a(z^\star) \left(\nabla h_a(z^\star)^\top \nabla h_a(z^\star) \right)^{-1} \nabla h_a(z^\star)^\top$ being the orthogonal projector onto the tangent space to the constraints $h_a(z^\star)$ at $z^\star$.

Then by considering the norms of the above quantities we conclude that

$$\begin{aligned} \left\| z^+ - z^\star \right\|_2 &\le \left\| T(z^\star) \left(I - \alpha \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right)(z - z^\star) \right\|_2 + \gamma \|z - z^\star\|_2^2 \\ &\le \Big(\underbrace{\left\| I - \alpha \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right\|_2 + \gamma \|z - z^\star\|_2}_{=:\eta} \Big) \|z - z^\star\|_2 \end{aligned}$$

for all sufficiently large iterations and some $\gamma > 0$, independent of the iteration. Here we have used the property of the orthogonal projector $\|T(z^\star)\, v\|_2 \le \|v\|_2$ for any vector v. Since $\nabla^2 \mathcal{L}(z^\star, \xi^\star) \succ 0$, by choosing $\alpha \le \frac{1}{\max \operatorname{eig} \nabla^2 \mathcal{L}(z^\star, \xi^\star)}$, the term $\left\| I - \alpha \nabla^2 \mathcal{L}(z^\star, \xi^\star) \right\|_2$ can be made strictly smaller than 1. Thus, for a sufficiently small initialization distance $\left\| z^{(0)} - z^\star \right\|_2$, we have $\eta < 1$, and the sequence $\left(z^{(i)} \right)_i$ converges at a linear rate due to the contraction mapping theorem. ∎

4.2. Global convergence. Before showing the convergence result of the paper, some technical lemmas are presented that show the properties of the proposed algorithm. In particular, we will show that a sufficient decrease of the merit function can be obtained at every iteration whenever we are not at a critical point. This argument will require some technical lemmas showing boundedness of some quantities and giving a tuning rule for the penalty parameter $\rho^{(i)}$. We start with the following lemma, which ensures that the desired algorithm determines the correct active set at a critical point of (2.1).

LEMMA 4.2. *The following properties hold for Algorithm* 2*:*

(i) $||d_z^{(i)}||_2 = 0$ *if and only if* $z^{(i)}$ *is a critical point for* (2.1)*;*

(ii) *there exists* $\bar{\varepsilon} \in \mathbb{R}_{>0}$ *such that if* $\|d_z\|_2 \leq \bar{\varepsilon}$, *then the active set* $\mathcal{I}_G$ *in* (4.1) *of* (3.3) *coincides with the set of constraints that are active at a critical point* $z^\star$ *for* (2.1).

Proof. (i) We first prove that, if $||d_z^{(i)}||_2 = 0$, then $z^{(i)}$ is a KKT point for (2.1). Note that $||d_z^{(i)}||_2 = 0$ implies $d_z^{(i)} = 0$, i.e., that there exist $\lambda_G^{(i)}$ and $\nu_G^{(i)}$ such that (3.4) holds with $d_z^{(i)} = 0$. Therefore, by setting $z^\star = z^{(i)}$, $\lambda^\star = \lambda_G^{(i)}$, and $\nu^\star = \nu_G^{(i)}$, the KKT conditions in (2.2) are satisfied. Strict complementarity follows from Assumption 4.1.

Conversely, if $z^{(i)}$ is a critical point for (2.1), then (2.2) holds for $z^\star = z^{(i)}$, $\lambda^\star = \lambda_G^{(i)}$, and $\nu^\star = \nu_G^{(i)}$. Now, suppose that the vector $d_z^{(i)}$ resulting from the projection in (3.1) is nonzero, i.e., there exist $\lambda_G^{(i)} \in \mathbb{R}^m_{\geq 0}$ and $\nu_G^{(i)} \in \mathbb{R}^p$ such that (3.4) holds for some $d_z^{(i)} \neq 0$, with strict complementarity by Assumption 4.1. On the other hand, because of (2.2), the KKT conditions in (3.4) hold also for $d_z = 0$, with dual variables $\bar{\lambda}_G^{(i)} \in \mathbb{R}^m_{\geq 0}$ and $\bar{\nu}_G^{(i)} \in \mathbb{R}^p$. This generates a contradiction, because the projection (3.1) onto the convex set $\mathcal{C}^{(i)}$, which is nonempty by Standing Assumption 2.2, is unique. (ii) This result follows from Lemma 4.1 since the proposed method can be rewritten as a general recursive algorithm in the form of (4.2). ∎

The following lemma derives a bound for the dual variables $\lambda^{(i)}$ and $\nu^{(i)}$.

LEMMA 4.3. *For all* $i \in \mathbb{N}$, *it holds that*

$$\left\|\lambda^{(i+1)}\right\|_2 \leq \max_{k\in[0,i]} \left\|\lambda_G^{(k)}\right\|_2, \quad \left\|\nu^{(i+1)}\right\|_2 \leq \max_{k\in[0,i]} \left\|\nu_G^{(k)}\right\|_2.$$

In addition, $||d_\lambda^{(i)}||_2$ *and* $||d_\nu^{(i)}||_2$ *are uniformly bounded for all* $i \in \mathbb{N}$.

Proof. First note that the dual variables $\lambda_G^{(i)}$ and $\nu_G^{(i)}$ defined in (3.4) are bounded in norm, since by Assumption 4.1 the active set is linearly independent and strong duality holds.

Then the proof follows the same argument of [22, Lemma 4.2], where the structure of (3.5) is exploited. We equivalently define the dual variable for the inequality constraints as

$$\begin{aligned} \lambda^{(0)} &:= \lambda_G^{(0)}, \\ \lambda^{(i+1)} &:= \lambda^{(i)} + t^{(i)}(\lambda_G^{(i)} - \lambda^{(i)}) \quad \forall i \in \mathbb{N}. \end{aligned} \tag{4.8}$$

We proceed by induction. The result holds for $\lambda^{(0)}$. Now we assume that the result holds for $\lambda^{(i)}$. Then, since $t^{(i)} \in (0, 1]$ we have that

$$\begin{aligned} \left\|\lambda^{(i+1)}\right\|_2 &= t^{(i)} \left\|\lambda_G^{(i)}\right\|_2 + (1 - t^{(i)}) \left\|\lambda^{(i)}\right\|_2 \\ &\leq t^{(i)} \left\|\lambda_G^{(i)}\right\|_2 + (1 - t^{(i)}) \max_{k\in[0,i-1]} \left\|\lambda_G^{(k)}\right\|_2 \\ &\leq t^{(i)} \max_{k\in[0,i]} \left\|\lambda_G^{(k)}\right\|_2 + (1 - t^{(i)}) \max_{k\in[0,i]} \left\|\lambda_G^{(k)}\right\|_2 \\ &= \max_{k\in[0,i]} \left\|\lambda_G^{(k)}\right\|_2. \end{aligned}$$

This proves the boundedness of $\left\|\lambda^{(i)}\right\|_2$, since both $\lambda_G^{(i)}$ and $d_\lambda^{(i)}$ are bounded by (3.5).

The proof for the boundedness of the dual variables associated to the equality constraints is analogous. ∎

The following lemma serves as a tuning rule for the parameter $\rho^{(i)}$.

LEMMA 4.4. *There exists* $\hat{\rho}^{(i)} \in \mathbb{R}_{\geq 0}$ *such that*

$$\sup_{\rho \geq \hat{\rho}^{(i)}} \phi'(0, \rho) \leq -\frac{1}{2\alpha^{(i)}} \left\| d_{\mathrm{z}}^{(i)} \right\|_2^2,$$

where ϕ *is as in* (2.10) *and* $\alpha^{(i)}$ *and* $d_{\mathrm{z}}^{(i)}$ *are from* (3.1).

Proof. For a given $\rho \in \mathbb{R}_{\geq 0}$, the gradient of $\mathcal{L}_{\mathrm{aug}}$ in (2.6) is

$$\begin{bmatrix} \nabla J(z) + \nabla g(z)\lambda + \nabla h(z)\nu + \rho \nabla g(z)(g(z)+s) + \rho \nabla h(z)\, h(z) \\ g(z) + s \\ h(z) \\ \lambda + \rho\,(g(z)+s) \end{bmatrix}.$$

Therefore, using a simplified notation, we have

$$\begin{aligned} \phi'(0) &= d_{\mathrm{z}}^{\top}\,(\nabla J + \nabla g\,\lambda + \nabla h\,\nu + \rho \nabla g\;(g+s) + \rho \nabla h\, h) \\ &\quad + d_{\lambda}^{\top}(g+s) + d_{\nu}^{\top} h + d_{\mathrm{s}}^{\top}\;(\lambda + \rho(g+s)) \\ &= d_{\mathrm{z}}^{\top} \nabla J + (\nabla g^{\top} d_{\mathrm{z}} + d_{\mathrm{s}})^{\top} \lambda + \rho(\nabla g^{\top} d_{\mathrm{z}} + d_{\mathrm{s}})^{\top}(g+s) \\ &\quad + d_{\mathrm{z}}^{\top} \nabla h \nu + d_{\lambda}^{\top}(g+s) + d_{\nu}^{\top} h + \rho(\nabla h^{\top} d_{\mathrm{z}})^{\top} h \\ &= d_{\mathrm{z}}^{\top} \nabla J - (g+s)^{\top}(\lambda - d_{\lambda}) - \rho(g+s)^{\top}(g+s) \\ &\quad - h^{\top}(\nu - d_{\nu}) - \rho h^{\top} h, \end{aligned}$$

where the last step follows from (2.9) and the last equation in (3.4), as these imply that d_{z} satisfies

$$\nabla g^{\top} d_{\mathrm{z}} + d_{\mathrm{s}} = -(g+s), \tag{4.9a}$$

$$\nabla h^{\top} d_{\mathrm{z}} = -h. \tag{4.9b}$$

By rearranging the first equation in (3.4) and using the definition of λ_{G} and ν_{G} we have

$$\nabla J = -\frac{1}{\alpha} d_{\mathrm{z}} - \nabla g \lambda_{\mathrm{G}} - \nabla h \nu_{\mathrm{G}},$$

which, combined with (4.9a), yields

$$\begin{aligned} \phi'(0) &= -\frac{1}{\alpha} d_{\mathrm{z}}^{\top} d_{\mathrm{z}} - d_{\mathrm{z}}^{\top} \nabla g\, \lambda_{\mathrm{G}} - d_{\mathrm{z}}^{\top} \nabla h\, \nu_{\mathrm{G}} - (g+s)^{\top}(\lambda - d_{\lambda}) \\ &\quad - \rho(g+s)^{\top}(g+s) - h^{\top}(\nu - d_{\nu}) - \rho\, h^{\top} h \\ &= -\frac{1}{\alpha} d_{\mathrm{z}}^{\top} d_{\mathrm{z}} + d_{\mathrm{s}}^{\top} \lambda_{\mathrm{G}} + (g+s)^{\top} \lambda_{\mathrm{G}} - (g+s)^{\top}(\lambda - d_{\lambda}) \\ &\quad - \rho(g+s)^{\top}(g+s) + h^{\top} \nu_G - h^{\top}(\nu - d_{\nu}) - \rho\, h^{\top} h. \end{aligned}$$

