SIAM J. OPTIM.
Vol. 27, No. 1, pp. 538–564

A PREDICTOR-CORRECTOR PATH-FOLLOWING ALGORITHM FOR DUAL-DEGENERATE PARAMETRIC OPTIMIZATION PROBLEMS*

VYACHESLAV KUNGURTSEV† AND JOHANNES JÄSCHKE‡

Abstract. Most path-following algorithms for tracing a solution path of a parametric nonlinear optimization problem are only certifiably convergent under strong regularity assumptions about the problem functions. In particular, linear independence of the constraint gradients at the solutions is typically assumed, which implies unique multipliers. In this paper we propose a procedure designed to solve problems satisfying a weaker set of conditions, allowing for nonunique (but bounded) multipliers. Each iteration along the path consists of three parts: (1) a Newton corrector step for the primal and dual variables, which is obtained by solving a linear system of equations, (2) a predictor step for the primal and dual variables, which is found as the solution of a quadratic programming problem, and (3) a jump step for the dual variables, which is found as the solution of a linear programming problem. We present a convergence proof and demonstrate the successful solution tracking of the algorithm numerically on a couple of illustrative examples.

Key words. parametric optimization, predictor-corrector path-following, dual-degeneracy, optimal solution sensitivity

AMS subject classifications. 90C30, 90C31

DOI. 10.1137/16M1068736

1. Introduction.

We consider the parametric optimization problem with $f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ and $c : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$,

$$(1.1)\qquad \begin{array}{ll} \min_{x\in\mathbb{R}^n} & f(x,t) \\ \text{subject to} & c_i(x,t) = 0,\ i \in \mathcal{E}, \\ & c_i(x,t) \geq 0,\ i \in \mathcal{I}, \end{array}$$

where $\mathcal{E} = \{1, \ldots, m_e\}$, and $\mathcal{I} = \{m_e + 1, \ldots, m\}$, and we seek to trace the solution path along a parameter change from $t = 0$ to $t = 1$.

We assume that $\nabla_x c(x, t_1) = \nabla_x c(x, t_2)$ and $\nabla_{xx} f(x, t_1) = \nabla_{xx} f(x, t_2)$ for all t_1, t_2 as well as $\nabla_{xt} f(x_1, t) = \nabla_{xt} f(x_2, t)$ and $\nabla_t c(x_1, t) = \nabla_t c(x_2, t)$ for any two x_1 and x_2. In particular, these conditions imply that $f(x, t)$ and $c(x, t)$ are of the form $f(x, t) = f_0(x) + (a_f^T x)t$ and $c(x, t) = c^0(x) + a_c t$, where $a_f \in \mathbb{R}^n$ and $a_c \in \mathbb{R}^m$. This places the optimization problem under the standard notion of *canonical perturbations* [28].

*Received by the editors April 4, 2016; accepted for publication (in revised form) December 27, 2016; published electronically March 28, 2017.
http://www.siam.org/journals/siopt/27-1/M106873.html
Funding: The first author's research was supported by the European social fund within the framework of realizing the project support of inter-sectoral mobility and quality enhancement of research teams at the Czech Technical University in Prague, CZ.1.07/2.3.00/30.0034, the Cisco-CTU Sponsored Research Agreement project WP5, and the Czech Science Foundation project 17-26999S.
†Agent Technology Center, Department of Computer Science, Faculty of Electrical Engineering, Czech Technical University in Prague, Prague, Czech Republic (vyacheslav.kungurtsev@fel.cvut.cz).
‡Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway (jaschke@ntnu.no).

Note that the problem class in problem (1.1) is not as restrictive as it may seem, as a more generic parametric optimization problem,

$$\begin{array}{ll} \min_{x\in\mathbb{R}^n} & \tilde{f}(x,s) \\ \text{subject to} & \tilde{c}_i(x,s)=0,\, i\in\{1,\dots,\tilde{m}_e\}, \\ & \tilde{c}_i(x,s)\geq 0,\, i\in\{\tilde{m}_e+1,\dots,\tilde{m}\}, \end{array}$$

where $s \in \mathbb{R}^p$ is a vector and the solution is traced from s_0 to s_f, and $\tilde{f}(x,s)$ and $\tilde{c}(x,s)$ have more arbitrary, potentially nonlinear dependence on s, can be rewritten in the form (1.1) by the incorporation of another variable z, writing $\tilde{f}$ and $\tilde{c}$ as $\tilde{f}(x,z)$ and $\tilde{c}(x,z)$ and adding the equality constraint $c_{\tilde{m}+1} = z-(1-t)s_0 - ts_f = 0$.

Parametric problems such as (1.1) occur in many applications such as model predictive control [7, 24, 29] and stochastic [12], global [38], and bilevel optimization [5]. Fast and accurate performance is especially demanded for real-time model predictive control, for which procedures that perform with sufficient speed for linear problems are abundant, for example, [36], but the implementation for nonlinear models has generally been more challenging

In this paper we present an algorithm for parametric optimization algorithms that applies sensitivity theory of optimization problems subject to perturbations. A novel feature of this path-following algorithm is that it is provably convergent for problems that are dual-degenerate, i.e., do not have a unique multiplier for the optimization problem at every value of the parameter.

1.1. Notation. Given vectors a, $b \in \mathbb{R}^n$, $\min(a,b)$ is the vector with components $\min(a_i,b_i)$. The vectors e and e_j denote, respectively, the column vector of ones and the jth column of the identity matrix I. The dimensions of e, e_i, and I are defined by the context. The ith component of a vector labeled with a subscript will be denoted by $[v_\alpha]_i$. Similarly, if $\mathcal{K}$ is an index set, $v_\mathcal{K}$ and $[v_\alpha]_\mathcal{K}$ indicate the vector with $|\mathcal{K}|$ components composed of the entries of v and v_α, respectively, corresponding to those indices in $\mathcal{K}$. If there exists a positive constant γ such that $\|\alpha_j\| \leq \gamma\beta_j$, we write $\alpha_j = O(\beta_j)$. If there exists a sequence $\gamma_j \to 0$ such that $\|\alpha_j\| \leq \gamma_j\beta_j$, we say that $\alpha_j = o(\beta_j)$.

1.2. Definitions. It is important to define some foundational terminology necessary for understanding the presentation of the material. We begin with presenting the form of the optimality conditions we will be using in this paper. These conditions necessarily hold for any local minimizer x^* which satisfies a constraint qualification, a geometric regularity condition involving the local properties of the feasible region. There are a number of constraint qualifications of varying restrictiveness, and we will mention a few of them later in this section.

DEFINITION 1.1 (first-order necessary conditions). *A vector $x^* \in \mathbb{R}^n$ satisfies the* first-order necessary optimality conditions *for* (1.1) *at t if there exists a $y^* \in \mathbb{R}^m$ such that*

$$\begin{aligned} \nabla_x f(x^*,t) &= \nabla_x c(x^*,t)y^*, \\ c_i(x^*,t) &= 0,\, i\in\mathcal{E}, \\ c_i(x^*,t) &\geq 0,\, i\in\mathcal{I}, \\ c(x^*,t)^T y^* &= 0, \\ y_i^* &\geq 0,\, i\in\mathcal{I}. \end{aligned} \tag{1.2}$$

We denote $\Lambda(x^*,t)$ as the set of dual vectors y^* corresponding to x^* such that (x^*,y^*) satisfy the first-order necessary conditions at t.

Denoting the cone $\mathcal{N}(y)$ to be

$$\mathcal{N}(y) = \begin{cases} \{z | z \geq 0 \text{ and } z^T y = 0\} & \text{if } y \geq 0, \\ \emptyset & \text{otherwise,} \end{cases}$$

an alternative formulation of (1.2) is given as

$$\begin{aligned} \nabla_x f(x^*, t) &= \nabla_x c(x^*, t) y^*, \\ c_{\mathcal{E}}(x^*, t) &= 0, \\ c_{\mathcal{I}}(x^*, t) &\in \mathcal{N}(y_{\mathcal{I}}). \end{aligned}$$

We will define $\mathcal{A}(x^*, t)$ to be the set of inequality constraint indices $i \in \mathcal{I}$ such that for $i \in \mathcal{A}(x^*, t)$, $c_i(x^*, t) = 0$, $\mathcal{A}_0(x^*, y^*, t) \subseteq \mathcal{A}(x^*, t)$ to be the set such that $i \in \mathcal{A}_0(x^*, y^*, t)$ implies that $[y^*]_i = 0$ and $\mathcal{A}_+(x^*, y^*, t) \subseteq \mathcal{A}(x^*, t)$ to be the set such that $i \in \mathcal{A}_+(x^*, y^*, t)$ implies that $[y^*]_i > 0$. We define $\mathcal{A}_+(x^*, t) = \cup_{y^* \in \Lambda(x^*, t)} \mathcal{A}_+(x^*, y^*, t)$ and $\mathcal{A}_0(x^*, t) = \cap_{y^* \in \Lambda(x^*, t)} \mathcal{A}_0(x^*, y^*, t)$.

The Lagrangian function associated with (1.1) is $L(x, y, t) = f(x, t) - c(x, t)^T y$. The Hessian of the Lagrangian with respect to x is denoted by

$$H(x, y, t) := \nabla^2_{xx} f(x, t) - \sum_{i=1}^{m} y_i \nabla^2_{xx} c_i(x, t).$$

The strong form of the second-order sufficiency condition is defined as follows.

DEFINITION 1.2 (strong second-order sufficient conditions (SSOSC)). *Let us define the set,*

$$\tilde{\mathcal{C}}(x^*, y^*, t) := \left\{ d : \nabla_x c_i(x^*, t)^T d = 0 \text{ for } i \in \mathcal{A}_+(x^*, y^*, t) \cup \mathcal{E} \right\}$$

A primal-dual pair (x^*, y^*) satisfies the strong second-order sufficient optimality conditions *at t if it satisfies the first-order conditions* (1.2) *and*

$$d^T H(x^*, y^*, t) d > 0 \text{ for all } d \in \tilde{\mathcal{C}}(x^*, y^*, t) \setminus \{0\}. \tag{1.3}$$

We will be interested in the general SSOSC, applied across all multipliers in the optimal set.

DEFINITION 1.3 (general SSOSC (GSSOSC)). *A primal vector x^* satisfies the* General Strong Second-order Sufficient Optimality Conditions *at t if (x^*, y^*) satisfies the SSOSC for all $y^* \in \Lambda(x^*, t)$.*

We shall now define a few constraint qualifications (CQs) relevant for this paper. A CQ is required to hold at a feasible point in order for a point being a local minimizer for an optimization problem to imply that the optimality conditions hold. It is standard for convergence proofs for algorithms to assume some CQ, and the weaker the CQ condition is, the more problems the convergence theory applies to.

DEFINITION 1.4. *The* linear independence constraint qualification (LICQ) *holds for* (1.1) *at t for a feasible point x if the set of vectors $\{\nabla_x c_i(x, t)\}_{i \in \mathcal{E} \cup \mathcal{A}(x, t)}$ is linearly independent.*

DEFINITION 1.5. *The* Mangasarian–Fromovitz constraint qualification (MFCQ) *holds for* (1.1) *at t for a feasible point x if*

1. $\{\nabla_x c_i(x, t)\}_{i \in \mathcal{E}}$ *is linearly independent and*

2. *there exists a* p *such that* $\nabla_x c_i(x,t)^T p = 0$ *for all* $i \in \mathcal{E}$ *and* $\nabla_x c_i(x,t)^T p > 0$ *for all* $i \in \mathcal{A}(x,t)$.

Equivalently, by the theorem of the alternative [34], *the MFCQ holds if there is no set of scalars* $\{\alpha_i\}_{i\in\{1,\dots,m\}}$ *such that*

1. *for* $i \in \mathcal{I}$, $\alpha_i \geq 0$,
2. *either there exists* $i \in \mathcal{E}$ *such that* $\alpha_i \neq 0$ *or* $\sum_{i\in\mathcal{I}} \alpha_i > 0$, *and*
3. $$\sum_{i\in\mathcal{E}\cup\mathcal{A}(x,t)} \alpha_i \nabla_x c_i(x,t) = 0.$$

DEFINITION 1.6. *The* constant rank constraint qualification (CRCQ) *holds for* (1.1) *at* t *for a feasible point* x *if there exists a neighborhood* $\mathcal{N}$ *of* x *such that for all subsets* $\mathcal{U} \subseteq \mathcal{E} \cup \mathcal{A}(x,t)$, *the rank of* $\{\nabla_x c_i(x,t)\}_{i\in\mathcal{U}}$ *is equal to the rank of* $\{\nabla_x c_i(\bar{x},t)\}_{i\in\mathcal{U}}$ *for all* $\bar{x} \in \mathcal{N}$.

Of special interest in this paper is the multifunction corresponding to a local primal-dual solution set for the NLP subject to a parameter,

$$K(t) := \{x^*(t), \Lambda(x^*(t),t)\}.$$

Since in general, under the assumptions we are concerned with in this paper, $\Lambda(x^*(t),t)$ is not a singleton, some notion of set-valued differentiabiliy will be necessary, as we are interested in tracing some optimal $\lambda^*(t) \in \Lambda(x^*(t),t)$ along t. We shall employ the outer graphical derivative and the concept of proto-differentiability [33], one generalized notion of differentiability for set-valued maps.

DEFINITION 1.7. *For a multifunction* $S : \mathbb{R}^p \rightrightarrows \mathbb{R}^d$, *the* outer graphical derivative *of* S *at* $\bar{w}$ *for* $\bar{v} \in S(\bar{w})$ *is denoted by* $DS(\bar{w}|\bar{v}) : \mathbb{R}^p \rightrightarrows \mathbb{R}^d$ *and defined as*

$$DS(\bar{w}|\bar{v})w' = \{v' | \exists w'_\nu \to w', \tau_\nu \downarrow 0 \text{ with } (\bar{v}_\nu - \bar{v})/\tau_\nu \to v' \text{ for some } \bar{v}_\nu \in S(\bar{w} + \tau_\nu w'_\nu)\}.$$

S *is said to be* proto-differentiable *at* $\bar{w}$ *for* $\bar{v}$ *if every vector* $(w',v') \in$ *graph*$(DS(\bar{w}|\bar{v})w')$ *is equal to the following limit:*

$$(w',v') = \lim_{s\downarrow 0} \frac{(w(s),v(s)) - (\bar{w},\bar{v})}{s} \tag{1.4}$$

for some selection mapping $s \to (w(s),v(s)) : [0,\epsilon] \to$ *graph*(S) *for some small* ϵ.

We further define the distance of a point to the nearest primal-dual solution by $\theta(x,y,t)$,

$$\theta(x,y,t) = \sqrt{\|x - x^*(t)\|^2 + \text{dist}(y, \Lambda(x^*(t),t))^2},$$

where $x^*(t)$ is the closest primal solution to (1.1) at t. We will also sometimes write $\theta(x,t)$ to denote $\theta(x,t) = \|x - x^*(t)\|$. The optimality residual $\eta(x,y,t)$ is defined as

$$\eta(x,y,t) = \left\| \begin{pmatrix} \nabla_x f(x,t) - \nabla_x c(x,t) y \\ c(x,t)_{\mathcal{E}} \\ [\min(c(x,t),y)]_{\mathcal{I}} \end{pmatrix} \right\|_\infty .$$

2. Background.

2.1. Previous results on parametric optimization and contribution of this paper. In this section we review the literature on parametric optimization, highlighting several important features that have proven useful for the development of fast and reliable algorithms.

A typical path-following procedure for nonlinear parametric equations includes a predictor and a corrector, where a predictor uses the tangent to the solution path to estimate the solution at a subsequent parameter value, and a corrector modifies the predictor step by incorporating additional information to take a step closer to the solution path. (For an overview, see [1].) Predictor steps are first-order approximations, and corrector steps use some form of Newton iteration. In the case of parametric NLPs, the notion of *sensitivity*, how the optimal solution and objective value change as a parameter in an optimization problem is subject to small perturbations [3], is essential for formulating appropriate predictors. For understanding corrector steps and a discussion of Newton's method for continuation, see, for instance, [6, Chapter 5]. When a predictor and corrector are combined in an overall algorithm, to avoid additional function evaluations, approximate information is used for problem data for a Newton step. This necessitates the analysis framework of inexact Newton methods [4]. (See also [11, 23] for contemporary frameworks.) In addition, in some cases arising in optimal control, approximate problem information is used to begin with [8], also necessitating analysis from an inexact Newton perspective.

The seminal book [19] considered the parametric optimization problem in considerable detail. The authors classified all of the possible forms of primal or dual bifurcation of the parametric optimization problem and the degeneracy conditions that are associated with these bifurcations. Based on this classification and the sensitivity theory available at the time, the authors formulated procedures for tracing the solution path along a homotopy. New sensitivity theory has been developed after this book appeared, however, particularly for degenerate problems (see, e.g., [22]). We note, also, that out of this literature arose path-following algorithms that can be applied as machinery to solve one-off NLPs [20].

New results on the sensitivity of nonlinear programs (NLPs) subject to a parameter have motivated some predictor methods [24, 40]. (See also [31] for a predictor algorithm for multiobjective optimization.) However, the directional derivative (predictor) is just a first-order estimate. If the problem is highly nonlinear and there is notable curvature in the solution path, the computed path can diverge from the true solution path.

Because of their desirable warm-start properties, sequential quadratic programming (SQP) methods [18], wherein a solution of an NLP is found by solving a sequence of approximating quadratic programs (QP), were presented as a reasonable choice for solving parametric problems [7]. In particular, one solves the problem at each parameter with a few steps of an SQP algorithm, using the solution at the previous value of the parameter as an initial guess for the subsequent one. With a correctly estimated active set, an SQP iteration can be equivalent to a Newton step and thus corresponds precisely to a corrector. One major difficulty with practical implementation of SQP methods, however, is that whereas at the solution satisfying the SSOSC, the reduced Hessian (Lagrangian Hessian in the particular subspace) is positive definite, the full Hessian may not be, and the resulting subproblems may be nonconvex, with possibly multiple or unbounded solutions. (For a detailed discussion of these challenges, see [25, Chapter 5].) There are a number of strategies devised to deal with these issues, but they typically require the use of inexact Hessian information and thus lose the Newton/corrector-like behavior of the algorithm and as such the major potential benefit of using SQP for parametric optimization.

In addition, a strongly desirable property of a parametric optimization algorithm, in particular one applicable to optimal control problems, is the capacity to handle degeneracy. In this paper we will assume that the MFCQ and the CRCQ hold for every t across the solution path, but not necessarily the LICQ. These conditions are typical for optimal control problems [35] (see also [24]) because the dynamics of the system appear discretized as equality constraints in the optimization problem, and aside from certain classes of singular ODEs/DAEs, these are expected to have linearly independent gradients if they are well formulated. However, the controls and states are often subject to bound constraints or simple linear constraints, and when many of these constraints are active, the entire set of equality plus active bound constraints becomes overdetermined, and the set gradients corresponding to active constraints become linearly dependent. However, bound constraints are "nice" in the sense of providing a strictly feasible direction (to satisfy the MFCQ) and are constant, and thus trivially have constant rank gradients across the entire primal space (and thus satisfy the CRCQ).

The literature with proofs of convergence of parametric optimization algorithms, while producing some interesting and powerful algorithms, with insightful analysis on the properties of the parametric problems, has always assumed strong regularity, which requires linear independence of the constraint gradients at the solution, or at least the uniqueness of the optimal multiplier at the solution for every parameter [9, 10, 39]. The paper [26] is the sole exception and considers the case of general degeneracy, including problems satisfying no constraint qualification (but the existence of a Lagrange multiplier), but the procedure suggested requires solving a number of linear programs (LPs) to find possible multipliers to branch from, and thus relies on too weak assumptions for our purposes. We note that even achieving fast local convergence for problems under weak constraint qualification conditions has just fairly recently shown to be possible, under perturbed Newtonian methods. (See, for instance, [14, 37], and [16] for a complete algorithm.)

The contributions of this paper are the following: First, we develop a predictor-corrector path-following algorithm that includes a Newton corrector step, a QP predictor step, and a multiplier jump LP step. Although the multipliers can be nonunique along the path, our algorithm also traces a multiplier path that is used to calculate the optimal primal sensitivity for the predictor step. Second, we give a proof of the convergence for the algorithm. Finally, we demonstrate that the algorithm functions as intended numerically on illustrative examples.

The corrector step is based on principles of SQP to obtain a step that contracts superlinearly to the primal-dual solution. The predictor QP and multiplier jump step are based on the sensitivity results of [30], which shows B-differentiability of the primal solution path under the CRCQ and MFCQ and the form of the directional derivative as well as how to identify the optimal multiplier to follow the path from. We also incorporate active set estimation based on [13].

In the next section we present a small example to demonstrate some of the issues that arise when tracking a solution path with nonunique multipliers.

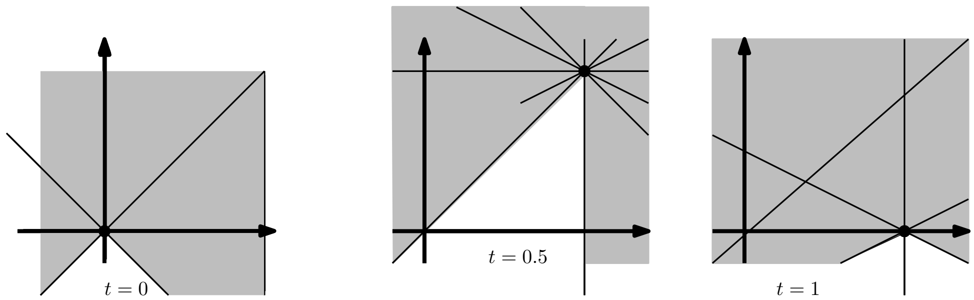

FIG. 1. *Problem (2.1) illustrated. The axes are for x_1 and x_2, the straight lines are the constraints, the black dot is the optimal point, and the grey area indicates the infeasible region.*

2.2. Illustrative example. We consider the following problem:

$$\begin{array}{lll} \min_x & -e^{x_2} + \frac{1}{2}(x_1 - x_3)^2 & \\ \text{subject to} & x_3 - 10t & = 0, \\ & x_1 - x_2 & \geq 0, \\ & 10t - x_2 & \geq 0, \\ & -x_1 - x_2 + 20t & \geq 0, \\ & 5 - x_1 & \geq 0, \\ & \frac{1}{2}x_1 - x_2 + \frac{15}{2} - 10t & \geq 0, \\ & -\frac{1}{2}x_1 - x_2 + \frac{25}{2} - 10t & \geq 0. \end{array} \tag{2.1}$$

It can be verified that the solution is $x^*(t) = (10t, 10t, 10t)$ for $t \in [0, \frac{1}{2}]$ and $x^*(t) = (5, 10 - 10t, 10t)$ for $t \in [\frac{1}{2}, 1]$. The MFCQ, but not the LICQ, holds for all $x^*(t)$, $t \in [0, 1]$ (and indeed for all t). Since all of the constraints are linear, the CRCQ also holds.

The first three inequality constraints are active for $t \in [0, \frac{1}{2})$, all constraints are active at $t = \frac{1}{2}$, and the last three constraints are active for $t \in (\frac{1}{2}, 1]$. Noting that x_3 is just a placeholder for nonlinear dependence of the objective function with respect to the parameter, i.e., setting $x_3 = 10t$, we illustrate the problem with respect to x_1 and x_2 in Figure 1 for $t = 0, \frac{1}{2}, 1$, noting by a circular dot where the primal solution is, the lines the constraints, and the grey area corresponds to the infeasible region.

The sets of optimal multipliers are

$$\Lambda(x^*(0)) = \{(0, y_1, y_2, y_1, 0, 0, 0) : 2y_1 + y_2 = 1,\ y_1, y_2 \geq 0\},$$

$$\Lambda(x^*(0.5)) = \Big\{(0, y_1, y_2, y_3, y_4, y_5, y_6) : y_1 + y_2 + y_3 + y_5 + y_6 = e^5,$$
$$y_1 - y_3 - y_4 + \frac{1}{2}y_5 - \frac{1}{2}y_6 = 0,\ y_i \geq 0, \text{ for all } i\Big\},$$

$$\Lambda(x^*(1)) = \{(5, 0, 0, 0, y_1, y_2, y_3) : -2y_1 + y_2 - y_3 = -5,\ y_2 + y_3 = 1\}.$$

Note that the last three multipliers are always zero for $t \in [0, \frac{1}{2})$, and then at least two components are strictly positive for $t \in (\frac{1}{2}, 1]$. Therefore, it is desired for a parametric optimization algorithm to be able to formulate a multiplier that jumps discontinuously across $t < 0.5$ to $t > 0.5$.

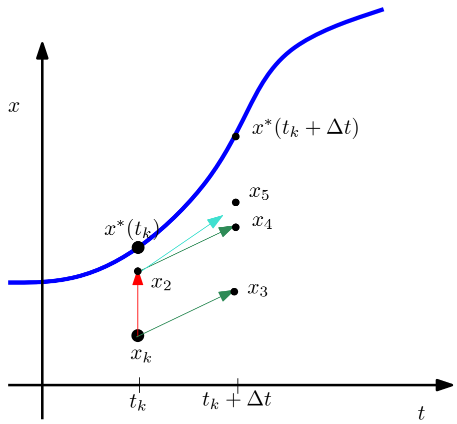

FIG. 2. *Predictor, corrector, predictor-corrector, and corrected predictor steps.*

3. Algorithm.

3.1. Overview. To illustrate our algorithm, we recall the notions of a path-following as given in, for example, [1], and consider Figure 2. Let the current point be x_k and the solution at t_k and $t_k + \Delta t$ be $x^*(t_k)$ and $x^*(t_k + \Delta t)$, respectively. We wish to take a step from x_k that approximates $x^*(t_k + \Delta t)$. A predictor step uses information about the tangent of the solution path, using the slightly inaccurate information at x_k (as in, one can take an approximation of the tangent at $x^*(t_k)$ using problem information at x_k). Taking a pure predictor step would result in moving to x_3. A pure corrector takes a step toward a more accurate solution at a given point, which corresponds to x_2. If we do not reevaluate the function or its derivatives at x_2 or x_3, combining the predictor and corrector would result in a step to x_4.