By (3.5) and the right-hand side of (3.6), we want to prove that

$$\begin{aligned} & d_{\mathrm{s}}^{\top} \lambda_{\mathrm{G}} + 2(g+s)^{\top} d_{\lambda} - \rho\,(g+s)^{\top}(g+s) + 2h^{\top} d_{\nu} - \rho\, h^{\top} h \\ &= d_{\mathrm{s}}^{\top} \lambda_{\mathrm{G}} + 2 \begin{bmatrix} (g+s)^{\top} & h^{\top} \end{bmatrix} \begin{bmatrix} d_{\lambda} \\ d_{\nu} \end{bmatrix} - \rho(g+s)^{\top}(g+s) - \rho\, h^{\top} h \\ &\leq \frac{1}{2\alpha} \|d_{\mathrm{z}}\|_2^2 \end{aligned}$$

for a specific choice of ρ. Note that $d_{\mathrm{s}}^\top \lambda_{\mathrm{G}} \leq 0$ because of (2.9) and the complementarity conditions in (3.4). The determination of $\hat{\rho}$ is nontrivial only if

$$2\begin{bmatrix}(g+s)^\top & h^\top\end{bmatrix}\begin{bmatrix}d_\lambda\\ d_\nu\end{bmatrix} \geq \frac{1}{2\alpha}\|d_{\mathrm{z}}\|_2^2; \tag{4.10}$$

otherwise we can take $\hat{\rho}=0$. Hence, if (4.10) holds, then we take $\hat{\rho}$ such that

$$2\begin{bmatrix}(g+s)^\top & h^\top\end{bmatrix}\begin{bmatrix}d_\lambda\\ d_\nu\end{bmatrix} \leq 2\left\|\begin{bmatrix} g+s\\ h\end{bmatrix}\right\|_2\left\|\begin{bmatrix} d_\lambda\\ d_\nu\end{bmatrix}\right\|_2 \leq \hat{\rho}\left\|\begin{bmatrix} g+s\\ h\end{bmatrix}\right\|_2^2.$$

This is equivalent to

$$\hat{\rho} = 2\left(\left\|\left[\begin{smallmatrix} d_\lambda\\ d_\nu\end{smallmatrix}\right]\right\|_2\right)/\left\|\left[\begin{smallmatrix} g+s\\ h\end{smallmatrix}\right]\right\|_2. \tag{4.11}$$

∎

Therefore, we can define the penalty parameter at the beginning of each iteration i as follows:

$$\rho^{(i)} := \begin{cases}\rho^{(i-1)} & \text{if } \phi'(0,\rho^{(i-1)}) \leq -\frac{1}{2\alpha^{(i)}}\left\|d_{\mathrm{z}}^{(i)}\right\|_2^2,\\ \max\{\hat{\rho}^{(i)},\, 2\rho^{(i-1)}\} & \text{otherwise,}\end{cases} \tag{4.12}$$

where $\hat{\rho}^{(i)}=\hat{\rho}$ as in (4.11) with d_λ, d_ν, z, and s evaluated at the current iteration i.

Remark 4.1. Note that the parameter $\rho^{(i)}$ can possibly diverge for $i\to\infty$, if there exists an infinite set of iterations $\{i_l\}_l$ where the parameter strictly increases. The statements given next consider this possibility and prove convergence in a general case.

LEMMA 4.5. *Suppose $\{i_l\}_{l\in\mathbb{N}}$ is the set of iterations in which the penalty parameter $\rho^{(i_l)}$ increases. Then,*

$$\rho^{(i_l)}\left\|d_{\mathrm{z}}^{(i_l)}\right\|_2^2 \leq N_\rho, \tag{4.13a}$$

$$\rho^{(i_l)}\left\|\begin{bmatrix} g(z^{(i_l)})+s^{(i_l)}\\ h(z^{(i_l)})\end{bmatrix}\right\|_2 \leq N_\rho \tag{4.13b}$$

for some $N_\rho\in\mathbb{R}_{>0}$.

Proof. The argument of the functions and the index i_ρ are dropped for ease of notation. In order for the penalty parameter to increase, the conditions in (4.10) must hold; that is,

$$\frac{1}{2\alpha}\|d_{\mathrm{z}}\|_2^2 \leq 2\begin{bmatrix}(g+s)^\top & h^\top\end{bmatrix}\begin{bmatrix}d_\lambda\\ d_\nu\end{bmatrix} \leq 2\left\|\begin{bmatrix} g+s\\ h\end{bmatrix}\right\|_2\left\|\begin{bmatrix} d_\lambda\\ d_\nu\end{bmatrix}\right\|_2,$$

and hence

$$\left\|\begin{bmatrix} g+s\\ h\end{bmatrix}\right\|_2 \geq \frac{1}{4\alpha}\frac{\|d_{\mathrm{z}}\|_2^2}{\left\|\begin{bmatrix} d_\lambda\\ d_\nu\end{bmatrix}\right\|_2}.$$

By substituting this last inequality into the definition of $\hat{\rho}$, we have that

$$\hat{\rho} = \frac{2\left\|\begin{bmatrix} d_\lambda \\ d_\nu \end{bmatrix}\right\|_2}{\left\|\begin{bmatrix} g+s \\ h \end{bmatrix}\right\|_2} \leq 8\alpha \frac{\left\|\begin{bmatrix} d_\lambda \\ d_\nu \end{bmatrix}\right\|_2^2}{\|d_{\mathrm{z}}\|_2^2},$$

and the desired result in (4.13a) holds due to Lemma 4.3. The following relation proves (4.13b):

$$\rho\left\|\begin{bmatrix} g+s \\ h \end{bmatrix}\right\|_2 \leq 2\hat{\rho}\left\|\begin{bmatrix} g+s \\ h \end{bmatrix}\right\|_2 = 2\frac{2\left\|\begin{bmatrix} d_\lambda \\ d_\nu \end{bmatrix}\right\|_2}{\left\|\begin{bmatrix} g+s \\ h \end{bmatrix}\right\|_2}\left\|\begin{bmatrix} g+s \\ h \end{bmatrix}\right\|_2 = 4\left\|\begin{bmatrix} d_\lambda \\ d_\nu \end{bmatrix}\right\|_2. \quad \square \tag{4.14}$$

The following two lemmas, provided without proofs, give intermediate technical results that are required for the main results. Their proofs follow the same arguments in [22] with minor adjustments to reflect the update rule in Algorithm 2.

LEMMA 4.6 (see [22, Lemma 4.6]). *Let $\{i_l\}_{l\in\mathbb{N}}$ denote the set of iterations for which the parameter $\rho^{(i_l)}$ increases. Then, there exists $M \in \mathbb{R}_{>0}$ such that, for all $l \in \mathbb{N}$,*

$$\rho^{(i_l)} \sum_{i=i_l}^{i_{l+1}-1} \left\|t^{(i)} d_{\mathrm{z}}^{(i)}\right\|_2^2 < M. \tag{4.15}$$

$\square$

LEMMA 4.7 ([22, Lemma 4.9]). *The step size $t^{(i)}$ defined according to (2.11) satisfies $\phi(t^{(i)}) - \phi(0) \leq \sigma_1 t^{(i)}\phi'(0)$, where $\sigma_1 < \frac{1}{2}$ and $t^{(i)} > \bar{t}$, for some $\bar{t} > 0$ independent of i.* $\square$

We are now ready to state the main result of the paper, that is, the global convergence of our proposed algorithm.

THEOREM 4.2. *Algorithm 2 is such that* $\lim_{i\to\infty} ||d_{\mathrm{z}}^{(i)}||_2 = 0$.

Proof. The proof is similar to [22, proof of Theorem 4.1]. If $\|d_{\mathrm{z}}^{(i)}\|_2 = 0$ for a finite i, then the algorithm terminates and the statement is true. We assume in the following that $\|d_{\mathrm{z}}^{(i)}\|_2 \neq 0$ for all $i \in \mathbb{N}$.

If there is no upper bound on ρ, then the uniform lower bound $\bar{t} > 0$ from Lemma 4.7 and (4.15) implies that, for all $\delta > 0$, there exists $\tilde{i} \in \mathbb{N}$ such that $\|d_{\mathrm{z}}^{(i)}\|_2 \leq \delta$ for all $i \geq \tilde{i}$, which proves the statement.

In the bounded case, there exists a value $\tilde{\rho}$ and an index $\tilde{i}$ such that $\rho^{(i)} = \tilde{\rho}$ for all $i \geq \tilde{i}$. The proof is then by contradiction. We assume that there exist $\varepsilon > 0$ and $\tilde{i} \in \mathbb{N}$ such that $\|d_{\mathrm{z}}^{(i)}\|_2 > \varepsilon$ for all $i \geq \tilde{i}$. Now, every subsequent iteration must yield a decrease in the merit function in (2.6) with $\rho = \tilde{\rho}$, since because of (2.11), Lemma 4.4, and Lemma 4.7, we have

$$\phi(t^{(i)}) - \phi(0) \leq \sigma_1 t^{(i)}\phi'(0) \leq -\frac{1}{2\alpha^{(i)}}\sigma_1 \bar{t}\varepsilon^2 < 0,$$

where the step size $\alpha^{(i)} > 0$ is designed to be bounded. The addition of the slack variable $s^{(i)}$ in (2.7) can only lead to a further reduction in the merit function. Therefore, since the merit function with $\rho^{(i)} = \tilde{\rho}$ decreases by at least a fixed quantity at every iteration, it must be unbounded from below. Since by Lemma 4.3 the dual variables $\lambda^{(i)}$ and $\nu^{(i)}$ are bounded, the merit function in (2.6) can be unbounded from below only if the objective, or the constraints functions, are unbounded from below. This leads to a contradiction, since due to Standing Assumptions 2.4 and 2.5 all the iterates lie in a region Ω, where the objective and constraints functions are bounded in norm. Therefore, the result follows. ∎

Finally, we can show convergence of the algorithm of the primal and dual iterates to a KKT triple of (2.1).

THEOREM 4.3. *It holds that the primal and dual iterates in Algorithm* 2 *converge to the KKT triple associated to a critical point* $z^\star$ *of* (2.1)*. That is,*

$$\lim_{i\to\infty} \left\| z^{(i)} - z^\star \right\|_2 = \lim_{i\to\infty} \left\| \lambda^{(i)} - \lambda^\star \right\|_2 = \lim_{i\to\infty} \left\| \nu^{(i)} - \nu^\star \right\|_2 = 0.$$

Proof. The proof follows the line of [22, proofs of Corollary 4.1 and Theorem 4.2]. See Appendix A for the proof details. ∎

4.3. Asymptotic linear convergence. In this section we show that step sizes $t = 1$ are not precluded by the Wolfe conditions in (2.11) when the iterates are sufficiently close to the solution. Therefore local convergence at a linear rate (Theorem 4.1) can be recovered. The following standard SQP assumption is considered in the analysis [22, 38].