Alternatively, one can reevaluate the problem functions at x_2 or x_3 yielding a more accurate path-following procedure at the expense of additional computational cost. In our algorithm, however, starting from x_k we take a corrector step to x_2 and, without performing any additional function evaluations, use the corrector step and updated function estimates arising from linearizations, to generate a more accurate predictor, and so obtain the estimate x_5 as the approximation to $x^*(t_k + \Delta t)$. This procedure is repeated until the final value of t is reached.

3.2. Algorithm description. Now we describe the main steps of our path-following algorithm. A more detailed discussion about practical implementation will follow in section 5.

We begin each iteration with a point x_k and a multiplier y_k. The primal vector x_k needs to be (sufficiently) close to a primal solution $x^*(t_k)$, and the dual vector y_k needs to be close to a multiplier $\hat{y} \in \Lambda(x^*(t_k), t_k)$ in the optimal multiplier set for which the set $\{\nabla_x c_i(x^*(t_k), t_k) : \hat{y}_i > 0\}$ is linearly independent. We will see later how to obtain this multiplier.

Consider the robust active set estimator,

$$\mathcal{A}_\gamma(x, y, t) = \{i \in \mathcal{I} : c_i(x, t) \leq \eta(x, y, t)^\gamma\}, \tag{3.1}$$

where γ is a constant satisfying $0 < \gamma < 1$ [13]. We then form an estimate of the active set $A = \mathcal{A}_\gamma(x_k, y_k, t_k)$ and strongly active set $A_+ = \{i : y_i > 0\} \cup \mathcal{E}$.

3.2.1. Corrector step. At a given (x_k, y_k, t) we first solve the linear system (CorrectStep)

$$\begin{pmatrix} H(x_k, y_k, t) & -\nabla_x c_{A_{+,k}}(x_k, t) \\ \nabla_x c_{A_{+,k}}(x_k, t)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta_c x \\ \Delta_+ y \end{pmatrix} = - \begin{pmatrix} \nabla_x f(x_k, t) - \nabla_x c(x_k, t) y_k \\ c_{A_{+,k}}(x_k, t) \end{pmatrix}$$

to obtain the corrector step $(\Delta_c x, \Delta_+ y)$. We let $\Delta_c y \in \mathbb{R}^m$ be such that $[\Delta_c y]_{A_{+,k}} = \Delta_+ y$ and $[\Delta_c y]_{\{1,\ldots,m\}\setminus A_{+,k}} = 0$.

It can be observed that the corrector step is essentially a Newton step on the NLP at the current value of t subject to only the strongly active constraints. Given a correctly estimated strongly active set and a good starting point, at a given t, it should hold that $(\|x_k + \Delta_c x - x^*(t)\| + \text{dist}(y_k + \Delta_c y, \Lambda(x^*(t), t))) \leq C(\|x_k - x^*(t)\| + \text{dist}(y_k, \Lambda(x^*(t), t)))^2$, and the corrector step results in an iterate that is closer to the primal-dual solution set for the current t. In Figure 2, the corrector step corresponds to a step from x_k to x_2.

3.2.2. Predictor. We then solve the perturbed predictor QP subproblem (QPPredict)

$$\begin{aligned} \min_{\Delta_p x} \quad & (\nabla_x f(x_k, t + \Delta t) - \nabla_x f(x_k, t))^T \Delta_p x + \frac{1}{2} \Delta_p x^T H(x_k, y_k, t + \Delta t) \Delta_p x \\ \text{subject to} \quad & \nabla_t c_i(x_k, t)\Delta t + (\nabla_x c_i(x_k, t + \Delta t) + \nabla^2_{xx} c_i(x_k, t + \Delta t) \Delta_c x)^T \Delta_p x = 0, \\ & \qquad\qquad i \in A_{+,k}, \\ & \nabla_t c_i(x_k, t)\Delta t + (\nabla_x c_i(x_k, t + \Delta t) + \nabla^2_{xx} c_i(x_k, t + \Delta t) \Delta_c x)^T \Delta_p x \geq 0, \\ & \qquad\qquad i \in A_k \setminus A_{+,k}. \end{aligned}$$

Note that the second derivative terms $\nabla^2_{xx} c_i(x_k, t + \Delta t)$ are multiplying the corrector $\Delta_c x$ that was previously computed in (CorrectStep), and the primal variable $\Delta_p x$ of (QPPredict) appears just once in each constraint, so despite the presence of the second derivative, the constraints are still affine and the subproblem is a standard QP.

A standard predictor step, as used, for instance, in [24, 40], corresponds to this QP without the additional term $\nabla^2_{xx} c_i(x_k, t + \Delta t) \Delta_c x$ in the constraints. Note that by the properties of f and c as a function of t, no new function evaluations need to be performed between solving (CorrectStep) and (QPPredict).

In effect, (CorrectStep) produces an iterate closer to the primal-dual solution set at the given t, and without the additional term involving $\nabla^2_{xx} c_i(x_k, t + \Delta t) \Delta_c x$, (QPPredict) would be an estimate for the tangent of the path-following solution curve, with the estimate derived from information at (x_k, y_k). In Figure 2 this would correspond to, after the corrector step to x_2, taking a step to x_4. By including this extra term, the predictor uses some information from the corrector, and so the step is to x_5. Thus, this modification allows for the predictor step to improve upon and generate a more accurate prediction without requiring any new function evaluations.

Denote the primal-dual solution of this subproblem as $(\Delta_p x, \Delta_p y)$. Let $(\Delta x, \Delta y) = (\Delta_c x + \Delta_p x, \Delta_c y + \Delta_p y)$. We then obtain a new estimate for the active set by $A_{k+1} = \mathcal{A}_\gamma(x_k + \Delta x, y_k + \Delta y, t + \Delta t) \cup \mathcal{E}$.

3.2.3. Multiplier jump step. Under the MFCQ and the CRCQ, but without the LICQ holding for (1.1) across all values of t, the optimal primal solution $x^*(t)$ is continuous, but it could hold that any optimal multiplier path $y^*(t) \in \Lambda(x^*(t), t)$ is discontinuous and has one or more discrete jumps. This has been illustrated in the example given in section 2.2, where there is no continuous path of optimal multipliers that solve the problem across each side of $t = 0.5$. For $t \leq 0$ some multipliers are positive and bounded away from zero that must become zero for $t > 0.5$, and some multipliers must be zero for $t < 0$ and then become positive for $t > 0.5$. Thus we need to compute a step in the estimate of the optimal dual variables y that performs this jump.

We calculate the new multiplier, and we allow for jumps by selecting the multiplier as the solution of the following LP problem:

(JumpLP)

$$\begin{array}{rl} \min_y & y^T \nabla_t c(x+\Delta x, t+\Delta t)\Delta t \\ \text{subject to} & -|\nabla L(x_k+\Delta x, y_k+\Delta y, t+\Delta t)| \\ & \quad \leq \nabla_x f(x_k+\Delta x, t+\Delta t) + \sum_{i\in A_{k+1}} \nabla_x c_i(x_k+\Delta x, t+\Delta t) y_i \\ & \quad \leq |\nabla L(x_k+\Delta x, y_k+\Delta y, t+\Delta t)|, \\ & y_{\mathcal{I}} \geq 0, \\ & y_{i\notin A_{k+1}} = 0, \end{array}$$

where the absolute value is performed componentwise. We let the solution of this LP be y_{k+1} and redefine $A_{+,k} = \{i : [y_{k+1}]_i > 0\} \cup \mathcal{E}$. We then set $x_{k+1} = x_k + \Delta x$ and $k = k + 1$ and repeat from the beginning.

Note that for the next iteration, the constraint gradients corresponding to the equality constraints together with inequality constraints corresponding to the positive components of y_{k+1} must be linearly independent. This is required for the system defining (CorrectStep) to be nonsingular. The constraints of (JumpLP), the approximate KKT conditions, define a polytope for the feasible set of multipliers. If the solution is on a vertex, then under the assumptions we make, the positive components of the corresponding multiplier solution corresponds to a linearly independent set of constraint gradients. Thus a problem can arise only when the solution to (JumpLP) is found on a face of the polytope. For the time being, we assume that the solution to (JumpLP) is on a vertex and the constraint gradients $\{\nabla_x c_i(x_k, y_k) : [y_k]_i > 0\}$ are linearly independent. Note that this will always be the case if a simplex solver is used for (JumpLP). Even otherwise, however, there are a number of reasons to believe that, in practice, a vertex solution to the LP corresponding to linearly independent constraint gradients for positive components of y_{k+1} will be found. We will revisit this topic in section 5, where we discuss this issue in more detail.

The basic steps of the procedure are summarized in Algorithm 1. A more detailed description of our implementation is given in section 5.

4. Convergence of the predictor-corrector path-following algorithm.

4.1. Preliminaries. We shall use the following results throughout the convergence theory.

LEMMA 4.1. *If the SSOSC and the MFCQ hold at* $(x^*(t), y^*)$ *for some* $y^* \in \Lambda(x^*(t), t)$*, then for* (x, y) *sufficiently close to* $(x^*(t), y^*)$*, there exist constants* $C_1(t) > 0$ *and* $C_2(t) > 0$ *such that it holds that*

$$C_1(t)\theta(x, y, t) \leq \eta(x, y, t) \leq C_2(t)\theta(x, y, t).$$

Proof. See, e.g., Wright [37, Theorem A.1]. ∎

The next lemma verifies that the active set estimate $\mathcal{A}_\gamma(x, y, t)$, given in (3.1) is accurate.

Algorithm 1 Predictor Corrector Sensitivity Path-Following Method.

Input: Δt, initial x, y such that $\{\nabla_x c_i(x,0)\}_{\{i\in\mathcal{I}:y_i>0\}\cup\mathcal{E}}$ is linearly independent.

Define γ satisfying $0<\gamma<1$.
Estimate A using (3.1) evaluated at $(x,y,0)$ and define A_+
while $t<1$ **do**
  Solve (CorrectStep) for $(\Delta_c x, \Delta_+ y)$.
  Define $\Delta_c y\in\mathbb{R}^m$ such that $[\Delta_c y]_{A_+}=\Delta_+ y$ and zero otherwise.
  Solve (QPPredict) for $(\Delta_p x, \Delta_p y)$.
  Set $(\Delta x,\Delta y)=(\Delta_p x,\Delta_p y)+(\Delta_c x,\Delta_c y)$.
  Compute $A=\mathcal{A}_\gamma(x+\Delta x, y+\Delta y, t+\Delta t)\cup\mathcal{E}$.
  Solve (JumpLP) to redefine y.
  Let $A_+=\{i:[y]_i>0\}\cup\mathcal{E}$.
  Set $t=t+\Delta t$, $x=x+\Delta x$.
end while

LEMMA 4.2 (see [13, Theorem 3.7]). *For all x,y such that $\theta(x,y,t)$ is sufficiently small, $\mathcal{A}_\gamma(x,y,t)=\mathcal{A}(x^*(t))$.*

LEMMA 4.3 (see [21, Lemma 3]). *Given matrices Q^* and P^*, where Q^* is symmetric, suppose that*

$$w^TQ^*w\ge\alpha\|w\|^2,\ \textit{whenever}\ P^*w=0,\ w\in\mathbb{R}^n.$$

Then given any $\zeta>0$, there exists $\sigma>0$ and neighborhoods $\mathcal{P}$ of P^* and $\mathcal{Q}$ of Q^* such that

$$v^T\left(Q+\frac{1}{\rho}P^TP\right)v\ge(\alpha-\zeta)\|v\|^2$$

for all $v\in\mathbb{R}^n$, $0<\rho\le\sigma$, $P\in\mathcal{P}$, and $Q\in\mathcal{Q}$.

The next theorem summarizes the sensitivity results that hold under the MFCQ and the CRCQ that are the theoretical foundation for the predictor-corrector algorithm we have formulated.

THEOREM 4.4. *Let f and c be twice continuously differentiable in t and x near $(x^*(t_0),t_0)$, and let the MFCQ and the GSSOSC hold at $x^*(t_0)$.*

1. *The solution $x^*(t)$ is continuous in a neighborhood of $x^*(t_0)$ and the solution function $x^*(t)$ is directionally differentiable, i.e.,*

$$x^*(t_0+\Delta t)=x^*(t_0)+\delta x^*(x^*(t_0),t_0,\Delta t)+o(|\Delta t|),\tag{4.1}$$

where $\delta x^*(x^*(t_0),t_0,\Delta t)$ is the directional derivative of $x^*(t)$ with respect to t at t_0 scaled by Δt, i.e.,

$$\delta x^*(x^*(t_0),t_0,\Delta t)=\lim_{\epsilon\downarrow 0}\frac{x^*(t_0+\epsilon)-x^*(t_0)}{\epsilon}\Delta t.$$

2. *Moreover, for each t in a neighborhood of t_0 and direction Δt, there exists a multiplier $\hat{y}\in\Lambda(x^*,t_0)$ such that the directional derivative uniquely solves*

the following convex quadratic program:
(SensitivityQP)

$$\begin{array}{rl} \min_{\Delta x} & \Delta x^T \nabla_{xt} L(x^*(t_0), \hat{y}, t_0)\Delta t + \frac{1}{2}\Delta x^T H(x^*(t_0), \hat{y}, t_0)\Delta x \\ \textit{subject to} & \nabla_x c_i(x^*(t_0), t_0)^T \Delta x + \nabla_t c_i(x^*(t_0), t_0)\Delta t = 0, \\ & \qquad i \in \mathcal{A}_+(x^*(t_0), \hat{y}, t_0) \cup \mathcal{E}, \\ & \nabla_x c_i(x^*(t_0), t_0)^T \Delta x + \nabla_t c_i(x^*(t_0), t_0)\Delta t \geq 0, \\ & \qquad i \in \mathcal{A}_0(x^*(t_0), \hat{y}, t_0). \end{array}$$

We denote the solution set of this program as

$$(\delta x^*(x^*(t_0), t_0, \Delta t), \delta Y^*(x^*(t_0), \hat{y}, t_0, \Delta t))$$

(where $\delta x^*(x^*(t_0), t_0, \Delta t)$ is a singleton). Note that $\delta x^*(x^*(t_0), t_0, \Delta t)$ does not depend on $\hat{y}$ (if there are multiple $\hat{y}$ satisfying the conditions of this part of the Theorem), but $\delta Y^*(x^*(t_0), \hat{y}, t_0, \Delta t)$ may.

3. *If, in addition, the CRCQ holds, then the multiplier values $\hat{y}$ at which the QP* (SensitivityQP) *must be evaluated can be found as a solution of the following linear program:*

$$\text{(SensitivityLP)} \quad \begin{array}{rl} \min_y & y^T \nabla_t c(x^*(t_0), t_0)\Delta t \\ \textit{subject to} & \nabla_x f(x^*(t_0), t_0) - \nabla_x c(x^*(t_0), t_0) y = 0, \\ & y_{\mathcal{I}} \geq 0, \\ & c_i(x^*(t_0), t_0) y_i = 0 \textit{ for all } i \in \{1, \ldots, m\}. \end{array}$$

We denote its solution set as $\hat{Y}(x^*(t_0), t_0)$. Note that this set is independent of Δt as long as $\Delta t > 0$.

4. *Moreover, the set $\{\nabla_x c_i(x^*(t_0), t_0)\}_{i \in \mathcal{A}_+(x^*(t_0), \hat{y}, t_0) \cup \mathcal{E}}$ is linearly independent for some $\hat{y} \in \hat{Y}(x^*(t_0), t_0)$; specifically, all extreme points of $\hat{Y}(x^*(t_0), t_0)$ satisfy this condition.*
5. *The primal-dual solution set $K(t_0)$ is proto-differentiable at $(x^*(t_0), \hat{y})$ for any $\hat{y} \in \hat{Y}(x^*(t_0), t_0)$, and the outer graphical derivative is the solution set of* (SensitivityQP)*, i.e.,*

$$DK(t_0 | x^*(t_0), \hat{y})\Delta t = (\delta x^*(x^*(t_0), t_0, \Delta t), \delta Y^*(x^*(t_0), \hat{y}, t_0, \Delta t)).$$

This implies, in particular, that, for any $\delta y^* \in \delta Y^*(x^*(t_0), \hat{y}, t_0, \Delta t)$

$$\textit{dist}(\hat{y} + \delta y^*, \Lambda(x^*(t_0 + \Delta t), t_0 + \Delta t)) = o(|\Delta t|). \tag{4.2}$$

Proof. Parts 1–3 appear as [24, Theorem 5] and [30, Theorems 1-2].

Part 4 follows from the proof of [27, Theorem 2.2]

Part 5 follows from [28, Proposition 2.5.1], where it can be seen that [28, (2.34)] are the optimality conditions of (SensitivityQP), if we consider that the problem is independent of any nonlinear perturbation w, and let $v_1' = \nabla_{xt} f(x, t)\Delta t$ and $v_2' = \nabla_t c(x, t)\Delta t$. The implication (4.2) follows from the definition of proto-differentiability, e.g., in the notation of (1.4), let $v' = 1$, reparametrize $v(s)$ to be $t_0 + \Delta t$, and then since $(\delta x^*(x^*(t_0), t_0, \Delta t), \delta Y^*(x^*(t_0), \hat{y}, t_0, \Delta t))$ is the directional derivative of $K(x, y, t)$, it holds that, for any $\delta y^* \in \delta Y^*(x^*(t_0), \hat{y}, t_0, \Delta t)$, and for $\delta x^*(x^*(t_0), t_0, \Delta t)$, the corresponding selection $w(s) = (x^*(s), y^*(s))$ satisfies

$$w(s) - w(0) = (\delta x^*(x^*(t_0), t_0, \Delta t)), \delta y^*) + \Delta t \alpha(\Delta t),$$

where $\alpha(\Delta t) \to 0$ as $\Delta t \to 0$. ∎

4.2. Convergence of Algorithm 1. We make the following assumptions.

Assumption 1. The functions $f(x,t)$ and $c(x,t)$ are two times Lipschitz continuously differentiable for all x and $t \in [0,1]$ with respect to both x and t.

Assumption 2. There exists a continuous primal solution path $x^*(t)$ to (1.1) for $t \in [0,1]$.

Hereafter, every result, unless otherwise noted, is with respect to a particular continuous such path $x^*(t)$.

Assumption 3. The CRCQ, the MFCQ, and the GSSOSC hold for all $x^*(t)$, $t \in [0,1]$.

LEMMA 4.5 (see [15]). *$\Lambda(x^*(t),t)$ is bounded for all t.*

Note that since the KKT conditions are linear in y, this implies that $\Lambda(x^*(t),t)$ is a closed convex polytope for any given $x^*(t)$ and t.

LEMMA 4.6. *There exists a B such that for every $\hat{y} \in \hat{Y}(x^*(t),t)$ for all t, $\|\hat{y}\| \le B$.*

Proof. Suppose there is a sequence $(\hat{y}(t_k),t_k)$ with $\hat{y}(t_k) \in \hat{Y}(x^*(t),t)$ such that $\|\hat{y}(t_k)\| \ge k$. But since t is in a compact set, there exists a convergent subsequence and a cluster point t^*. However, this implies that $\|\hat{y}(t^*)\| = \infty$, which is impossible by Lemma 4.5. ∎

LEMMA 4.7. *Consider the QP* (SensitivityQP) *evaluated at a $\hat{y}(t)$ that is an extreme point of $\hat{Y}(x^*(t),t)$ satisfying Part* 4 *of Theorem* 4.4. *There exists a B_2 such that for all $t \in [0,1]$, there exists a $\delta y^*(t) \in \delta Y^*(x^*(t),\hat{y}(t),t)$, the dual solution set to* (SensitivityQP)*, that is bounded by B_2.*

Proof. For any t, we can obtain a set of constraints active (or equality) at $\delta x^*(t)$ with linearly independent gradients. Then $\delta x^*(t)$ is also a solution of the QP with the rest of the constraints removed, for which the LICQ now holds, and thus there is a bounded subset of $\delta Y^*(x^*(t),\hat{y}(t),t)$. Now suppose that there exists (t_k) such that for all $\delta y^*(t_k) \in \delta Y^*(x^*(t_k),\hat{y}(t_k),t_k)$ it holds that $\|\delta y^*(t_k)\| \ge k$. But this implies that there exists a cluster point t^* such that $\|\delta y(t^*)^*\| = \infty$ for all $\delta y^*(t^*) \in \delta Y^*(x^*(t^*),\hat{y}(t^*),t^*)$, and this is impossible. ∎

We are now ready to present the main result with regards to the predictor-corrector step.

THEOREM 4.8. *If $\theta(x_k,y_k,t_k)$, $\|y_k - \hat{y}\|$ and Δt are sufficiently small for some $\hat{y} \in \hat{Y}(x^*(t),t_k)$ satisfying the condition in part* 4 *of Theorem* 4.4*, and $A_{+,k} = \mathcal{A}_+(x^*(t_k),\hat{y},t_k) \cup \mathcal{E}$, then consider the point $(x_k+\Delta x, y_k+\Delta y)$, where $(\Delta x,\Delta y) = (\Delta_p x + \Delta_c x, \Delta_p y + \Delta_c y)$ is defined as follows:*

- *$(\Delta_c x, \Delta_+ y)$ solves* (CorrectStep)*,*
- *$\Delta_c y \in \mathbb{R}^m$ is such that $[\Delta_c y]_{A_{+,k}} = \Delta_+ y$ and $[\Delta_c y]_i = 0$ for $i \in \{1,\dots,m\} \setminus A_{+,k}$, and*
- *$\Delta_p x$ solves* (QPPredict) *with any associated dual solution $\Delta_p y$.*

This primal-dual point $(x_k+\Delta x, y_k+\Delta y)$ satisfies $\eta(x_k+\Delta x, y_k+\Delta y, t_k+\Delta t) \le \eta(x_k,y_k,t_k)^{1+\gamma}$, where γ is the constant appearing in the active set estimate (3.1).

The next result states that given a primal-dual point (x_k,y_k) sufficiently close to the primal-dual solution set $(x^*(t),\Lambda(x^*(t),t)$, the solution of (JumpLP) yields a good multiplier approximation from which to calculate the next predictor step.

THEOREM 4.9. *For all ϵ and Δt, there exists a ν such that if*

$$dist((x,y),(x^*(t),\Lambda(x^*(t),t))) \le \nu,$$

then the solution $\bar{y}$ *to*

$$
\begin{aligned}
&\min_{\bar{y}} \quad \bar{y}\nabla_t c(x,t)\Delta t \\
&\textit{subject to} \quad -|\nabla L(x,y,t)| \\
&\qquad \le \nabla f(x,t) + \sum_{i\in\mathcal{A}(x^*(t))\cup\mathcal{E}} \nabla_x c_i(x,t)\bar{y}_i \\
&\qquad \le |\nabla L(x,y,t)|
\end{aligned} \tag{4.3}
$$

satisfies dist$(\bar{y},\hat{Y}(x^*(t),t)) \le \epsilon$.

The first theorem guarantees that the primal-dual solution for the corrector-predictor problems result in a step that produces a point that is arbitrarily close to the solution to the parametric NLP evaluated at $t+\Delta t$, given an original iterate (x_k, y_k) sufficiently close to $(x^*(t),\hat{y})$. It implies that the primal dual point is within the ball of local superlinear contraction and the corrector step results in a point much closer to the solution of the problem for the current value of the parameter. Recall that the predictor (QPPredict) is a first-order estimate of the primal-dual solution path, and by coupling it with the corrector (CorrectStep), we can ensure that the solution estimate tracks the solution, in the sense of being able to produce a point that is an accurate enough approximation of the appropriate primal-dual solution such that all subsequent estimates are as close as desired by some predetermined amount.

The second theorem states that if the active set is estimated correctly, which follows from Lemma 4.2, if the point at which the LP (JumpLP) is evaluated is sufficiently close to $(x^*(t+\Delta t),\Lambda(x^*(t+\Delta t),t+\Delta t)$, then the multiplier solution can be made arbitrarily close to the set $\hat{Y}(x^*(t+\Delta t))$. By assumption, we can make the starting point sufficiently close to the primal dual solution $(x^*(t),\hat{y})$, so as to (possibly also by decreasing Δt) to make the solution to the predictor-corrector subproblems arbitrarily close to $(x^*(t+\Delta t),\Lambda(x^*(t+\Delta t),t+\Delta t))$. Theorem 4.9 implies that the final multiplier estimate can get within any desired distance to $\hat{Y}(x^*(t+\Delta t),t+\Delta t)$. This suggests that the new $x_k+\Delta x$ can be in the neighborhood for which the CRCQ applies at $t+\Delta t$, and together with the multiplier returned by (JumpLP) yield a primal-dual point that satisfies the conditions of Theorem 4.8 for the problem at $t+\Delta t$. The argument then repeats itself at $t+\Delta t$, and so on, until reaching $t=1$. For any t to which we apply the argument, we can make the solution estimate at the initial $t=0$ close enough to yield all the desired results across all the subsequent $(0,t]$ up until the current value of the parameter.