Assumption 4.2. For all sufficiently large i, the following holds:

$$\begin{aligned} z^{(i)} + d_{\mathrm{z}}^{(i)} - z^\star &= o\left(\left\| z^{(i)} - z^\star \right\|_2\right), \\ \lambda^{(i)} + d_\lambda^{(i)} - \lambda^\star &= o\left(\left\| \lambda^{(i)} - \lambda^\star \right\|_2\right), \\ \nu^{(i)} + d_\nu^{(i)} - \nu^\star &= o\left(\left\| \nu^{(i)} - \nu^\star \right\|_2\right), \\ \left[d_\lambda^{(i)}; d_\nu^{(i)}\right] &= \mathcal{O}\left(\left\| d_{\mathrm{z}}^{(i)} \right\|_2\right). \end{aligned}$$

□

This assumption implies that

$$\left\| d_{\mathrm{z}}^{(i)} \right\|_2 \sim \left\| z^{(i)} - z^\star \right\|_2, \left\| d_\lambda^{(i)} \right\|_2 \sim \left\| \lambda^{(i)} - \lambda^\star \right\|_2, \left\| d_\nu^{(i)} \right\|_2 \sim \left\| \nu^{(i)} - \nu^\star \right\|_2,$$

where the notation "$\sim$" indicates that the quantities are of similar order as i approaches infinity. Note that this assumption may be restrictive, as it implies a faster convergence (superlinear) than that assessed in Theorem 4.1. However, we observe that this holds true in some practical situation, in particular if the Hessian of the Lagrangian at the solution is a multiple of the identity matrix.

Next, we show that the penalty parameter ρ is bounded.

LEMMA 4.8. *If Assumption* 4.2 *holds, then there exists a finite* $\bar{\rho}$ *such that* $\rho^{(i)} \leq \bar{\rho}$ *for all* $i \in \mathbb{N}$.

Proof. The proof follows the same argument as [22, Lemma 5.1]. Assume that the parameter ρ is unbounded. Then, by Lemma 4.4, the condition in (4.10) must hold over an infinite subsequence of iterations. Thus, using simplified notation,

$$\left\| \begin{bmatrix} g+s \\ h \end{bmatrix} \right\|_2 \geq \frac{1}{4\alpha} \frac{\|d_z\|_2^2}{\left\| \begin{bmatrix} d_\lambda \\ d_\nu \end{bmatrix} \right\|_2},$$

and in turn, by Assumption 4.2, there exists a constant M such that

$$\frac{\left\| \begin{bmatrix} g+s \\ h \end{bmatrix} \right\|_2}{\left\| \begin{bmatrix} d_\lambda \\ d_\nu \end{bmatrix} \right\|_2} \geq \frac{1}{4\alpha} \frac{\|d_z\|_2^2}{\left\| \begin{bmatrix} d_\lambda \\ d_\nu \end{bmatrix} \right\|_2^2} = \frac{M}{4\alpha} > 0$$

for all sufficiently large iterations i, hence the constraints are bounded from below in norm. By Lemma 4.5, the penalty parameter ρ must be bounded over the infinite subsequence of iterations, contradicting the unboundedness assumption. ∎

LEMMA 4.9. *Under Assumption* 4.2*, the condition in* (2.11b) *holds with step size* $t = 1$ *for sufficiently large* i*, i.e.,*

$$\phi(1) - \phi(0) \leq \sigma_1 \phi'(0),$$

where $0 < \sigma_1 < \frac{1}{2}$.

Proof. The proof is similar to [22, Lemma 5.2] and [38, Lemma 4.2]; see Appendix A for the proof details. ∎

LEMMA 4.10. *Under Assumption* 4.2*, the condition in* (2.11a) *holds with step size* $t = 1$ *for sufficiently large* i*, i.e.,*

$$|\phi'(1)| \leq \sigma_2 |\phi'(0)|$$

where $\sigma_2 < \frac{1}{2}$.

Proof. The proof is similar to [22, Lemma 5.3]. See Appendix A for the proof details. ∎

5. Practical implementation. In section 4 we have proven the convergence of the proposed method for general inequality and equality constrained, smooth, optimization problems. On the other hand, it is known that the projected gradient method is inefficient if the feasible set is a general polytope [1].

In the following, we outline two methods for simplifying the computation of the projection. In section 5.1 we transform the general nonlinear problem into an equality constrained problem via squared-slack variables and compute the projection onto the resulting affine subspace in closed form.

In section 5.2, we apply the method to nonlinear MPC problems with box constraints on the input variables and terminal quadratic constraints on the state variables. As in standard gradient method for linear MPC, by writing (condensing) the state variables as an explicit function of the input, the equality constraints are directly embedded into the objective function. Then, the considered constraints are shown to be easy to project for the proposed algorithm by the introduction of the squared-slack variables.

5.1. General optimization problems. The problem in (2.1) can be reformulated as the equality constrained problem

$$\begin{aligned} \min_{(z,y)\in\mathbb{R}^n\times\mathbb{R}^m} \quad & J(z) \\ \text{s.t.} \quad & g(z) + \frac{1}{2}\operatorname{diag}(y)\, y = 0 \\ & h(z) = 0, \end{aligned}$$

hence more generally as

$$\begin{aligned} \min_{v\in\mathbb{R}^{n+m}} \quad & J(v) \\ \text{s.t.} \quad & p(v) = 0 \end{aligned} \tag{5.1}$$

with primal variable $v := [z; y]$ and $p : \mathbb{R}^{n+m} \to \mathbb{R}^{m+p}$ defined as $p([z; y]) := [g(z) + \frac{1}{2}\operatorname{diag}(y)\, y; h(z)]$. Let us define $\mu := [\lambda; \nu]$ as the dual variables associated to the constraints $p(v) = 0$ (μ), $g(z) + \frac{1}{2}\operatorname{diag}(y)y = 0$ (λ), and $h(z) = 0$ (ν), respectively. The equivalence between (2.1) and (5.1) is shown in technical details in Appendix B.

In Algorithm 2, the primal update d_{v} is computed in (3.1) as a function of the current v, i.e.,

$$d_{\mathrm{v}} := \Pi_{\{d_{\mathrm{v}}\in\mathbb{R}^{n+m} \;|\; \nabla p(v)^\top d_{\mathrm{v}} = -p(v)\}} \left(-\alpha \nabla J(v)\right),$$

where

$$p(v) = \begin{bmatrix} g(z) + \frac{1}{2}\operatorname{diag}(y)\, y \\ h(z) \end{bmatrix}, \qquad \nabla p(v) = \begin{bmatrix} \nabla g(z) & \nabla h(z) \\ \operatorname{diag}(y) & 0 \end{bmatrix}.$$

The projection admits a closed form solution. In fact, we can determine the dual variable $\mu_{\mathrm{G}} := [\lambda_{\mathrm{G}}; \nu_{\mathrm{G}}]$ as the solution of the dual problem [4]:

$$\begin{aligned} \mu_{\mathrm{G}} := \begin{bmatrix} \lambda_{\mathrm{G}} \\ \nu_{\mathrm{G}} \end{bmatrix} &= \left(\alpha \nabla p(v)^\top \nabla p(v)\right)^{-1} \left(p(v) - \nabla p(v)^\top \alpha \nabla J(v)\right) \\ &= \begin{bmatrix} \nabla g(z)^\top \nabla g(z) + \operatorname{diag}(y)^2 & \nabla g(z)^\top \nabla h(z) \\ \nabla h(z)^\top \nabla g(z) & \nabla h(z)^\top \nabla h(z) \end{bmatrix}^{-1} \\ &\qquad \cdot \begin{bmatrix} \frac{1}{\alpha} g(z) + \frac{1}{2\alpha}\operatorname{diag}(y) y - \nabla g(z)^\top \nabla J(z) \\ \frac{1}{\alpha} h(z) - \nabla h(z)^\top \nabla J(z) \end{bmatrix}. \end{aligned} \tag{5.2}$$

Then, the primal solution is given by

$$\begin{aligned} d_{\mathrm{v}} = \begin{bmatrix} d_{\mathrm{z}} \\ d_{\mathrm{y}} \end{bmatrix} &= -\alpha \nabla J(v) - \alpha \nabla p(v) \mu_{\mathrm{G}} \\ &= \begin{bmatrix} -\alpha \nabla J(z) \\ 0 \end{bmatrix} - \alpha \begin{bmatrix} \nabla g(z) & \nabla h(z) \\ \operatorname{diag}(y) & 0 \end{bmatrix} \begin{bmatrix} \lambda_{\mathrm{G}} \\ \nu_{\mathrm{G}} \end{bmatrix} \\ &= \begin{bmatrix} -\alpha \nabla J(z) - \alpha \nabla g(z) \lambda_{\mathrm{G}} - \alpha \nabla h(z) \nu_{\mathrm{G}} \\ -\alpha \operatorname{diag}(y) \lambda_{\mathrm{G}} \end{bmatrix}, \end{aligned} \tag{5.3}$$

and dual increments d_μ from (3.5) are $d_\mu := \left[\begin{smallmatrix} d_\lambda \\ d_\nu \end{smallmatrix}\right] = \mu_{\mathrm{G}} - \mu$.

By Assumption 4.1, the matrix $\nabla p(v)$ can be proven to be full rank; therefore, the matrix $\left(\alpha \nabla p(v)^\top \nabla p(v)\right)$ is invertible. For this to hold, the squared-slack variable initialization, $y^{(0)}$, has to be set different from zero; otherwise $y^{(i)} = 0$ for all i. Also note that by (5.2), the squared-slack variables do not increase the dimension of the matrix, thus the complexity of the matrix inversion does not increase.

The matrix inversion in (5.2) is commonly obtained via Cholesky factorization, which for a general dense matrix has a computational complexity of $\mathcal{O}\left((m+p)^3\right)$ floating point operations (FLOPS). Therefore, for the factorization to be efficient, the sparsity of the gradients ∇g and ∇h should be exploited. In the following section, we consider a general nonlinear MPC problem for which the matrix inversion can be computed symbolically, thus avoiding the Cholesky factorization.

5.2. MPC problems. Sparsity patterns naturally arise in MPC problems, due to the causality of the dynamics and the structure of the constraints.

Let us consider a typical nonlinear MPC problem with box input constraints and a quadratic terminal state constraint,

$$
\begin{aligned}
\min_{(x_{k+1},u_k)_{k=0}^{N-1}} \quad & \sum_{k=0}^{N-1} \left\{ \frac{1}{2} x_k^\top Q x_k + \frac{1}{2} u_k^\top R u_k \right\} + \frac{1}{2} x_N^\top P x_N \\
\text{s.t.} \quad & x_{k+1} = f\left(x_k, u_k\right) \quad \forall k \in \mathbb{Z}[0, N-1], \\
& u_k \in [a_k, b_k] \qquad\quad \forall k \in \mathbb{Z}[0, N-1], \\
& \tfrac{1}{2} x_N^\top P x_N \leq c,
\end{aligned}
\tag{5.4}
$$

where the index k spans the predicted state x_{k+1} and input u_k in the horizon N. The bounds satisfy $a_k < b_k \in \mathbb{R}^{n_\mathrm{u}}$ componentwise, $c > 0$, and $Q, P, R \succcurlyeq 0$. The discrete-time dynamics, f, are nonlinear; hence the program in (5.4) is in general nonconvex.