Conceptually, the argument can be thought of as presenting results that are slightly stronger than basic $\epsilon-\delta$ theory. Specifically, all primal-dual points can be arbitrarily close to the desired points at the subsequent iteration, and in particular the error with respect to the primal dual solution can be made to always be bounded by the previous error. However, nothing stronger than that can be claimed. In particular, unless the radius of the Newton–Kantarovich ball of quadratic convergence is uniformly bounded from below across t (i.e., there exists some $\tilde{c}_1$ independent of t such that if $\operatorname{dist}((x,y),(x^*(t),\hat{y})) \le \tilde{c}_1$, then Newton's method initiated at (x,y) applied to the equality constrained NLP (4.4) results in a superlinearly/quadratically convergent sequence of iterates converging to $(x^*(t),\hat{y})$), the neighborhood for which the CRCQ applies is also bounded across t, and other such conditions, finite termination without occasional reliance on a globalized algorithm cannot be ensured. However, we expect that for many well-formulated problems, it can be expected that the required conditions do hold. Since little is known about regions of applicability of constraint qualifications and local convergence for Newton methods for NLPs, however, no precise statements can be made to that effect. The nature and limitations of the proof theory presented here also apply to other algorithms implementing path-following for parametric NLP problems, e.g., [9, 39]. So these techniques seem to be standard for this class of problems and algorithms. An exception is [10], which proves a uniform strong regularity result which permits a proof of finite termination upon finding an initial solution. This was proved in the case of strong regularity for the problem (in this case a variational inequality) at all values of the parameter, however, and the question as to whether something similar can be shown for the setting assumed in this paper, or, alternatively, what the weakest conditions for such a result would be, can be a topic for future research.

A technical point is the following: We can get the solution of the algorithm at any given iteration arbitrarily close to the solution of the PNLP at $t_k + \Delta t$ if we make the previous point close enough to the solution at t_k. This means that when we use O notation, it is all with respect to a specific k (specifically as the base point approaches the solution at the base parameter value). This is different from standard NLP algorithms and global convergence proofs, which are concerned with limit points of the sequence (x_k, y_k) as $k \to \infty$.

4.3. Proofs of results. In this section we prove the main two results of this paper.

Proof of Theorem 4.8. The system (CorrectStep) is the Newton–Lagrange system for solving the NLP,

$$\begin{array}{rl} \min_x & f(x, t_k), \\ \text{subject to} & c_i(x, t_k) = 0,\ i \in \mathcal{A}_+(x^*(t_k), \hat{y}, t_k) \cup \mathcal{E}, \end{array} \tag{4.4}$$

for which $x^*(t_k)$ is a stationary solution. By the linear independence of $\{\nabla c_i(x^*(t_k), t_k)\}_{i \in \mathcal{A}_+(x^*(t_k),\hat{y},t_k)\cup\mathcal{E}}$ (thus the MFCQ holds for (4.4)) and the SSOSC, we can invoke [32, Theorem 2.2] to conclude that the solution is isolated. Thus the Newton–Lagrange step (CorrectStep) is associated with local convergence to $(x^*(t_k), \hat{y})$ uniquely.

Denoting the solution to (CorrectStep) as $(\Delta_c x, \Delta_c y)$, we have, from the usual Newton local convergence properties,

$$\|(x_k + \Delta_c x - x^*(t_k), y_k + \Delta_c y - \hat{y})\| = O(\|(x_k - x^*(t_k), y_k - \hat{y})\|^2), \tag{4.5}$$

and

$$\|(\Delta_c x, \Delta_c y)\| = O(\|(x_k - x^*(t_k), y_k - \hat{y})\|). \tag{4.6}$$

By the GSSOSC, subproblem (SensitivityQP) is strongly convex and has a unique solution. By (4.6) and Lemma 4.3 it holds that (QPPredict) is also strongly convex and has a unique primal solution for $\|(x_k - x^*(t_k), y_k - \hat{y})\|$ sufficiently small.

We can now write the optimality conditions of (QPPredict) as, using $\nabla_x c(x, t_k) = \nabla_x c(x, t_k + \Delta t)$ and $H(x, y, t) = H(x, y, t + \Delta t)$,

$$\begin{aligned} &(\nabla_x f(x_k, t_k + \Delta t) - \nabla_x c(x_k, t_k + \Delta t) y_k - \nabla_x f(x_k, t_k) + \nabla_x c(x_k, t_k) y_k) \\ &\qquad\qquad + H(x_k, y_k, t_k + \Delta t)\Delta_p x - (\nabla_x c(x_k, t_k) + \nabla^2_{xx} c(x_k, t_k)\Delta_c x)\Delta_p y = 0, \\ &(\nabla_x c_i(x_k, t_k + \Delta t) + \nabla^2_{xx} c_i(x_k, t_k)\Delta_c x)^T \Delta_p x + \nabla_t c_i(x_k, t_k)\Delta t = 0,\ i \in A_{+,k}, \\ &(\nabla_x c_i(x_k, t_k + \Delta t) + \nabla^2_{xx} c_i(x_k, t_k)\Delta_c x)^T \Delta_p x + \nabla_t c_i(x_k, t_k)\Delta t \in \mathcal{N}([\Delta_p y]_i), \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad i \in A_k \setminus A_{+,k}. \end{aligned} \tag{4.7}$$

Now, using the properties of f and c as functions with respect to t,

$$(\nabla_x f(x_k,t_k+\Delta t) - \nabla_x c(x_k,t_k+\Delta t)y_k - \nabla_x f(x_k,t_k) + \nabla_x c(x_k,t_k)y_k) \\ = \nabla_{xt}L(x_k,y_k,t_k)\Delta t,$$

we can rewrite (4.7) as

(4.8)

$$\begin{aligned} &\nabla_{xt}L(x_k,y_k,t_k)\Delta t + H(x_k,y_k,t_k+\Delta t)\Delta_p x - (\nabla_x c(x_k,t_k) \\ &\qquad\qquad + \nabla^2_{xx}c(x_k,t_k)\Delta_c x)\Delta_p y = 0, \\ &(\nabla_x c_i(x_k,t_k) + \nabla^2_{xx}c_i(x_k,t_k)\Delta_c x)^T\Delta_p x + \nabla_t c_i(x_k,t_k)\Delta t = 0,\ i \in A_{+,k}, \\ &(\nabla_x c_i(x_k,t_k) + \nabla^2_{xx}c_i(x_k,t_k)\Delta_c x)^T\Delta_p x + \nabla_t c_i(x_k,t_k)\Delta t \in \mathcal{N}([\Delta y]_i),\ i \in A_k \setminus A_{+,k}. \end{aligned}$$

We may consider this system as a perturbation of the optimality conditions of (SensitivityQP). Consider any $\overline{\delta x}$ and an associated dual $\overline{\delta y}$ solution to (4.8). We shall apply the upper Lipschitz continuity of solutions subject to perturbations given for a QP satisfying the SSOSC and the MFCQ in [32, Theorem 4.2]. In particular, in the notation of the theorem, for the base perturbation γ_0, $\nabla_{\delta x}\tilde{f}(\delta x;\gamma_0) = \nabla_{xt}L(x^*(t),\hat{y},t_k)\Delta t + H(x^*(t),\hat{y},t_k+\Delta t)\delta x$, and for the current point, considered at γ_1, $\nabla_{\delta x}\tilde{f}(\delta x;\gamma_1) = \nabla_{xt}L(x_k,y_k,t_k)\Delta t + H(x_k,y_k,t_k+\Delta t)\delta x$, and the constraint $\tilde{c}$ as $\tilde{c}(\delta x;\gamma_0) = (\nabla_x c_i(x^*(t),t_k))^T\delta x + \nabla_t c_i(x^*(t),t_k)\Delta t$ and $\tilde{c}(\delta x;\gamma_1) = (\nabla_x c_i(x_k,t_k) + \nabla^2_{xx}c_i(x_k,t_k)\Delta_c x)^T\delta x + \nabla_t c_i(x_k,t_k)\Delta t$.

We see from the conclusion of [32, Theorem 4.2] that the solution $(\overline{\delta x},\overline{\delta y})$ to (QPPredict) satisfies

(4.9)

$$\begin{aligned} &\|\overline{\delta x} - \delta x^*\| + \operatorname{dist}(\overline{\delta y},\delta Y^*(x^*(t_k),\hat{y},t_k)) \\ &\le \left\| \begin{pmatrix} \nabla_{xt}L(x_k,y_k,t_k)\Delta t - \nabla_{xt}L(x^*(t_k),\hat{y},t_k)\Delta t \\ + (H(x_k,y_k,t_k+\Delta t) - H(x^*(t_k),\hat{y}(t),t_k+\Delta t))\overline{\delta x} \\ - (\nabla_x c(x^*(t_k),t_k) - \nabla_x c(x_k,t_k) - \nabla^2_{xx}c(x_k,t_k)\Delta_c x)\overline{\delta y} \\ ((\nabla_x c(x_k,t_k) + \nabla^2_{xx}c(x_k,t_k)\Delta_c x)^T - \nabla_x c(x^*(t_k),t_k)^T)\overline{\delta x} \\ + \nabla_t c(x_k,t_k)\Delta t - \nabla_t c(x^*(t_k),t_k)\Delta t \end{pmatrix} \right\|. \end{aligned}$$

By the fact that $\nabla_t c(x,t)$ and $\nabla_{xt}f(x,t)$ is a constant, we have that

$$\nabla_{xt}L(x_k,y_k,t_k)\Delta t - \nabla_{xt}L(x^*(t_k),\hat{y},t_k)\Delta t = 0 \text{ and} \\ \nabla_t c(x_k,t_k)\Delta t - \nabla_t c(x^*(t_k),t_k)\Delta t = 0.$$

Given the two-times Lipschitz continuity of c and f we can obtain

(4.10)

$$\begin{aligned} &\|(H(x_k,y_k,t+\Delta t) - H(x^*(t_k),\hat{y},t_k+\Delta t))\overline{\delta x}\| \\ &\quad = \Big\|((\nabla^2_{xx}f(x_k,t_k+\Delta t) - \nabla^2_{xx}f(x^*(t_k),t+\Delta t) \\ &\qquad - \sum[y_k]_i(\nabla^2_{xx}c_i(x_k) - \nabla^2_{xx}c_i(x^*(t_k))) - \sum([y_k]_i - [\hat{y}]_i)\nabla^2_{xx}c_i(x^*(t_k)))\overline{\delta x}\Big\| \\ &\quad \le ((C_L + B + \|y_k - \hat{y}\|)\|x_k - x^*(t_k)\| + B\|y_k - \hat{y}\|)\|\overline{\delta x}\|, \end{aligned}$$

where C_L is an upper bound for the Lipschitz constant for function and the first and second derivatives of f and c and B is an upper bound on $\hat{y}$ by Lemma 4.6. Using Taylor's theorem and (4.5), (4.6),

$$\begin{aligned} &\|(\nabla_x c(x^*(t_k),t_k) - \nabla_x c(x_k,t_k) - \nabla^2_{xx}c(x_k,t_k)\Delta_c x)\overline{\delta y}\| \\ &\quad = \|(\nabla_x c(x^*(t_k),t_k) - \nabla_x c(x_k+\Delta_c x,t_k))\overline{\delta y} + O(\|\Delta_c x\|^2\|\overline{\delta y}\|) \\ &\quad = O(\|(x_k - x^*(t_k), y_k - \hat{y})\|^2\|\overline{\delta y}\|), \end{aligned} \tag{4.11}$$

and similarly,

$$\|(\nabla_x c(x_k,t_k)^T + \nabla^2_{xx} c(x_k,t_k)\Delta_c x - \nabla_x c(x^*(t_k),t_k)^T)\overline{\delta x}\| = O(\|(x_k - x^*(t_k))\|^2 \|\overline{\delta x}\|). \tag{4.12}$$

Let δy^* be an element of $\delta Y^*(x^*(t_k),\hat{y},t_k)$ satisfying Lemma 4.7.

From applying (4.10), (4.11), and (4.12) to (4.9) and then applying the triangle inequality to write $\|\overline{\delta x}\| \leq \|\overline{\delta x} + \delta x^*\| + \|\delta x^*\|$ and $\|\overline{\delta y}\| \leq \|\overline{\delta y} + \delta y^*\| + \|\delta y^*\|$, we can deduce

$$\begin{aligned} &\|\overline{\delta x} - \delta x^*(x^*(t_k),t_k)\| + \|\overline{\delta y} - \delta y^*\| \\ &\quad = O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|)(\|\overline{\delta x} - \delta x^*\| + \|\delta x^*\|) \\ &\qquad + O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|^2)(\|\delta x^*\| + \|\overline{\delta y} - \delta y^*\| + \|\delta y^*\|)). \end{aligned} \tag{4.13}$$

By taking $\|(x_k,y_k) - (x^*(t_k),\hat{y}(t_k))\|$ sufficiently small, we can ensure that all terms above of the form $O(\|(x_k,y_k) - (x^*(t_k),\hat{y}(t_k))\|)$ are less than one half. We can then subtract $\frac{1}{2}\|\overline{\delta x} - \delta x^*\| + \frac{1}{2}\|\overline{\delta y} - \delta y^*\|$ from both sides of (4.13) and then double both sides of the resulting equation to get

$$\begin{aligned} \|\overline{\delta x} - \delta x^*\| + \|\overline{\delta y} - \delta y^*\| = O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|^2)(\|\delta x^*\| + \|\delta y^*\|) \\ + O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|)\|\delta x^*\|. \end{aligned} \tag{4.14}$$

Theorem 4.4 parts 1 and 5 imply that

$$\|x^*(t_k) + \delta x^*(x^*(t_k),t_k) - x^*(t_k+\Delta t)\| + \|\hat{y} + \delta y^* - y^*(t_k+\Delta t)\| = o(\Delta t), \tag{4.15}$$

where $y^*(t_k+\Delta t)$ satisfies

$$\|\hat{y} + \delta y^* - y^*(t_k+\Delta t)\| = \operatorname{dist}(\hat{y} + \delta y^*, \Lambda(x^*(t_k+\Delta t), t_k+\Delta t)).$$

Furthermore, it holds that as $\Delta t \to 0$, $\delta x^*(x^*(t_k),t_k,\Delta t) \to 0$, and by Lemma 4.7 it holds that $\|\delta y^*\| \leq B_2$. Let $E(\Delta t) = \|\delta x^*(x^*(t_k),t_k,\Delta t)\|$.

Finally, using Lemma 4.1, we get

$$\begin{aligned} &\eta(x_k+\Delta x, y_k+\Delta y, t+\Delta t) \\ &\quad \leq C_2(t_k+\Delta t)\big(\|x_k + \Delta_c x + \overline{\delta x} - x^*(t+\Delta t)\| \\ &\qquad\qquad + \operatorname{dist}(x_k + \Delta_c y + \overline{\delta y}, \Lambda(x^*(t_k+\Delta t), t_k))\big) \\ &\quad \leq C_2(t_k+\Delta t)\,(\|x_k + \Delta_c x - x^*(t_k)\| + \|y_k + \Delta_c y - \hat{y}\| \\ &\qquad\qquad + \|x^*(t_k) + \delta x^*(x^*(t_k),t_k) - x^*(t_k+\Delta t)\| \\ &\qquad\qquad + \|\hat{y} + \delta y^* - y^*(t_k+\Delta t)\| \\ &\qquad\qquad + \|\overline{\delta x} - \delta x^*(x^*(t_k),t_k)\| + \|\overline{\delta y} - \delta y^*\|\big) \\ &\quad = O(\|(x_k - x^*(t_k), y_k - \hat{y})\|^2) + o(\Delta t) + O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|^2) \\ &\qquad + O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|)E(\Delta t) \\ &\quad = O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|^2) + o(\Delta t) + O(\|(x_k,y_k) - (x^*(t_k),\hat{y})\|)E(\Delta t) \\ &\quad = O(\eta(x_k,y_k,t)^2) + o(\Delta t) + O(\eta(x_k,y_k,t))E(\Delta t). \end{aligned} \tag{4.16}$$

Thus, by making Δt sufficiently small, we get the desired result for sufficiently small $\|(x_k,y_k) - (x^*(t_k),\hat{y})\|$. ∎

Proof of Theorem 4.9. Let the constant vector r be defined to be $r \equiv \nabla_t c(x,t)\Delta t$. Recall that by the MFCQ and Gauvin [15], it holds that the set $\Lambda(x^*(t),t)$ is bounded, and since the KKT conditions are linear with respect to y, it is also closed and convex, and thus compact, and is defined as a polytope.

Let $\{\tilde{y}_j\}_{j\in\{1,\dots,J\}} = \tilde{Y}(x^*(t),t)$ be the set of extreme points of this polytope. It holds that for each $\tilde{y}_j$, for the set $I(\tilde{y}_j) \subseteq \mathcal{I}$ such that $[\tilde{y}_j]_{I(\tilde{y}_j)} > 0$, the corresponding set of constraint gradients $\{\nabla c_i(x^*(t),t)\}_{i\in I(\tilde{y}_j)\cup\mathcal{E}}$ is linearly independent. Moreover, by CRCQ there exists a ν_1 such that $\{\nabla c_i(\tilde{x},t)\}_{i\in I(\tilde{y}_j)\cup\mathcal{E}}$ is linearly independent for $\tilde{x}$ such that $\|\tilde{x}-x^*(t)\| \le \nu_1$. By the implicit function theorem, for any $\tilde{\epsilon} > 0$ there exists ϵ_1 such that if J_ρ and g_ρ are perturbations of the objective and constraint gradients satisfying $\|J_\rho - \nabla_x c(x^*(t),t)\| \le \epsilon_1$ and $\|g_\rho - \nabla_x f(x^*(t),t)\| \le \epsilon_1$, it holds that $[J_\rho]_{I(\tilde{y}_j)\cup\mathcal{E}}[\tilde{y}_j(\rho)]_{I(\tilde{y}_j)\cup\mathcal{E}} = g_\rho$ with $[\tilde{y}_j(\rho)]_{\mathcal{I}} \ge 0$ and $\|\tilde{y}_j(\rho)-\tilde{y}_j\| \le \tilde{\epsilon}$.

By part 3 of Theorem 4.4 we have that $r^T\hat{y} < r^T y^*$ for all $y^* \in \Lambda(x^*(t),t)$ with $y^* \notin \hat{Y}(x^*(t),t)$, $\hat{y} \in \hat{Y}(x^*(t),t)$. Therefore for all ϵ_2, there exists some $\hat{\epsilon}_2$ such that for perturbations ρ of the extremal multipliers $\tilde{y}_j$ satisfying $\|\tilde{y}_j(\rho)-\tilde{y}_j\| \le \hat{\epsilon}_2$, it holds that

$$r^T\tilde{y}_{\hat{j}}(\rho) < r^T\tilde{y}_{\check{j}}(\rho) - \epsilon_2 \tag{4.17}$$

for all $\hat{j}$ such that $\tilde{y}_{\hat{j}} \in \hat{Y}(x^*(t),t) \cap \tilde{Y}(x^*(t),t)$ and $\check{j}$ such that $\tilde{y}_{\check{j}} \in \tilde{Y}(x^*(t),t) \setminus \hat{Y}(x^*(t),t)$.

Since $\nabla_x L(x,y)$, $\nabla_x f(x,t) - \nabla_x f(x^*(t),t)$ and $\nabla_x c(x,t) - \nabla_x c(x^*(t),t)$ are all $O(\theta(x,y,t))$, the constraints of (4.3) correspond to a set of perturbation of the stationarity conditions. Let us say they are all bounded by $C\theta(x,y,t)$.

Finally, choose ϵ as given in the statement of Theorem 4.9, i.e., satisfying the desired estimate $\operatorname{dist}(\bar{y},\hat{Y}(x^*(t),t) \le \epsilon$. Let $\epsilon_2 \ge \epsilon$, and a corresponding $\hat{\epsilon}_2 \le \frac{1}{4}\epsilon^2(2\max\{1,\|r\|_\infty\}\max\{\|y\|_\infty : y \in \Lambda(x^*(t),t)\})^{-1}$ such that (4.17) holds for $\|\tilde{y}_j(\rho) - \tilde{y}_j\| \le \hat{\epsilon}_2$.

Now choose $\hat{\epsilon}_1$ and then take some appropriate x and y as stated in the conditions of the Theorem, such that for $C\theta(x,y,t) \le \min(\nu_1,\hat{\epsilon}_1)$, for all y^α satisfying

$$-|\nabla_x L(x,y,t)| \le \nabla_x f(x,t) - \nabla_x c(x,t)y^\alpha \le |\nabla_x L(x,y,t)|, \quad y^\alpha \ge 0,$$

it holds that $\|y^\alpha - \sum_{j\in J}\alpha_j\tilde{y}_j\| \le \min(\hat{\epsilon}_2,\frac{\epsilon}{2})$ for some α_j with $\alpha_j \ge 0$ for $j \in \mathcal{I}$, $\sum_j \alpha_j = 1$.

Now let y^α be a solution to the (4.3), i.e., y^α is feasible and $r^T y^\alpha \le r^T y^\beta$ for all feasible y^β. Let α_j be defined as above with

$$\left\| y^\alpha - \sum_{j\in J}\alpha_j\tilde{y}_j \right\| \le \min\left(\hat{\epsilon}_2, \frac{\epsilon}{2}\right). \tag{4.18}$$

Consider any feasible y^β satisfying $\|y^\beta - \tilde{y}_{\hat{k}}\| \le \hat{\epsilon}_2$ for some $\tilde{y}_{\hat{j}} \in \hat{Y}(x^*(t),t) \cap \tilde{Y}(x^*(t),t)$.

Let $\hat{J}$ be such that if $\hat{j} \in \hat{J}$, then $\tilde{y}_{\hat{j}} \in \hat{Y}(x^*(t),t) \cap \tilde{Y}(x^*(t),t)$.

It holds that

$$
\begin{aligned}
& r^T y^\alpha \leq r^T y^\beta \\
(4.18) & \Longrightarrow r^T \sum_{j\in J} \alpha_j \tilde{y}_j - \|r\|_\infty \hat{\epsilon}_2 < r^T y^\beta \\
& \Longrightarrow r^T \sum_{j\in J} \alpha_j \tilde{y}_j - 2\|r\|_\infty \hat{\epsilon}_2 < r^T \tilde{y}_{\hat{j}} \\
& \Longrightarrow r^T \sum_{j\in J} \alpha_j \tilde{y}_j - \frac{\epsilon^2}{2} \left(\max\{\|y\|_\infty : y \in \Lambda(x^*(t),t)\}\right)^{-1} < r^T \tilde{y}_{\hat{j}} \\
& \Longrightarrow r^T \sum_{j\in \hat{J}} \alpha_j \tilde{y}_j + r^T \sum_{j\in J\setminus\hat{J}} \alpha_j \tilde{y}_j - \frac{\epsilon^2}{2} \left(\max\{\|y\|_\infty : y \in \Lambda(x^*(t),t)\}\right)^{-1} < r^T \tilde{y}_{\hat{j}} \\
& \Longrightarrow \epsilon_2 \sum_{j\in J\setminus\hat{J}} \alpha_j - \frac{\epsilon^2}{2} \left(\max\{\|y\|_\infty : y \in \Lambda(x^*(t),t)\}\right)^{-1} < 0 \\
& \Longrightarrow \epsilon_2 \sum_{j\in J\setminus\hat{J}} \alpha_j \left(\max\{\|y\|_\infty : y \in \Lambda(x^*(t),t)\}\right) < \frac{\epsilon^2}{2} \\
& \Longrightarrow \sum_{j\in J\setminus\hat{J}} \alpha_j \left(\max\{\|y\|_\infty : y \in \Lambda(x^*(t),t)\}\right) < \frac{\epsilon}{2},
\end{aligned}
$$

which together with (4.18) implies that $\|y^\alpha - \sum_{j\in\hat{J}} \alpha_j \tilde{y}_j\| \leq \epsilon$, proving the theorem. ◻

5. Discussion and practical implementation. Algorithm 1 in section 3.2 is a basic skeleton of a path-following algorithm for tracing a solution curve of a parametric optimization problem. The convergence results presented in the previous section show the fundamental theoretical results that demonstrate the efficacy of the steps in the procedure. This section discusses some of the issues that are relevant for a practical implementation.

To begin with, an initial point satisfying the requirements laid out needs to be generated. For generating this point, at $t = 0$, consider that we solve the standalone NLP, (1.1) evaluated at $t = 0$, using some globalized NLP solver, to obtain at least an approximate solution, i.e., one for which $\eta(x, y, 0)$ is close to zero, or the magnitude of tolerance one desires to track across the homotopy. By assumption the MFCQ and the CRCQ both hold at the solution of the problem. Thus any NLP solver that is globally convergent for NLPs satisfying these constraint qualifications is sufficient to perform this task. This is not a very stringent requirement; for example, augmented Lagrangian methods are known to be globally convergent for assumptions weaker than this [2, 17]. The appropriate y_0 can then be found by solving (JumpLP) evaluated at this primal-dual (approximate) solution.

As mentioned in section 4.2, it may happen that the $\epsilon - \delta$'s do not line up, i.e., that the distance to the solution required for Theorems 4.8 and 4.9 to apply at a particular point happens to be smaller than the one obtained from the previous iteration. Fundamentally, this means that to entirely avoid having to rerun a globalized solver one would need to a priori know the required balls of local convergence and neighborhoods relevant to the CQ across the homotopy in order to determine Δt as well as the required distance of the initial point to the solution at $t = 0$. In practice, however, we expect this to rarely be an issue, as in most contexts in which one would use a path-following optimization algorithm the differences in the NLPs across the possible values of t is not so large, so we expect relatively stable convergence and CQ neighborhood radii required for tracking. Note that this is a standard issue for parametric optimization algorithms, such as [9, 39]. Homotopy solvers rely on local properties of the NLP and also prove similar $\epsilon - \delta$ type results. As such they require an initial point sufficiently close to the solution as well as possibly a resolve if things break down.