For ease of notation, let us first define the vectors $\mathbf{u} := [u_0; \ldots; u_{N-1}]$ for the control input sequence, with bounds $\mathbf{a} := [a_0; \ldots; a_{N-1}]$ and $\mathbf{b} := [b_0; \ldots; b_{N-1}]$ and the corresponding state evolution $\mathbf{x} := [x_1; \ldots; x_N]$, and stack the state and input cost matrices $\mathcal{Q} = \mathrm{blockdiag}\left(Q, \ldots, Q, P\right)$ and $\mathcal{R} = \mathrm{blockdiag}\left(R, \ldots, R\right)$. We recast the dynamics in a compact form as $\mathbf{x} = \psi(\mathbf{u})$, where for a fixed initial state x_0, the function $\psi : \mathbb{R}^{N n_\mathrm{u}} \to \mathbb{R}^{N n_\mathrm{x}}$ maps the sequence of inputs $\mathbf{u}$ to the predicted sequence of states $\mathbf{x}$ according to the nonlinear dynamics $x_{k+1} = f(x_k, u_k)$. Thus, by including the nonlinear dynamics within the objective and by adding the nonlinear slacks $\mathbf{y}_\mathrm{a}, \mathbf{y}_\mathrm{b} \in \mathbb{R}^{N n_\mathrm{u}}$ and $y_\mathrm{c} \in \mathbb{R}$ as in (5.1), the MPC problem in (5.4) reads as

$$
\begin{aligned}
\min_{\mathbf{u}, \mathbf{y}_\mathrm{a}, \mathbf{y}_\mathrm{b}, y_\mathrm{c}} \quad & \tfrac{1}{2} \psi\left(\mathbf{u}\right)^\top \mathcal{Q} \psi\left(\mathbf{u}\right) + \tfrac{1}{2} \mathbf{u}^\top \mathcal{R} \mathbf{u} =: J(\mathbf{u}) \\
\text{s.t.} \quad & -\mathbf{u} + \boldsymbol{a} + \frac{1}{2} \operatorname{diag}(\mathbf{y}_\mathrm{a}) \mathbf{y}_\mathrm{a} = 0, \\
& \mathbf{u} - \boldsymbol{b} + \frac{1}{2} \operatorname{diag}(\mathbf{y}_\mathrm{b}) \mathbf{y}_\mathrm{b} = 0, \\
& \tfrac{1}{2} \psi_N(\mathbf{u})^\top P \psi_N(\mathbf{u}) - c + \tfrac{1}{2} y_\mathrm{c}^2 = 0.
\end{aligned}
\tag{5.5}
$$

This formulation, albeit unusual compared to other approaches when solving nonlinear MPC problems [13, 45], leads to computational advantages for the proposed Algorithm 2.

The primal and dual variable updates of Algorithm 2 are determined as explained in subsection 5.1. Note that the matrix inversion in (5.2) can be computed analytically

offline. In fact, since the gradient of the constraint is $\nabla g(\mathbf{u}) = [\ -I \mid I \mid q\ ]$, where $q \in \mathbb{R}^{Nn_{\mathrm{u}}}$ is the gradient of the terminal constraints with respect to $\mathbf{u}$, the matrix is inverted as follows:

$$\begin{aligned}
&\left(\nabla g(\mathbf{u})^\top \nabla g(\mathbf{u}) + \operatorname{diag}([\mathbf{y}_{\mathrm{a}}; \mathbf{y}_{\mathrm{b}}; y_{\mathrm{c}}])^2\right)^{-1} \\
&= \begin{bmatrix} I + \operatorname{diag}(\mathbf{y}_{\mathrm{a}})^2 & -I & -q \\ -I & I + \operatorname{diag}(\mathbf{y}_{\mathrm{b}})^2 & q \\ -q^\top & q^\top & q^\top q + y_{\mathrm{c}}^2 \end{bmatrix}^{-1} \\
&= \begin{bmatrix} D + B + r(Bq)(Bq)^\top & D - r(Bq)(Aq)^\top & rBq \\ D - r(Aq)(Bq)^\top & D + A + r(Aq)(Aq)^\top & -rAq \\ r(Bq)^\top & -r(Aq)^\top & r \end{bmatrix},
\end{aligned}$$

with

$$D = \operatorname{diag}\left(\frac{1}{y_{\mathrm{a},j}^2 + y_{\mathrm{b},j}^2 + y_{\mathrm{a},j}^2 y_{\mathrm{b},j}^2}\right),\ A = \operatorname{diag}\left(\frac{y_{\mathrm{a},j}^2}{y_{\mathrm{a},j}^2 + y_{\mathrm{b},j}^2 + y_{\mathrm{a},j}^2 y_{\mathrm{b},j}^2}\right)$$

$$B = \operatorname{diag}\left(\frac{y_{\mathrm{b},j}^2}{y_{\mathrm{a},j}^2 + y_{\mathrm{b},j}^2 + y_{\mathrm{a},j}^2 y_{\mathrm{b},j}^2}\right),\ r = \left(\sum_{j=1}^{Nn_{\mathrm{u}}} \frac{y_{\mathrm{a},j}^2 y_{\mathrm{b},j}^2}{y_{\mathrm{a},j}^2 + y_{\mathrm{b},j}^2 + y_{\mathrm{a},j}^2 y_{\mathrm{b},j}^2} q_j^2 + y_{\mathrm{c}}^2\right)^{-1}.$$

Under Assumption 4.1, we can guarantee that the denominators in D, A, B are always nonzero and r is finite. Because of the diagonal structure of A and B, the vectors Aq and Bq are cheap to compute, and this allows one to compute the matrix multiplication in (5.2) in only $\mathcal{O}(Nn_{\mathrm{u}})$ FLOPS. Moreover, since the terminal constraint in (5.4) has the same structure of the terminal cost in the objective function, the computation of q is inexpensive when performed together with the computation of $\nabla J(\mathbf{u})$.

The primal variable updates $\mathbf{d}_{\mathrm{u}}$ and the slack updates $\mathbf{d}_{\mathrm{y,a}}$, $\mathbf{d}_{\mathrm{y,b}}$, and $d_{\mathrm{y,c}}$ follow from (5.3). The computation of the gradient of the objective function can also be done efficiently by exploiting the causality of the nonlinear dynamics, f. Let us define the following auxiliary functional $J_{\mathrm{a}}(\mathbf{u})$:

$$J_{\mathrm{a}}(\mathbf{u}) := \sum_{k=0}^{N-1} \left\{ \frac{1}{2} x_k^\top Q x_k + \frac{1}{2} u_k^\top R u_k + \theta_{k+1}^\top \left(f(x_k, u_k) - x_{k+1}\right) \right\} + \frac{1}{2} x_N^\top P x_N,$$

with $x_k = \psi_k(\mathbf{u})$ and multipliers $\theta_k\ \forall k$. Note that, regardless the value of the multipliers, $J(\mathbf{u}) = J_{\mathrm{a}}(\mathbf{u})$ since $\psi_{k+1}(\mathbf{u}) = f(\psi_k(\mathbf{u}), u_k)$. Differentiating $J_{\mathrm{a}}(\mathbf{u})$ with respect to $\mathbf{u}$ yields

(5.6)

$$\begin{aligned}
\nabla J_{\mathrm{a}}(\mathbf{u}) &= \sum_{k=0}^{N-1} \left\{ \nabla \psi_k(\mathbf{u}) Q x_k + \left(\nabla \psi_k(\mathbf{u}) F_k^\top + \frac{\partial u_k}{\partial \mathbf{u}} G_k^\top - \nabla \psi_{k+1}(\mathbf{u}) \right) \theta_{k+1} \right\} + \mathcal{R}\mathbf{u} \\
&\quad + \nabla \psi_N(\mathbf{u}) P x_N \\
&= \sum_{k=1}^{N-1} \nabla \psi_k(\mathbf{u}) \left(Q x_k - \theta_k + F_k^\top \theta_{k+1} \right) + \sum_{k=0}^{N-1} \left(\frac{\partial u_k}{\partial \mathbf{u}} G_k^\top \right) \theta_{k+1} + \mathcal{R}\mathbf{u} \\
&\quad + \nabla \psi_N(\mathbf{u}) \left(P x_N - \theta_N \right) + \nabla \psi_0(\mathbf{u}) \left(Q x_0 + F_0^\top \theta_1 \right),
\end{aligned}$$

where the partial derivative $\frac{\partial u_k}{\partial \mathbf{u}}$ results in a block matrix of zeros for $j \neq k$ and the identity otherwise, and the matrices $\nabla\psi_k(\mathbf{u})$ contain the standard linearization matrices of the nonlinear dynamics

$$F_k := \tfrac{\partial f}{\partial x}(x_k, u_k), \qquad G_k := \tfrac{\partial f}{\partial u}(x_k, u_k),$$

and $\nabla\psi_0(\mathbf{u}) = 0$ because of the causality of the dynamics. By choosing

$$\theta_N = Px_N, \qquad \theta_k = F_k^\top \theta_{k+1} + Qx_k \ \forall k \in \{1, \dots, N-1\},$$

then

$$\nabla J(\mathbf{u}) = \nabla J_{\mathrm{a}}(\mathbf{u}) = \mathcal{R}\mathbf{u} + \sum_{k=0}^{N-1} \left(\frac{\partial u_k}{\partial \mathbf{u}} G_k^\top \right) \theta_{k+1},$$

which can be efficiently determined by backward substitution. For diagonal cost matrices Q, R and full P, the number of FLOPS required to compute the last vector $G_{N-1}^\top \theta_N = G_{N-1}^\top P x_N$ is upper bounded by $2(n_{\mathrm{x}}^2 + n_{\mathrm{x}} n_{\mathrm{u}})$ FLOPS. By not recomputing the term θ_N, the computation of θ_{N-1} requires a number of FLOPS upper bounded by $2(n_{\mathrm{x}}^2 + n_{\mathrm{x}} n_{\mathrm{u}} + n_{\mathrm{x}})$ FLOPS. Since every multiplier θ_i can be computed from the successive one and by also considering the term $\mathcal{R}\mathbf{u}$, the computational complexity of computing the gradient step is $\mathcal{O}\left(N(n_{\mathrm{x}}^2 + n_{\mathrm{x}} n_{\mathrm{u}})\right)$. Since the subsequent steps of Algorithm 2 to determine $\rho^{(i)}$ from (2.10) and $t^{(i)}$ from (2.11) have lower complexity, including the computation of the merit function $\phi(t)$ and its derivative $\phi'(t)$, this is the resulting complexity of the algorithm. Note that this complexity is comparable to standard gradient method for linear MPC [40], $\mathcal{O}\left((N n_{\mathrm{u}})^2\right)$, and since the Hessian H never needs to be computed, the complexity depends linearly (instead of quadratically) on the prediction horizon, N. Further, the complexity of the SQP method depends on the complexity of the QP solver. The active set method for the presented MPC problem requires $\mathcal{O}\left((N n_{\mathrm{u}})^2\right)$ FLOPS [17], while an interior point method exploiting sparsity of the MPC requires $\mathcal{O}\left(N(n_{\mathrm{x}}^3 + n_{\mathrm{x}}^2 n_{\mathrm{u}})\right)$ FLOPS [16].

The terminal constraint $\frac{1}{2} x_N^\top P x_N$ is a level set of a (local) control Lyapunov function. To ensure recursive feasibility, the control Lyapunov function must be defined such that there exists a local (constraint admissible) control law such that the level set is positively invariant. Several classes of control Lyapunov functions have been considered in the MPC literature, e.g., quadratic in [6]. Such a constraint need not necessarily act on the terminal state: an alternative formulation such as contractive MPC, employing quadratic Lyapunov conditions in time steps other than the last one [9], can be analogously considered. On the other hand, similar to the projected gradient method for MPC, stage-wise state constraints cannot be efficiently included in the formulation. In fact, because of the nonlinear dynamics, the state constraints will turn into nonlinear constraints of all the inputs up to the considered stages. This makes the matrix inversion in (5.2) no longer analytically computable.

The proposed method applies to a more general class of MPC problems than that in (5.5). We refer to the open-source software FalcOpt [47] for the automatic generation of efficient C code interfaced with MATLAB via generated MEX interfaces. FalcOpt exploits the sparsity of the problem and can be deployed on embedded devices. Note that the matrix inversion is performed analytically, hence consists of algebraic instructions only.

Finally, we emphasize that, if applied to continuous-time systems, the proposed method is best suited for nonstiff systems that can be approximated via explicit methods with a sufficiently small sampling time. For stiff problems that require a refined integration routine, we refer to [39, 13].

6. Numerical example: Nonlinear MPC of an inverted pendulum. We consider an inverted pendulum as a rod of length $l = 0.3$ m with mass $m = 0.2$ kg concentrated at the tip and no friction acting on the cart and swing. The mass of the cart is $M = 0.5$ kg, and the gravitational acceleration is $g = 10$ m/s^2. The states x_1 and x_2 are respectively the cart position and velocity, and x_3 and x_4 are the pendulum angle and angular velocity. The input u is the applied force on the cart, and it is subject to box constraints. We discretize the continuous-time dynamics

$$\begin{aligned}
\dot{x}_1 &= x_2,\\
\dot{x}_2 &= \frac{mg\sin x_3\cos x_3 - mlx_4^2\sin x_3 + u}{M + m\sin^2(x_3)},\\
\dot{x}_3 &= x_4,\\
\dot{x}_4 &= \frac{g}{l}\sin x_3 + \frac{mg\sin x_3\cos^2 x_3 + u\cos x_3 - mlx_4^2\sin x_3\cos x_3}{l\left(M + m\sin^2 x_3\right)},
\end{aligned}$$

with the explicit Euler method with sampling time $T_s = 0.1$ s, and hence obtain an MPC problem of the form (5.4), with model Jacobians computed symbolically. The desired closed-loop performance is achieved with a prediction horizon of $N = 8$ and the cost matrices are $Q = \mathrm{diag}([10; 0.1; 100; 0.1])$ and $R = 1$. The terminal cost matrix, P, is determined by the algebraic Riccati equation using linearized dynamics around the desired equilibrium. The constant c in the terminal constraint is set to $c = 1.5$, so that such constraint is active at the first iteration.

The system has two sets of unforced equilibria, the unstable ones $[p; 0; 2k\pi; 0]$ and the stable ones $x = [p; 0; \pi + 2k\pi; 0]$, with $p \in \mathbb{R}$ and $k \in \mathbb{Z}$. Physically, the former correspond to the pendulum in the upright position, while the latter correspond to the pendulum in the natural upside-down configuration. The goal of the controller is to stabilize the system around the origin, that is, to the unstable equilibrium, starting from the stable one at $x_0 := [0; 0; \pi; 0]$.

For many nonlinear problems, the RTI yields a sufficiently good approximation of the nonlinear solution [13]. This consists of solving only one QP in (2.3) at every time step. For the specific problem considered here, the RTI effectively stabilizes the pendulum, but its closed-loop cost is much larger than that obtained by a full nonlinear solution, as shown in Figure 6.1 and Table 6.1, and it has the advantage of requiring low computational times [15]. Further, the linearization of the terminal constraint may result to be unfeasible, even though the terminal constraint is feasible for the original nonlinear problem. Thus, a reformulation with soft constraints is required.

The nonlinear solution is determined via the proposed method and with the SQP approach in Algorithm 1, and the computational times obtained are in Table 6.1. Specifically, the solver SNOPT is used, running in Fortran and interfaced via TOMLAB to MATLAB, and the computational time is measured internally by the solver [21, 25]. We do not observe a significant difference in the computational time with direct calls to SNOPT. The proposed algorithm has been coded in C, and the time indicated comprises both the preparation of the problem, i.e., building the Jacobian matrices, and solving the optimization problem. The calculations have been performed on a commercial off-the-shelf Windows PC with processor Intel Core i7-3740QM 2.70Ghz. A pure sequential C code was used, compiled by Intel Composer 2016 with the optimization flag `/Ox` enabled.

In the implementation we have noticed that the solver SNOPT requires an overly long time to solve the first optimization problem. As the level of optimality does

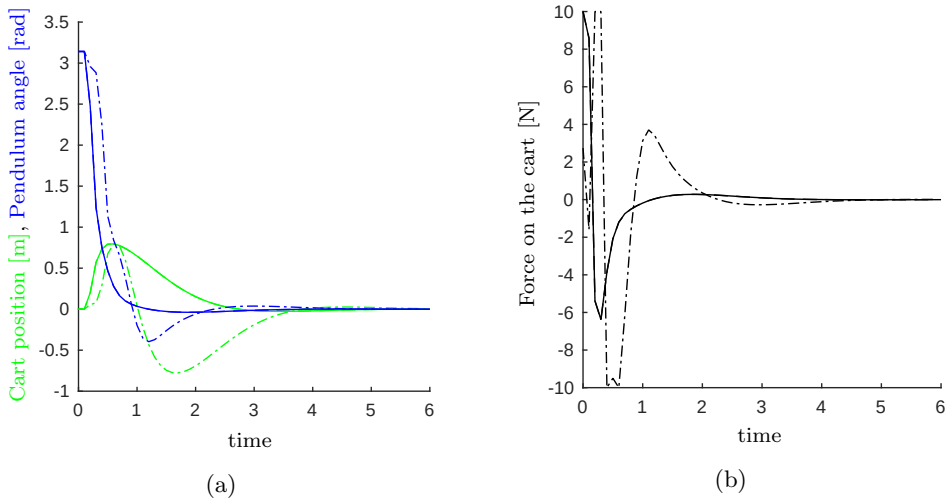

FIG. 6.1. *Swing-up simulation of an inverted pendulum:* (a) *state dynamics, where the angle is in blue and the cart position in light green;* (b) *input: force applied on the cart. The solid lines show the proposed gradient algorithm solution, and they overlap with those of the SQP solution. The dash-dot lines show the RTI solution.*

TABLE 6.1
Computational times.

Method	Avg. time (ms)	Best (ms)	Worst (ms)	Cost
Proposed Algorithm	2.39	0.06	4.15	318
SQP (SNOPT)	9.66*	4.00	370*	318
RTI (FORCES Pro)	0.13	0.10	0.17	527

* The first MPC optimization in SNOPT exceeds the maximum number of major iterations and is not considered in the average and worst case times.

not reach the desired tolerance, the solver exceeded the maximum number of major iterations (150) in the first MPC instance, thus requiring more than 1 second for the solution of the problem. The average and worst case times reported in Table 6.1 for the SQP solver do not account for this first MPC instance. The problem does not occur in the proposed algorithm with the same initialization.

The computational times obtained make the proposed algorithm competitive with the solver SNOPT. The average time is 80% faster than the SQP, while the best and worst cases are significantly better. Warm starting makes the algorithm particularly effective when it is initialized close to the optimal solution, as only a few gradient steps are required for convergence. The proposed algorithm is slower than the RTI by around one order of magnitude, as the RTI requires only the solution to one QP at each time step.

Additional benefits in terms of computational speed can be obtained by accelerating the proposed algorithm via a specific heuristic that modifies (3.1). More details and computational times for a similar example are given in [49].

We remark that the computational benefit yielded by the proposed algorithm is possible only thanks to the sparsity of the problem, which allows us to compute the matrix inversion in (5.2) analytically. If this were not the case, the computational benefit of the proposed method would be lost.

6.1. Computational considerations. We consider standard convergence conditions to terminate the algorithm. Instead of considering an exact null $d_z^{(i)}$ as in Algorithm 2 for terminating the optimization routine, we relax the condition to a small enough value. Leveraging (3.4), if feasibility of the constraint is achieved, by bounding appropriately $\|d_z\|_2$ we can equivalently impose that the iterate $z^{(i)}$ is a KKT point for the nonlinear problem with a desired tolerance. Another condition involves the derivative of the augmented Lagrangian function, $\phi'(0)$. By (3.6), setting a negative lower bound to the derivative $\phi'(0)$ corresponds to checking that the iterate $\|d_z\|_2$ is small. As usual in gradient methods, a limit on the maximum number of iterations is also set. We discuss this last criterion in the considerations listed next.

The theoretical results presented in section 4 show that guarantees on the convergence speed can be derived only when the iterate is close to the solution and the correct active set is determined, achieving linear rate. In fact, by Theorem 4.1, linear convergence is achieved via a specific tuning of α, based the maximum eigenvalue of Hessian of the Lagrangian at the optimum. This would allow for $t = 1$ as shown in subsection 4.3. On the other hand, such a value is unknown a priori, and an approximation based on the Hessian of the current iterate can be overly expensive to compute. From our numerical experience, we recommend instead to set an α (not necessarily linked to the problem Lagrangian) and let the line-search variable t become smaller than 1 in the line search. In fact, this approach achieves a comparable convergence speed to the linear rate obtained with the specific α of subsection 4.3 and $t = 1$.

From our numerical experience we also observe that the convergence rate might be faster than linear when the solution is far from optimality, albeit theoretically the convergence rate is unknown, and then slow down to linear once it is close to the optimal point. Figure 6.2 shows, as an example, the KKT optimality, computed via the iterate norm d_z, as a function of the iteration number for a particular MPC instance. Analogous to standard convex optimization theory, the linear rate factor depends on second-order information, and, in our case, a large condition number of the Hessian of the Lagrangian at the solution can yield a slow linear convergence in practice [1]. This means that most of the iterates might be employed to reach the

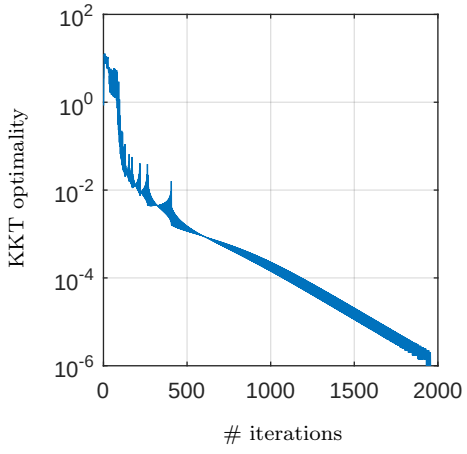

FIG. 6.2. *Level of optimality (derivative of the Lagrangian) as a function of the iteration number for a particular MPC instance. The desired tolerance is set to* 1e−6.

desired tolerance. In the particular example considered here, we set as a limit 3000 iterations. In our numerical experience, mildly nonlinear problems, e.g., due to polynomial system dynamics, result in a significantly smaller number of total iterations required for convergence; see [46, sections 7–9] for other applications.