Recall from the theoretical results that there are two important quantities that determine the success of the primary steps of the path-following algorithm, the value of Δt and the distance of the current point to the solution. Both have to be small enough. We may not know, if a complete iteration is a failure, which quantity was not sufficiently small. We may check required properties of individual steps, and through experimentation we can devise a reasonable procedure that seeks to diminish these two quantities, either by changing Δt or obtaining a closer point with a globalized solver, appropriately. Of course diminishing Δt is far less difficult and computationally expensive, so we naturally err toward reducing Δt and only revert to a globalized solver when necessary. In our experiments, we found that as long as the first point corresponding to the problem for $t = 0$ is sufficiently close to the solution, we did not need to revert to a globalized solver during the course of the homotopy, but had to modify Δt at times.

At each iteration on the first step, the algorithm computes (CorrectStep). We assume the original point to be in the radius of local convergence for the problem, and so the resulting step would result in a point that corresponds to at least a superlinear reduction in the distance to the solution. We perform a simple check that the initial point was within the radius of local convergence by looking at the sign of the resulting multipliers. In particular, if there exists an $i \notin \mathcal{E}$ such that $[y_{A_{+,k}} + \Delta_+ y]_i < 0$, then this suggests that we are not sufficiently close to a primal-dual solution, and so we revert to an external independent globalized optimization software to find a new primal-dual point (x_k, y_k) closer to $(x^*(t), \Lambda(x^*(t)))$ and satisfying the needed conditions for the multiplier. One can also check the optimality residual for superlinear contraction.

In addition, the step (QPPredict) must be evaluated at a point sufficiently close to the primal-dual solution to result in a step that is sufficiently close to the solution at $t + \Delta t$. Specifically, we check if any new constraints have become violated by too high a tolerance, or that the step was not a sufficiently accurate estimate of $(x^*(t + \Delta t), \Lambda(x^*(t + \Delta t), t + \Delta t))$. Specifically, if

$$\eta(x_k + \Delta x, y_k + \Delta y, t + \Delta t) > \max(\eta(x_k, y_k, t), \eta_{\text{tol}}), \tag{5.1}$$

we decrease Δt and solve (QPPredict) again. In this case, η_{tol} is some value that defines an approximate optimality tolerance. We found that if we were to solely enforce monotonic tracking, the algorithm would be forced to make ever tighter approximations, necessitating ever smaller values of Δt, which is inefficient and practically unnecessary.

Finally, we discuss the issue of generating a solution at the vertex and hence obtaining a multiplier whose positive components correspond to a set of linearly independent constraint gradients.

1. The problem can entirely be avoided by using a simplex solver for (JumpLP) which completely ensures that a vertex solution is sought. In practice, simplex methods perform very well, so this would be the recommended approach.
2. A nonvertex solution can arise only when the ratio of $\nabla_t c_i(x + \Delta x, t + \Delta t)$ to $\nabla_x c_i(x + \Delta x, t + \Delta t)$ is the same for two or more constraints. For nonlinear problems, this is extremely unlikely, and except for particularly contrived

Algorithm 2 Predictor Corrector Sensitivity Path-Following Method.

Input: t, x, y close to $(x^*(t), \hat{Y}(t))$ such that $\{\nabla_x c_i(x,t)\}_{\{i\in\mathcal{I}: y_i>0\}\cup\mathcal{E}}$ is linearly independent.

1: Define parameters γ satisfying $0<\gamma<1$ and $\iota_{\max}\in\mathbb{N}$.
2: Estimate A using (3.1) evaluated at $(x,y,0)$ and define A_+.
3: Set $\iota = 1$
4: **while** $t<1$ **do**
5: Solve (CorrectStep) for $(\Delta_c x, \Delta_+ y)$.
6: **if** for some $i\notin\mathcal{E}$, $[y_{A_+}+\Delta_+ y]_i<0$ **then**
7: Solve (1.1) at t for an approximate solution (x,y).
8: Set $\iota = 1$.
9: **end if**
10: Define $\Delta_c y\in\mathbb{R}^m$ such that $[\Delta_c y]_{A_+}=\Delta_+ y$ and zero otherwise.
11: Solve (QPPredict) for $(\Delta_p x, \Delta_p y)$.
12: Set $(\Delta x, \Delta y) = (\Delta_p x, \Delta_p y) + (\Delta_c x, \Delta_c y)$
13: **if** (5.1) does hold **then**
14: **if** $\iota\ge\iota_{\max}$ **then**
15: Solve (1.1) at t for an approximate solution (x,y).
16: Set $\iota = 1$.
17: Go to Line 5
18: **else**
19: Decrease Δt. Set $\iota = \iota + 1$.
20: Go to Line 11
21: **end if**
22: **end if**
23: **if** $\eta(x_k+\Delta x, y_k+\Delta y, t+\Delta t) < \eta(x_k,y_k,t)^{1+\gamma}$ (very good step) **then**
24: Increase Δt.
25: **end if**
26: Compute $A=\mathcal{A}_\gamma(x+\Delta x, y+\Delta y, t+\Delta t)\cup\mathcal{E}$.
27: Solve (JumpLP) to redefine y.
28: Let $A_+=\{i:[y]_i>0\}\cup\mathcal{E}$.
29: Set $t=t+\Delta t$, $x=x+\Delta x$.
30: **end while**

problems could hold for only a finite set of $x\in\mathbb{R}$. In inexact numerical arithmetic the probability of this occurring would be zero for any problems. For linear problems, it could happen that throughout the space some of these constraint gradients are proportional in this sense, but in this case they can be precomputed and the constraints appropriately regularized prior to running the path-following algorithm.

3. Finally, there is a procedure outlined on page 493 of [37] that finds an appropriate y that satisfies the linear independence of constraint gradients corresponding to positive multiplier values condition, starting from any feasible multiplier. The procedure requires the computation of a null-space, which can be done by a matrix factorization.

We outline our complete implementation with worst case safeguards in Algorithm 2. We found that, aside from finding the initial starting point, we never had to revert to a globalized solver in our experiments, but we did sometimes have to decrease Δt following a poor predictor step and then have to resolve (QPPredict). We expect in general that reverting to a globalized solver is unnecessary for most practical problems. However, we included all safeguards for completeness sake.

6. Numerical results.

6.1. Problem with degenerate constraints throughout and an active set change.

We consider problem (2.1) for $t \in [0, 1]$. Notice at the point wherein there is an active set change, $t = 0.5$, the vector $\nabla_t c(x, t)\Delta t$ is

$$\nabla_t c(x,t)\Delta t = \begin{pmatrix} -10 & 0 & 10 & 20 & 0 & -10 & -10 \end{pmatrix}^T \Delta t,$$

and thus the solution to (SensitivityLP), recalling that

$$\begin{aligned} \Lambda\left(x^*\left(\frac{1}{2}\right)\right) &= (0, y_1, y_2, y_3, y_4, y_5, y_6),\ y_1 + y_2 + y_3 + y_5 + y_6 \\ &= e^5,\ y_1 - y_3 - y_4 + \frac{1}{2}y_5 - \frac{1}{2}y_6 = 0,\ y_i \geq 0, \end{aligned}$$

is $(0, 0, 0, 0, 0, \frac{1}{2}e^5, \frac{1}{2}e^5)$, indicating that the last two constraints should be strongly active, and for $t \in [0.5, 1]$, the solution should trace these constraints.

Indeed we find that the algorithm successfully traces the solution and (JumpLP) performs the jump in the multipliers at $t = 0.5$. If, in the implementation of the algorithm, we turn (JumpLP) off and instead use the multiplier solution Δy from (CorrectStep) and (QPPredict), then the algorithm gets stuck at $t = 0.5$ and proceeds no further along the homotopy. We show the plots of the primal and dual variables in Figure 3. Note that x follows the true solution closely, y_2 and y_4 are always effectively zero, and there is a discontinuous jump halfway along the homotopy path where y_3 jumps from being positive to zero and y_5, y_6 jump from zero to positive.

6.2. Degenerate nonlinear problem.

We now consider a problem with nonlinear constraints. In particular, we consider the problem

$$\begin{aligned} &\min_x \quad -x_2 \\ &\text{subject to:} \\ &c_1(x) := x_3 = 1 + 9t, \\ &c_2(x) := x_1 \geq 0, \\ &c_3(x) := -x_2^3 - x_1 x_2 - x_1^2 + x_3^2 \geq 0, \\ &c_4(x) := -e^{x_1} - e^{x_2} + e^{x_3} + 1 \geq 0, \\ &c_5(x) := -x_1^2 - x_1 x_2 + (x_2 - (2.5 + 0.5x_3))^2 - (2.5 + 0.5x_3)^4 x_1 \\ &\qquad\quad - 100(x_2 - (2.5 + 0.5x_3)) \geq 0, \\ &c_6(x) := -x_1^2 + x_1 x_2 + (x_2 - (2.5 + 0.5x_3))^2 + (2.5 + 0.5x_3)^4 x_1 \\ &\qquad\quad - 100(x_2 - (2.5 + 0.5x_3)) \geq 0. \end{aligned} \tag{6.1}$$

For this problem $x^*(t) = (0, 1+9t, 1+9t)$ for $t \in [0, \frac{4}{9}]$, and $x^*(t) = (0, 3+4.5t, 1+9t)$ for $t \in [\frac{4}{9}, 1]$. For $t \in [0, \frac{4}{9})$ constraints 1, 2, 3, and 4 are active (including one equality constraint), and for $t \in (\frac{4}{9}, 1]$, (only) constraints 1, 2, 5, and 6 are active. At $t = \frac{4}{9}$ all constraints are active. Since there are always at least four active constraints and n is three, the Jacobian is trivially rank deficient. However, for all t, $(0, -1, 0)$ is a strictly feasible direction at $x^*(t)$, and so the MFCQ holds. Furthermore, it can be seen that the CRCQ holds at $x^*(p)$ for all p. The results are given in Figure 4. Note the discontinuous jump of some of the multipliers for $t = 4/9$.

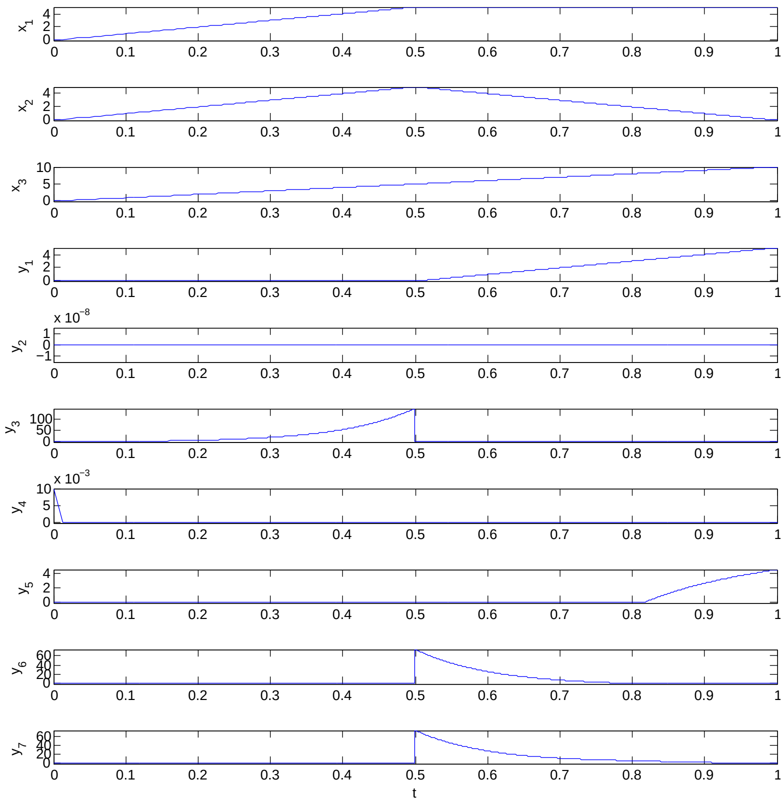

FIG. 3. *Path of primal and dual variables tracing the approximate solution to* (2.1) *across* $t = [0, 1]$.

6.3. Convergence and comparison. In order to illustrate the convergence properties of the algorithm and their implications for practical application, we plot the values of $\eta(x_k, y_k, t_k)$ as we trace the homotopy for problem (2.1) in Figure 5 and for problem (6.1) in Figure 6. The tolerance in (5.1) was chosen as η_{tol} =1e-5, and Δt is allowed to increase to speed up the tracking, as long as $\eta < \eta_{\text{tol}}$. We observe superlinear contraction followed by repeatedly allowing η to inch up toward the optimality tolerance before decreasing Δt again for achieving the required accuracy. In the figures, we also show how η evolves when we implement the path-following algorithm using only the predictor (and not the corrector), as proposed in [24]. We clearly see that it diverges, suggesting a necessity for a corrector. We also considered dropping (JumpLP) from the algorithm and just using the multipliers from (CorrectStep) and (QPPredict), and as expected, the homotopy is traced up until the jump seen at $t = 0.5$, at which point the algorithm fails to trace the optimal solution path. In this case, the values of all the variables and η are the same as in our algorithm up to $t = 0.5$. We note that we have essentially no algorithms to compare ours to in terms of performance in iterations or speed, because Algorithm 1 solves a new class of problems for which previously no algorithm exists, aside from [24], which has no convergence theory and as we see does not stay close to the solution curve, or [26], which requires solving an exponentially cascading sequence of LPs, unnecessary for the problem setting we consider, and thus becomes prohibitively slow for an increasing number of constraints.

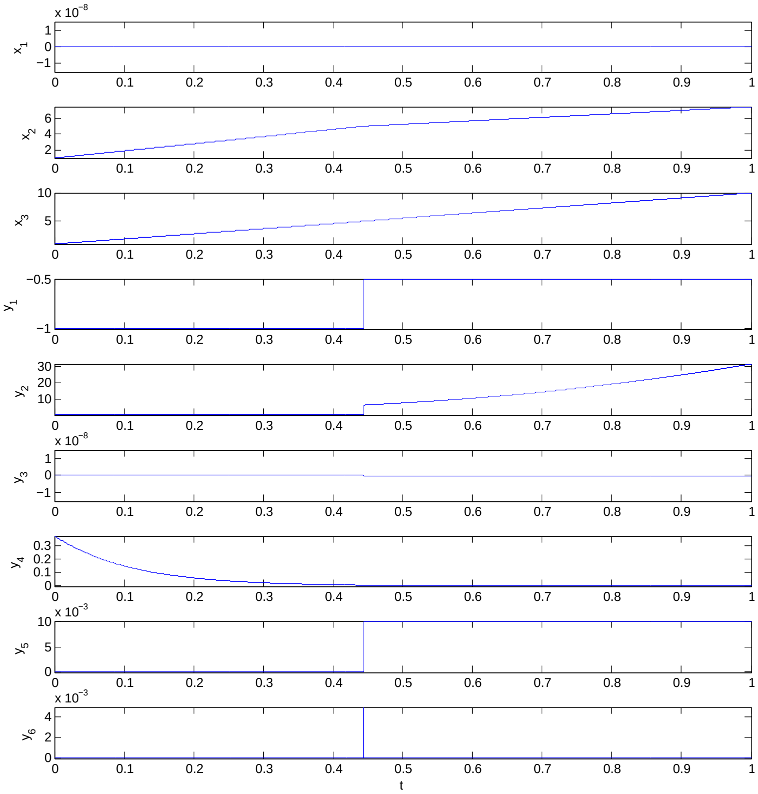

FIG. 4. *Primal-dual approximate solution trajectories for Algorithm* 1 *applied to* (6.1).

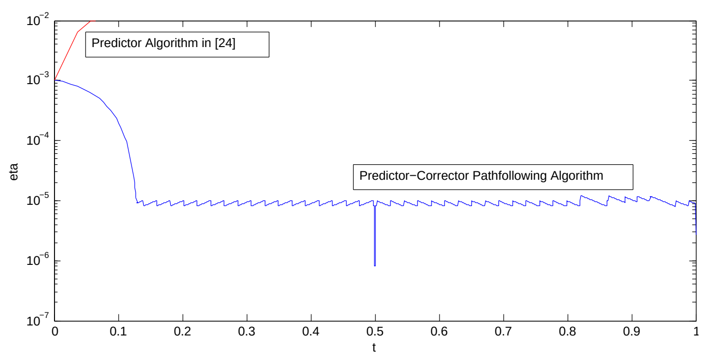

FIG. 5. *Plot of* η *across the homotopy for Algorithm* 1 *and pure predictor and LP Jump algorithm* [24] *to* (2.1) *across* $t = [0, 1]$.

7. Conclusion. In this paper we investigated the properties of a predictor-corrector path-following algorithm for parametric optimization. The algorithm consists of solving a linear system that corresponds to a corrector step, a QP that corresponds to a corrected predictor, and an LP used to jump over discontinuities in the optimal Lagrange multiplier. The procedure exhibits several desirable properties for an appropriate algorithm for the problems of interest, and in particular we have proved its convergence properties without assuming the LICQ holds at any of the primal solutions along the path.

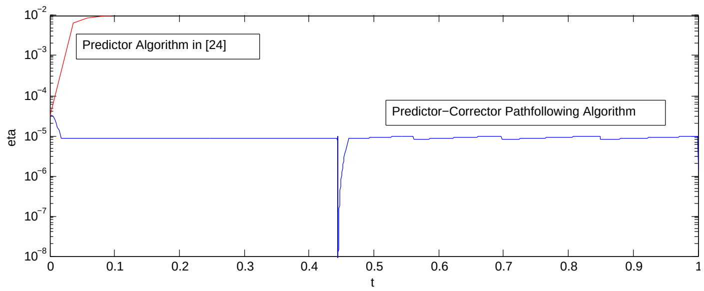

FIG. 6. *Plot of η across the homotopy for Algorithm* 1 *and pure predictor and LP Jump algorithm* [24] *to* (6.1) *across* $t = [0, 1]$.

Future research on the topic can include several directions: first, solving problems with even weaker assumptions on the problem data, inspired by singular systems in optimal control with bifurcations in the solution to the dynamics; second, applying the algorithm to problems arising in particular settings, especially nonlinear model predictive control; third, it may be possible to modify the predictor-corrector subproblems to be reformed as a phase 1-phase 2 type solver for standalone NLPs; fourth, studying uniformity properties of the sensitivity theory used in the paper, as mentioned in the discussion of the convergence theory results in section 4.2; finally, it would be interesting to investigate the application of this algorithm in parametric problems with uncertainty, i.e., subject to additional parameters, to be optimized in an either robust (worst-case) or stochastic/statistical sense.

Acknowledgments. J.J. would like to thank Larry Biegler for many interesting discussions on the topic of path-following and sensitivity. V.K. would like to thank Daniel P. Robinson for his helpful suggestions for improving the readability of the paper. In addition the authors would like to thank two anonymous referees as well as the Associate Editor, Mikhail Solodov, for their recommendations in revising the manuscript.

REFERENCES

[1] E. L. ALLGOWER AND K. GEORG, *Numerical path following*, Handb. Numer. Anal., 5 (1997), pp. 3–207.

[2] R. ANDREANI, E. G. BIRGIN, J. M. MARTÍNEZ, AND M. L. SCHUVERDT, *On augmented Lagrangian methods with general lower-level constraints*, SIAM J. Optim., 18 (2007), pp. 1286–1309.

[3] J. F. BONNANS AND A. SHAPIRO, *Optimization problems with perturbations: A guided tour*, SIAM Rev., 40 (1998), pp. 228–264.

[4] R. S. DEMBO, S. C. EISENSTAT, AND T. STEIHAUG, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[5] S. DEMPE, *Foundations of Bilevel Programming*, Springer, New York, 2002.

[6] P. DEUFLHARD, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, Vol. 35, Springer, New York, 2011.

[7] M. DIEHL, H. G. BOCK, J. P. SCHLÖDER, R. FINDEISEN, Z. NAGY, AND F. ALLGÖWER, *Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations*, J. Process Control, 12 (2002), pp. 577–585.

[8] M. DIEHL, A. WALTHER, H. G. BOCK, AND E. KOSTINA, *An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization*, Optim. Methods Softw., 25 (2010), pp. 531–552.

[9] Q. T. DINH, C. SAVORGNAN, AND M. DIEHL, *Adjoint-based predictor-corrector sequential convex programming for parametric nonlinear optimization*, SIAM J. Optim., 22 (2012), pp. 1258–1284.

[10] A. L. Dontchev, M. I. Krastanov, R. T. Rockafellar, and V. M. Veliov, *An Euler–Newton continuation method for tracking solution trajectories of parametric variational inequalities*, SIAM J. Control Optim., 51 (2013), pp. 1823–1840.

[11] A. L. Dontchev and R. T. Rockafellar, *Convergence of inexact Newton methods for generalized equations*, Math. Program., 139 (2013), pp. 115–137.

[12] J. Dupačová, *Stability and sensitivity-analysis for stochastic programming*, Ann. Oper. Res., 27 (1990), pp. 115–142.

[13] F. Facchinei, A. Fischer, and C. Kanzow, *On the accurate identification of active constraints*, SIAM J. Optim., 9 (1998), pp. 14–32.

[14] D. Fernández and M. Solodov, *Stabilized sequential quadratic programming for optimization and a stabilized Newton-type method for variational problems*, Math. Program., 125 (2010), pp. 47–73.

[15] J. Gauvin, *A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming*, Math. Program., 12 (1977), pp. 136–138.

[16] P. E. Gill, V. Kungurtsev, and D. P. Robinson, *A stabilized SQP method: superlinear convergence*, Math. Program., (2016), pp. 1–42.

[17] P. E. Gill, V. Kungurtsev, and D. P. Robinson, *A stabilized SQP method: Global convergence*, IMA J. Numer. Anal., 37 (2017), pp. 407–443.

[18] P. E. Gill and E. Wong, *Sequential quadratic programming methods*, in Mixed Integer Nonlinear Programming, Springer, New York, 2012, pp. 147–224.

[19] J. Guddat, F. Guerra Vazquez, and H. Th. Jongen, *Parametric Optimization: Singularities, Pathfollowing and Jumps*, Springer, New York, 1990.

[20] J. Guddat, F. Guerra Vázquez, D. Nowack, and J.-J. Rückmann, *A modified standard embedding with jumps in nonlinear optimization*, European J. Oper. Res., 169 (2006), pp. 1185–1206.

[21] W. W. Hager, *Stabilized sequential quadratic programming*, Comput. Optim. Appl., 12 (1999), pp. 253–273.

[22] A. F. Izmailov, *Solution sensitivity for Karush–Kuhn–Tucker systems with non-unique Lagrange multipliers*, Optimization, 59 (2010), pp. 747–775.

[23] A. F. Izmailov and M. V. Solodov, *Inexact Josephy–Newton framework for generalized equations and its applications to local analysis of Newtonian methods for constrained optimization*, Comput. Optim. Appl., 46 (2010), pp. 347–368.

[24] J. Jäschke, X. Yang, and L. T. Biegler, *Fast economic model predictive control based on NLP-sensitivities*, J. Process Control, 24 (2014), pp. 1260–1272.

[25] V. Kungurtsev, *Second-Derivative Sequential Quadratic Programming Methods for Nonlinear Optimization*, Ph.D. thesis, University of California, San Diego, 2013.

[26] V. Kungurtsev and M. Diehl, *Sequential quadratic programming methods for parametric nonlinear optimization*, Comput. Optim. Appl., 59 (2014), pp. 475–509.

[27] J. Kyparisis, *Sensitivity analysis for nonlinear programs and variational inequalities with nonunique multipliers*, Math. Oper. Res., 15 (1990), pp. 286–298.

[28] A. B. Levy, *Solution sensitivity from general principles*, SIAM J. Control Optim., 40 (2001), pp. 1–38.

[29] T. Ohtsuka, *A continuation/GMRES method for fast computation of nonlinear receding horizon control*, Automatica, 40 (2004), pp. 563–574.

[30] D. Ralph and S. Dempe, *Directional derivatives of the solution of a parametric nonlinear program*, Math. Program., 70 (1995), pp. 159–172.

[31] M. Ringkamp, S. Ober-Blöbaum, M. Dellnitz, and O. Schütze, *Handling high-dimensional problems with multi-objective continuation methods via successive approximation of the tangent space*, Eng. Optim., 44 (2012), pp. 1117–1146.

[32] S. M. Robinson, *Generalized equations and their solutions, part* II*: Applications to nonlinear programming*, in Optimality and Stability in Mathematical Programming, 1982, pp. 200–221.

[33] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Vol. 317, Springer, New York, 2009.

[34] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 2015.

[35] L. N. Vicente and S. J. Wright, *Local convergence of a primal-dual method for degenerate nonlinear programming*, Comput. Optim. Appl., 22 (2002), pp. 311–328.

[36] Y. Wang and S. Boyd, *Fast model predictive control using online optimization*, IEEE Trans. Control Syst. Tech., 18 (2010), pp. 267–278.

[37] S. J. Wright, *Modifying SQP for degenerate problems*, SIAM J. Optim., 13 (2002), pp. 470–497.

[38] P. O. Yapo, H. V. Gupta, and S. Sorooshian, *Multi-objective global optimization for hydrologic models*, J. Hydrology, 204 (1998), pp. 83–97.
[39] V. M. Zavala and M. Anitescu, *Real-time nonlinear optimization as a generalized equation*, SIAM J. Control Optim., 48 (2010), pp. 5444–5467.
[40] V. M. Zavala and L. T. Biegler, *The advanced-step NMPC controller: Optimality, stability and robustness*, Automatica, 45 (2009), pp. 86–93.