Combining the method with standard SQP would be a convenient solution for a general purpose solver, since using second-order information once the correct active set is determined speeds up the convergence rate to superlinear.

Appendix A. Proofs of section 4.

Proof of Theorem 4.3. The convergence of the primal variable $z^{(i)}$ follows in a straightforward manner from Theorem 4.2 and Lemma 4.2. It also follows from Theorem 4.2 and Lemma 4.2 that if $\|d_{\mathrm{z}}^{(i)}\|_2 = 0$, then, by (2.2), (2.8), and (3.4), $\lambda_{\mathrm{G}}^{(i)} = \lambda^\star$ and $\nu_{\mathrm{G}}^{(i)} = \nu^\star$, and the algorithm terminates. Thus, we will assume henceforth that $\|d_{\mathrm{z}}^{(i)}\|_2 \neq 0$ for all $i \in \mathbb{N}$. By defining, for all $k \leq i \in \mathbb{N}$,

$$\gamma_{k,i} := \begin{cases} \tilde{t}^{(i)} & \text{if } k = i, \\ \tilde{t}^{(k)} \prod_{m=k+1}^{i} \left(1 - \tilde{t}^{(m)}\right) & \text{otherwise,} \end{cases}$$

with $\tilde{t}^{(0)} := 1$ and $\tilde{t}^{(k)} = t^{(k)}$, the definition in (4.8) implies that

$$\lambda^{(i+1)} = \sum_{k=0}^{i} \gamma_{k,i} \lambda_{\mathrm{G}}^{(k)} \tag{A.1}$$

for all $i \geq 0$, because of the initial condition $\lambda^{(0)} = \lambda_{\mathrm{G}}^{(0)}$. Then, by Lemma 4.7 we have

$$0 < \bar{t} \leq \tilde{t}^{(i)} \leq 1 \quad \forall i \in \mathbb{N}, \tag{A.2a}$$

$$\sum_{k=0}^{i} \gamma_{k,i} = 1, \tag{A.2b}$$

$$\gamma_{k,i} \leq (1 - \bar{t})^{i-k} \quad \forall k < i. \tag{A.2c}$$

Since $z^{(i)} \to z^\star$, the iterates will reach a neighborhood of $z^\star$ where the problem in (3.3) identifies the correct active set (Lemma 4.2) and the active constraint will have full column rank. Assume that the property holds for $i > \tilde{i}$. From (3.4), the definition of $\lambda_{\mathrm{G}}^{(i)}$, and Standing Assumption 2.4, for $i \geq \tilde{i}$ there exists $M > 0$ such that

$$\lambda_{\mathrm{G}}^{(i)} = \lambda^\star + M^{(i)} d^{(i)} v^{(i)}, \tag{A.3}$$

with $|M^{(i)}| \leq M$, $d^{(i)} = \max\{||d_{\mathrm{z}}^{(i)}||_2, \left\|z^\star - z^{(i)}\right\|_2\}$ and $\left\|v^{(i)}\right\|_2 = 1$. For any given $\varepsilon > 0$, Theorem 4.2 implies that i_1 can be chosen such that for all $i \geq i_1$, we have

$$|M^{(i)} d^{(i)}| \leq \varepsilon/2. \tag{A.4}$$

Then, we define the iteration index i_2 such that for all $i \geq i_2$, we have

$$(1 - \bar{t})^i \leq \frac{\varepsilon}{2(i+1)(1 + \hat{\lambda}_{\mathrm{G}} + \|\lambda^\star\|_2)}, \tag{A.5}$$

with $\hat{\lambda}_{\mathrm{G}}$ being an upper bound to $\|\lambda_{\mathrm{G}}^{(i)}\|_2$ for all i. Now let $\tilde{i} := \max\{i_1, i_2\}$. Then, from (A.1) and (A.3), for all $i \geq \tilde{i}$ we have

$$\lambda^{(i+1)} = \sum_{k=0}^{\tilde{i}} \gamma_{k,i} \lambda_{\mathrm{G}}^{(k)} + \sum_{k=\tilde{i}+1}^{i} \gamma_{k,i} \left(\lambda^\star + M^{(k)} d^{(k)} v^{(k)} \right).$$

Hence it follows from (A.2b) that

$$\lambda^{(i+1)} - \lambda^\star = \sum_{k=0}^{\tilde{i}} \gamma_{k,i} (\lambda_{\mathrm{G}}^{(k)} - \lambda^\star) + \sum_{k=\tilde{i}+1}^{i} \gamma_{k,i} M^{(k)} d^{(k)} v^{(k)}.$$

Since the dual variable $\lambda_{\mathrm{G}}^{(k)}$ and $v^{(k)}$ are bounded in norm, it follows that

$$\left\| \lambda^{(i+1)} - \lambda^\star \right\|_2 \leq (\hat{\lambda}_{\mathrm{G}} + \|\lambda^\star\|_2) \sum_{k=0}^{\tilde{i}} \gamma_{k,i} + \sum_{k=\tilde{i}+1}^{i} \gamma_{k,i} |M^{(k)} d^{(k)}|. \tag{A.6}$$

For all iterations $i \geq 2\tilde{i}$, it follows from (A.2a) and (A.2c) that

$$\sum_{k=0}^{\tilde{i}} \gamma_{k,i} \leq \sum_{k=0}^{\tilde{i}} (1-\bar{t})^{i-k} \leq \sum_{k=0}^{\tilde{i}} (1-\bar{t})^{2\tilde{i}-k} \leq (\tilde{i}+1)(1-\bar{t})^{\tilde{i}}.$$

By using (A.5), we derive the bound $(\hat{\lambda}_{\mathrm{G}} + \|\lambda^\star\|_2) \sum_{k=0}^{\tilde{i}} \gamma_{k,i} \leq \frac{1}{2}\varepsilon$ on the first term on the right-hand side of (A.6), and to bound the second term in (A.6), we use (A.2b) and (A.4):

$$\sum_{k=\tilde{i}+1}^{i} \gamma_{k,i} |M^{(k)} d^{(k)}| \leq \frac{1}{2}\varepsilon \sum_{k=\tilde{i}+1}^{i} \gamma_{k,i} \leq \frac{1}{2}\varepsilon. \tag{A.7}$$

Finally, by combining (A.6) and (A.7), we obtain that for all ε, there exists $\tilde{i}$ such that

$$\left\| \lambda^{(i)} - \lambda^\star \right\|_2 \leq \varepsilon \ \forall \ i \geq 2\tilde{i} + 1,$$

which implies that $\lim_{i\to\infty} \left\| \lambda^{(i)} - \lambda^\star \right\|_2 = 0$. The convergence of the dual variables for the equality constraint is analogous. ∎

Proof of Lemma 4.9. By continuity of the second derivative we can derive the relation between the objective function evaluations,

(A.8)

$$\begin{aligned} J(z + d_{\mathrm{z}}) &= J(z) + \frac{1}{2}\nabla J(z)^\top d_{\mathrm{z}} + \frac{1}{2}\nabla J(z)^\top d_{\mathrm{z}} + \frac{1}{2} d_{\mathrm{z}}^\top \nabla^2 J(z) d_{\mathrm{z}} + o\left(\|d_{\mathrm{z}}\|_2^2\right) \\ &= J(z) + \frac{1}{2}\nabla J(z)^\top d_{\mathrm{z}} + \frac{1}{2}\left(\nabla J(z + d_{\mathrm{z}}) - \nabla^2 J(z) d_{\mathrm{z}}\right)^\top d_{\mathrm{z}} + \frac{1}{2} d_{\mathrm{z}}^\top \nabla^2 J(z) d_{\mathrm{z}} \\ &\quad + o\left(\|d_{\mathrm{z}}\|_2^2\right) \\ &= J(z) + \frac{1}{2}(\nabla J(z) + \nabla J(z + d_{\mathrm{z}}))^\top d_{\mathrm{z}} + o\left(\|d_{\mathrm{z}}\|_2^2\right), \end{aligned}$$

and analogously for the constraint functions g and h:

$$g(z+d_z) = g(z) + \frac{1}{2}(\nabla g(z) + \nabla g(z+d_z))^\top d_z + o\left(\|d_z\|_2^2\right), \tag{A.9}$$

$$h(z+d_z) = h(z) + \frac{1}{2}(\nabla h(z) + \nabla h(z+d_z))^\top d_z + o\left(\|d_z\|_2^2\right). \tag{A.10}$$

By Assumption 4.2 we have that

$$\begin{aligned} \nabla J(z+d_z) &= \nabla J(z^\star) + \nabla^2 J(z^\star)\,(z+d_z-z^\star) + o\left(\|z+d_z-z^\star\|_2\right) \\ &= \nabla J(z^\star) + o\left(\|d_z\|_2\right), \\ \nabla g(z+d_z) &= \nabla g(z^\star) + o\left(\|d_z\|_2\right), \\ \nabla h(z+d_z) &= \nabla h(z^\star) + o\left(\|d_z\|_2\right), \end{aligned} \tag{A.11}$$

hence substituting into (A.8), (A.9), (A.10) we obtain

$$\begin{aligned} J(z+d_z) &= J(z) + \frac{1}{2}(\nabla J(z) + \nabla J(z^\star))^\top d_z + o\left(\|d_z\|_2^2\right), \\ g(z+d_z) &= g(z) + \frac{1}{2}(\nabla g(z) + \nabla g(z^\star))^\top d_z + o\left(\|d_z\|_2^2\right), \\ h(z+d_z) &= h(z) + \frac{1}{2}(\nabla h(z) + \nabla h(z^\star))^\top d_z + o\left(\|d_z\|_2^2\right). \end{aligned} \tag{A.12}$$

Let us use the simplified notation and denote $J(z)$ as J, $J(z+d_z)$ as J^+, and $J(z^\star)$ as $J^\star$ (and similarly for g and h and their gradients). By (2.8), (2.9), and (3.5), $\phi(0)$ and $\phi(1)$ are

$$\begin{aligned} \phi(0) &= J + \lambda^\top(g+s) + \nu^\top h + \frac{1}{2}\rho(g+s)^\top(g+s) + \frac{1}{2}\rho h^\top h\,, \\ \phi(1) &= J^+ + \lambda_{\mathrm{G}}^\top\left(g^+ - g - \nabla g^\top d_z\right) + \nu_{\mathrm{G}}^\top h^+ \\ &\quad + \frac{1}{2}\rho\left(g^+ - g - \nabla g^\top d_z\right)^\top\left(g^+ - g - \nabla g^\top d_z\right) + \frac{1}{2}\rho h^{+\top} h^+ \\ &= J^+ + \lambda_{\mathrm{G}}^\top\left(g^+ - g - \nabla g^\top d_z\right) + \nu_{\mathrm{G}}^\top h^+ + o\left(\|d_z\|_2^2\right), \end{aligned}$$

where the last step follows by (3.1) and Taylor's expansions:

$$\begin{aligned} g^+ - g - \nabla g^\top d_z &= o\left(\|d_z\|_2\right), \\ h^+ &= \underbrace{h + \nabla h^\top d_z}_{=0\,(3.1)} + o\left(\|d_z\|_2\right) = o\left(\|d_z\|_2\right). \end{aligned}$$

By (A.12), $\phi(1)$ can be written as

$$\begin{aligned} \phi(1) = J + \frac{1}{2}(\nabla J + \nabla J^\star)^\top d_z + \frac{1}{2}\lambda_{\mathrm{G}}^\top\left(\nabla g^\star - \nabla g\right) d_z \\ + \nu_{\mathrm{G}}^\top\left(h + \frac{1}{2}(\nabla h + \nabla h^\star))^\top d_z\right) + o\left(\|d_z\|_2^2\right). \end{aligned}$$

This implies

$$\begin{aligned} \phi(1) - \phi(0) = \frac{1}{2}(\nabla J + \nabla J^\star)^\top d_z + \frac{1}{2}\lambda_{\mathrm{G}}^\top\left(\nabla g^\star - \nabla g\right)^\top d_z - \lambda^\top(g+s) - \nu^\top h \\ + \nu_{\mathrm{G}}^\top\left(h + \frac{1}{2}(\nabla h + \nabla h^\star)^\top d_z\right) - \frac{1}{2}\rho\,\|g+s\|_2^2 - \frac{1}{2}\rho\,\|h\|_2^2 + o\left(\|d_z\|_2^2\right). \end{aligned} \tag{A.13}$$

By Lemma 4.4, the derivative $\phi'(0)$ can be recast as

$$\phi'(0) = d_{\mathrm{z}}^{\top}\nabla J - 2(g+s)^{\top}\lambda - (\nabla g^{\top}d_{\mathrm{z}} + d_{\mathrm{s}})^{\top}\lambda_{\mathrm{G}} - \rho\left\|g+s\right\|_2^2 - 2h^{\top}\nu + h^{\top}\nu_{\mathrm{G}} - \rho\left\|h\right\|_2^2,$$

which when substituted in (A.13) yields

$$\phi(1) - \phi(0) = \frac{1}{2}\phi'(0) + \frac{1}{2}\left(\nabla J^{\star} + \nabla g^{\star}\lambda_{\mathrm{G}} + \nabla h^{\star}\nu_{\mathrm{G}}\right)^{\top} d_{\mathrm{z}} + \frac{1}{2}\lambda_{\mathrm{G}}^{\top}d_{\mathrm{s}} + o\left(\left\|d_{\mathrm{z}}\right\|_2^2\right).$$

Next we note that Assumption 4.2 implies specific conditions on the convergence of the variables λ_{G} and ν_{G}. That is,

$$\begin{aligned}\lambda_{\mathrm{G}} - \lambda^{\star} = d_{\lambda} + \lambda - \lambda^{\star} = o\left(\left\|\lambda - \lambda^{\star}\right\|_2\right) = o\left(\left\|d_{\lambda}\right\|_2\right) = o\left(\left\|d_{\mathrm{z}}\right\|_2\right),\\ \nu_{\mathrm{G}} - \nu^{\star} = o\left(\left\|d_{\mathrm{z}}\right\|_2\right).\end{aligned} \tag{A.14}$$

Therefore, by using the KKT conditions in (2.2) we have

$$\begin{aligned}\nabla J(z^{\star}) + \nabla g(z^{\star})\lambda_{\mathrm{G}} + \nabla h(z^{\star})\nu_{\mathrm{G}} = \nabla J(z^{\star}) + \nabla g(z^{\star})\lambda^{\star} + \nabla h(z^{\star})\nu^{\star}\\ + \nabla g(z^{\star})\left(\lambda_{\mathrm{G}} - \lambda^{\star}\right) + \nabla h(z^{\star})\left(\nu_{\mathrm{G}} - \nu^{\star}\right) = o\left(\left\|d_{\mathrm{z}}\right\|_2\right).\end{aligned}$$

Since $\lambda_{\mathrm{G}}^{\top}d_{\mathrm{s}} \leq 0$ (where equality holds when the correct active set is identified; see the proof of Lemma 4.10),

$$\phi(1) - \phi(0) \leq \frac{1}{2}\phi'(0) + \frac{1}{2}\lambda_{\mathrm{G}}^{\top}d_{\mathrm{s}} + o\left(\left\|d_{\mathrm{z}}\right\|_2^2\right) \leq \frac{1}{2}\phi'(0) + o\left(\left\|d_{\mathrm{z}}\right\|_2^2\right).$$

To prove that eventually $\phi(1) - \phi(0) - \sigma_1\phi'(0) \leq 0$ with $\sigma_1 \in (0, 1/2)$, from the previous inequality we have that

$$\phi(1) - \phi(0) - \sigma_1\phi'(0) \leq \underbrace{\left(\frac{1}{2} - \sigma_1\right)}_{>0}\underbrace{\phi'(0)}_{<0} + o\left(\left\|d_{\mathrm{z}}\right\|_2^2\right) \;\leq\; 0$$

for all sufficiently large i, since by Lemma 4.4 the term $\phi'(0)$ is upper bounded by a negative term proportional to $\|d_{\mathrm{z}}\|_2^2$, and the term $o(\|d_{\mathrm{z}}\|_2^2)$ vanishes at least quadratically fast as i approaches infinity. ∎

Proof of Lemma 4.10. Let us use the simplified notation and denote $J(z)$ as J, $J(z + d_{\mathrm{z}})$ as J^{+}, and $J(z^{\star})$ as $J^{\star}$ (and similarly for g and h and their gradients).

By definition, the derivative $\phi'(1)$ is

$$\begin{aligned}\phi'(1) = d_{\mathrm{z}}^{\top}\left(\nabla J^{+} + \nabla g^{+}\,\lambda_{\mathrm{G}} + \nabla h^{+}\,\nu_{\mathrm{G}} + \rho\nabla g^{+}\,\left(g^{+} - g - \nabla g^{\top}d_{\mathrm{z}}\right) + \rho\nabla h^{+}\,h^{+}\right)\\ + d_{\lambda}^{\top}(g^{+} - g - \nabla g^{\top}d_{\mathrm{z}}) + d_{\nu}^{\top}h^{+} + d_{\mathrm{s}}^{\top}\left(\lambda_{\mathrm{G}} + \rho(g^{+} - g - \nabla g^{\top}d_{\mathrm{z}})\right),\end{aligned}$$

and by (A.11) we have

$$\begin{aligned}\phi'(1) = d_{\mathrm{z}}^{\top}\left(\nabla J^{\star} + \nabla g^{\star}\,\lambda_{\mathrm{G}} + \nabla h^{\star}\,\nu_{\mathrm{G}}\right) + \rho d_{\mathrm{z}}^{\top}\nabla g^{+}\,\left(g^{+} - g - \nabla g^{\top}d_{\mathrm{z}}\right)\\ + d_{\mathrm{s}}^{\top}\left(\lambda_{\mathrm{G}} + \rho(g^{+} - g - \nabla g^{\top}d_{\mathrm{z}})\right) + o\left(\left\|d_{\mathrm{z}}\right\|_2^2\right),\end{aligned}$$

where we have used the fourth condition in Assumption 4.2 and the following relations resulting from (3.1) and Taylor expansions:

$$g^{+} - g - \nabla g^{\top}d_{\mathrm{z}} = o\left(\left\|d_{\mathrm{z}}\right\|_2\right),\; h^{+} = \underbrace{h + \nabla h^{\top}d_{\mathrm{z}}}_{=0} + o\left(\left\|d_{\mathrm{z}}\right\|_2\right) = o\left(\left\|d_{\mathrm{z}}\right\|_2\right).$$

Moreover, by (2.9) $d_{\mathrm{s}} = -\nabla g^{\top} d_{\mathrm{z}} - (g+s)$; hence,

$$\phi(1) = d_{\mathrm{z}}^{\top}\left(\nabla J^{\star} + \nabla g^{\star}\,\lambda_{\mathrm{G}} + \nabla h^{\star}\,\nu_{\mathrm{G}}\right) + d_{\mathrm{s}}^{\top}\lambda_{\mathrm{G}} + \rho(d_{\mathrm{z}}^{\top}(\nabla g^{+} - \nabla g) - (g+s)^{\top})\,\left(g^{+} - g - \nabla g^{\top} d_{\mathrm{z}}\right) + o\left(\|d_{\mathrm{z}}\|_2^2\right).$$

From (2.7) and Assumption 4.2, $\|g+s\|_2 = \mathcal{O}\left(\|d_{\mathrm{z}}\|_2\right)$ and $d_{\mathrm{s}}^{\top}\lambda_{\mathrm{G}} = 0$ when the correct active set is determined; see Lemma 4.2. In fact, if $g_j < 0$ for some j at $z^{\star}$ then $\lambda_{\mathrm{G},j} = 0$. Then, if $\rho = 0$, $s_j = -g_j$. Otherwise, if $\rho > 0$, as λ_j converges to $\lambda_j^{\star} = 0$, for sufficiently large iterates i we have that $0 \le \lambda_j < -\rho g_j$. Then, by (2.7) we have

$$s_j = -g_j - \frac{\lambda_j}{\rho} \quad \Rightarrow \quad g_j + s_j = -\frac{\lambda_j}{\rho} = -\frac{\lambda_j - \lambda_j^{\star}}{\rho} = \mathcal{O}\left(\|d_{\mathrm{z}}\|_2\right).$$

If $g_j = 0$, then $s_j = 0$ and $\lambda_{\mathrm{G},j} > 0$ by Assumption 4.1. By the complementarity conditions in (3.4) and (2.7) it results that $\lambda_{\mathrm{G},j}(s_j + d_{\mathrm{s},j}) = 0$, thus $d_{\mathrm{s},j} = 0$. Therefore, in any case we conclude that

$$|\rho(d_{\mathrm{z}}^{\top}(\nabla g^{+} - \nabla g) - (g+s))\left(g^{+} - g - \nabla g^{\top} d_{\mathrm{z}}\right)| = o\left(\|d_{\mathrm{z}}\|_2^2\right). \tag{A.15}$$

Since by Assumption 4.2 the quantity $\nabla J^{\star} + \nabla g^{\star}\lambda_{\mathrm{G}} + \nabla h^{\star}\nu_{\mathrm{G}}$ is $o(\|d_{\mathrm{z}}\|_2)$, then $\phi'(1) = o(\|d_{\mathrm{z}}\|_2^2)$. By Lemma 4.4, $|\phi'(0)| \ge \frac{1}{2\alpha}\|d_{\mathrm{z}}\|_2^2$; thus (2.11b) will eventually be satisfied at every iteration. ∎

Appendix B. Equivalence of the squared-slack problem. The following statement shows the equivalence properties between problems (2.1) and (5.1).

LEMMA B.1 (see [43, Proposition 1], [1, section 3.3.2]). *The following hold:*

(i) *$z^{\star}$ is a regular solution to* (2.1) *if and only if $[z^{\star}; y^{\star}]$ is a regular solution to* (5.1)*;*

(ii) *if $(z^{\star}, \lambda^{\star}, \nu^{\star})$ is a KKT triple for* (2.1)*, then $([z^{\star}; y^{\star}], [\lambda^{\star}; \nu^{\star}])$ is a KKT double for* (5.1)*;*

(iii) *if $([z^{\star}; y^{\star}], [\lambda^{\star}; \nu^{\star}])$ is a KKT double for* (5.1) *and $\lambda^{\star} \ge 0$, then $(z^{\star}, \lambda^{\star}, \nu^{\star})$ is a KKT triple for* (2.1)*.*

Note that by the feasibility of the optimal solution to (5.1), $y^{\star}$ is such that $y_j^{\star} = \left(-2g_j\left(z^{\star}\right)\right)^{1/2}$ for all $j \in \{1, \ldots, m\}$.

For the proof of part (iii), it is necessary to assume $\lambda^{\star} \ge 0$; in fact, the first-order conditions for (5.1) may in principle have negative dual variables $\lambda^{\star}$ associated with the squared-slack equality constraints. Note that the first-order conditions, derived with respect to y, already guarantee complementarity slackness, i.e., $\lambda_j^{\star} = 0$ for all indexes j of the active inequality constraints at $z^{\star}$ for (2.1).

The following lemma shows that the assumption $\lambda^{\star} \ge 0$ of part (iii) can be dropped if the solution $[z^{\star}; y^{\star}]$ is a local minimum.

LEMMA B.2. *If $[z^{\star}; y^{\star}]$ is a local minimum of* (5.1) *with dual variables $[\lambda^{\star}; \nu^{\star}]$, then $(z^{\star}, \lambda^{\star}, \nu^{\star})$ is a KKT triple for* (2.1)*.*

Proof. By the second-order necessary condition for (5.1), we have

$$\begin{bmatrix} \tilde{z}^{\top} & \tilde{y}^{\top} \end{bmatrix} \begin{bmatrix} \nabla_{zz}^2 \mathcal{L}(z^{\star}, \lambda^{\star}, \nu^{\star}) & 0 \\ 0 & \mathrm{diag}(\lambda^{\star}) \end{bmatrix} \begin{bmatrix} \tilde{z} \\ \tilde{y} \end{bmatrix} \ge 0 \tag{B.1}$$

for all $\tilde{z} \in \mathbb{R}^n$, $\tilde{y} \in \mathbb{R}^m$ such that

$$\nabla h(z^\star)^\top \tilde{z} = 0, \qquad \nabla g_j(z^\star)^\top \tilde{z} + y_j^\star \tilde{y}_j = 0 \ \text{ for all } j \in \{1, \dots, m\}. \tag{B.2}$$

Let j be the index of an arbitrary active constraint of $z^\star$. We can choose $\tilde{z} = 0$, with $\tilde{y}_j \neq 0$, and $\tilde{y}_k = 0$ for all $k \neq j$. Therefore, by (B.1), we obtain $\lambda_j^\star \tilde{y}_j^2 \geq 0$, thus $\lambda_j^\star \geq 0$. By the complementarity slackness, implied by the first-order conditions derived with respect to y, we conclude that $\lambda^\star \geq 0$. The result then follows from Lemma B.1, part (iii).

Acknowledgment. We thank the anonymous reviewer for suggesting the use of the adjoint functional $J_{\mathrm{a}}(u)$ to compute in a compact form the derivative of the objective function for the MPC problems.

REFERENCES

[1] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1999.
[2] P. T. Boggs and J. W. Tolle, *Sequential quadratic programming*, Acta Numer., 4 (1995), pp. 1–51.
[3] P. T. Boggs, J. W. Tolle, and P. Wang, *On the local convergence of quasi-Newton methods for constrained optimization*, SIAM J. Control Optim., 20 (1982), pp. 161–171.
[4] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, 2004.
[5] C. G. Broyden, J. E. Dennis, and J. J. Moré, *On the local and superlinear convergence of quasi-Newton methods*, IMA J. Appl. Math., 12 (1973), pp. 223–245.
[6] H. Chen and F. Allgöwer, *A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability*, Automatica J. IFAC, 34 (1998), pp. 1205–1217.
[7] T. F. Coleman, *On characterizations of superlinear convergence for constrained optimization*, in Computational Solution of Nonlinear Systems of Equations, E. Allgower, ed., American Mathematical Society, Providence, RI, 1990, pp. 113–133.
[8] R. H. Day, *Recursive Programming and Production Response*, North-Holland Amsterdam, 1963.
[9] S. de Oliveira Kothare and M. Morari, *Contractive model predictive control for constrained nonlinear systems*, IEEE Trans. Automat. Control, 45 (2000), pp. 1053–1071.
[10] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
[11] J. E. Dennis, Jr, and J. J. Moré, *Quasi-Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89.
[12] J. E. Dennis, Jr, and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, 16, SIAM, Classics Appl. Math Philadelphia, 1996.
[13] M. Diehl, H. 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.
[14] M. Diehl, R. Findeisen, F. Allgöwer, H. G. Bock, and J. P. Schlöder, *Nominal stability of real-time iteration scheme for nonlinear model predictive control*, IEE Proc. Control Theory, 152 (2005), pp. 296–308.
[15] A. Domahidi and J. Jerez, *FORCES Professional*, embotech GmbH, http://embotech.com/FORCES-Pro, July 2014.
[16] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, *Efficient interior point methods for multistage problems arising in receding horizon control*, in Proceedings of the IEEE Conference on Decision and Control, 2012, pp. 668–674.
[17] H. J. Ferreau, H. G. Bock, and M. Diehl, *An online active set strategy to overcome the limitations of explicit MPC*, Internat. J. Robust Nonlinear Control, 18 (2008), pp. 816–830.
[18] A. V. Fiacco, *Sensitivity analysis for nonlinear programming using penalty methods*, Math. Program., 10 (1976), pp. 287–311.
[19] R. Fletcher, *A class of methods for nonlinear programming. III. Rates of convergence*, in Numerical Methods for Nonlinear Optimitation, F. A. Lootsma, ed., Academic Press, New York, 1972, pp. 371–382.
[20] S. Ghadimi and G. Lan, *Accelerated gradient methods for nonconvex nonlinear and stochastic programming*, Math. Program., 156 (2016), pp. 59–99.
[21] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Rev., 47 (2005), pp. 99–131.

[22] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *Some theoretical properties of an augmented Lagrangian merit function*, in Advances in Optimization and Parallel Computing, North-Holland, Amsterdam, 1992, pp. 101–128.
[23] J. Goodman, *Newton's method for constrained optimization*, Math. Program., 33 (1985), pp. 162–171.
[24] S.-P. Han, *Superlinearly convergent variable metric algorithms for general nonlinear programming problems*, Math. Program., 11 (1976), pp. 263–282.
[25] K. Holmström, *The TOMLAB optimization environment in Matlab*, Adv. Model. Optim., 1 (1999), pp. 47–69.
[26] K. C. Kiwiel, *An algorithm for linearly constrained convex nondifferentiable minimization problems*, J. Math. Anal. Appl., 105 (1985), pp. 452–465.
[27] F. Leibfritz and E. W. Sachs, *Inexact SQP interior point methods and large scale optimal control problems*, SIAM J. Control Optim., 38 (1999), pp. 272–293.
[28] C. Lemaréchal, *A view of line-searches*, in Optimization and Optimal Control, Springer-Verlag, Berlin 1981, pp. 59–78.
[29] M. Morari and J. H. Lee, *Model predictive control: past, Present and future*, Comput. Chem. Eng., 23 (1999), pp. 667–682.
[30] W. Murray and F. J. Prieto, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, SIAM J. Optim., 5 (1995), pp. 590–640.
[31] Y. Nesterov, *A method of solving a convex programming problem with convergence rate* $O\,(1/k2)$, Doklady Math., 27 (1983), pp. 372–376.
[32] Y. Nesterov, *Introductory lectures on convex optimization: A basic course*, Appl. Optim. 87, Springer, New York, 2013.
[33] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Math. Comp., 35 (1980), pp. 773–782.
[34] J. Nocedal, *Theory of algorithms for unconstrained optimization*, Acta Numer., 1 (1992), pp. 199–242.
[35] J. Nocedal and S. Wright, *Numerical Optimization*, Springer-Verlag, Berlin, 1999.
[36] F. Palacios-Gomez, L. Lasdon, and M. Engquist, *Nonlinear optimization by successive linear programming*, Manag. Sci., 28 (1982), pp. 1106–1120.
[37] M. J. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, G. A. Watson, ed., Springer-Verlag, Berlin, 1978, pp. 144–157.
[38] M. J. Powell and Y. Yuan, *A recursive quadratic programming algorithm that uses differentiable exact penalty functions*, Math. Program., 35 (1986), pp. 265–278.
[39] R. Quirynen, S. Gros, B. Houska, and M. Diehl, *Lifted collocation integrators for direct optimal control in ACADO toolkit*, Math. Program. Comput., 9 (2017), pp. 527–571.
[40] S. Richter, C. N. Jones, and M. Morari, *Real-time input-constrained MPC using fast gradient methods*, in Proceedings of the IEEE Conference on Decision and Control and Chinese Control Conference, 2009, pp. 7387–7393.
[41] S. M. Robinson, *Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms*, Math. Program., 7 (1974), pp. 1–16.
[42] J. B. Rosen, *The gradient projection method for nonlinear programming. Part II. Nonlinear constraints*, SIAM J. Appl. Math., 9 (1961), pp. 514–532.
[43] R. Tapia, *A stable approach to Newton's method for general mathematical programming problems in* $\mathbb{R}^n$, J. Optim. Theory Appl., 14 (1974), pp. 453–476.
[44] R. Tapia, *Quasi-Newton methods for equality constrained optimization: Equivalence of existing methods and a new implementation*, in Nonlinear Programming 3 Nonlinear Programming 3, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Elsevier, New York, 1978, pp. 125–164.
[45] M. J. Tenny, S. J. Wright, and J. B. Rawlings, *Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming*, Comput. Optim. Appl., 28 (2004), pp. 87–121.
[46] G. Torrisi, *Low-Complexity Numerical Methods for Nonlinear Model Predictive Control*, Ph.D. thesis, ETH Zurich, 2017, https://doi.org/10.3929/ethz-b-000225851.
[47] G. Torrisi, D. Frick, T. Robbiani, S. Grammatico, R. S. Smith, and M. Morari, *FalcOpt: First-order Algorithm via Linearization of Constraints for OPTimization*, https://github.com/torrisig/FalcOpt, May 2017.
[48] G. Torrisi, S. Grammatico, D. Frick, T. Robbiani, R. S. Smith, and M. Morari, *Low-complexity first-order constraint linearization methods for efficient nonlinear MPC*, in Proceedings of the IEEE Conference on Decision and Control, 2017, pp. 4376–4381.
[49] G. Torrisi, S. Grammatico, R. S. Smith, and M. Morari, *A variant to sequential quadratic programming for nonlinear model predictive control*, in Proceedings of the IEEE Conference on Decision and Control, 2016, pp. 2814–2819.