INTERIOR POINT METHODS FOR OPTIMAL CONTROL OF DISCRETE-TIME SYSTEMS

STEPHEN J. WRIGHT *

Abstract. We show that recently developed interior point methods for quadratic programming and linear complementarity problems can be put to use in solving discrete-time optimal control problems, with general pointwise constraints on states and controls. We describe interior point algorithms for a discrete time linear-quadratic regulator problem with mixed state/control constraints, and show how it can be efficiently incorporated into an inexact sequential quadratic programming algorithm for nonlinear problems. The key to the efficiency of the interior-point method is the narrow-banded structure of the coefficient matrix which is factorized at each iteration.

Key words. interior point algorithms, optimal control, banded linear systems.

1. Introduction. The problem of optimal control of an initial value ordinary differential equation, with Bolza objectives and mixed constraints, is

$$\begin{aligned} &\min_{x,u} \int_0^T L(x(t),u(t),t)\,dt + \phi_f(x(T)), \\ &\dot{x}(t) = f(x(t),u(t),t), \quad x(0) = x_{\text{init}}, \\ &g(x(t),u(t),t) \le 0, \quad t \in [0,T], \qquad g_f(x(T)) \le 0. \end{aligned} \tag{1.1}$$

Here, $x : [0,T] \to I\!R^{n_s}$, $u : [0,T] \to I\!R^{n_c}$, $L : I\!R^{n_s} \times I\!R^{n_c} \times [0,T] \to I\!R$, $\phi_f : I\!R^{n_s} \to I\!R$, $g : I\!R^{n_s} \times I\!R^{n_s} \times [0,T] \to I\!R^{n_g}$, $g_f : I\!R^{n_s} \to I\!R^{n_f}$. A discrete-time counterpart is the problem

$$\begin{aligned} &\min_{x_i,u_i} \sum_{i=1}^{N} L_i(x_i,u_i) + \phi_N(x_{N+1}), \\ &x_{i+1} = f_i(x_i,u_i), \quad i = 1,\cdots,N, \qquad x_1 \text{ fixed}, \\ &g_i(x_i,u_i) \le 0, \quad i = 1,\cdots,N, \qquad g_f(x_{N+1}) \le 0. \end{aligned} \tag{1.2}$$

Efficient algorithms have been proposed for various special classes of these problems. In the "unconstrained" case (that is, when g, g_i and g_f are absent), Newton-like methods and conjugate gradient methods for (1.1) are described by Polak [27]; for (1.2), Newton's method, and its efficient implementation, is discussed in Dunn and Bertsekas [8]. A variety of quasi-Newton approaches have also been applied to the unconstrained version of (1.1); see, for example, Edge and Powers [9] and Kelley and Sachs [16], and the references therein. In the control-constrained case (in which g_f is absent, and the states x and x_i do not appear in g and g_i), the problem is traditionally treated as a constrained optimization problem in u or u_i. Because of the "pointwise" or separable nature of the constraints, methods of the gradient

* Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439. Research supported by the Applied Mathematical Sciences subprogram of the Office of Energy Research, U. S. Department of Energy, under Contract W-31-109-Eng-38.

projection class are easily implementable (see, for example, Demyanov and Rubinov [3], Dunn [5, 6]). In the finite-dimensional problem, these methods have the advantage that the set of currently-active constraints can change extensively at each iteration, whereas "active set" methods only allow a single change to the active set, which causes poor performance when there are many constraints. More recently, Newtonian scaling has been added to gradient projection algorithms (see Gafni and Bertsekas [12], Dunn [7]) to improve their asymptotic rate of convergence, and the resulting methods have proven to be useful for the control-constrained version of (1.2), as we see in Section 6. Pantoja and Mayne [26] have described a stagewise algorithm for the control-constrained case that, in a neighborhood of a solution of (1.2), produces iterates that are identical to sequential quadratic programming iterates. Instead of making use of the inherent structure in (1.2) at the level of the linear algebra computations, as we do in this paper and also in [32, 31], Pantoja and Mayne exploit the structure at a somewhat higher level.

The most general cases of (1.1),(1.2), in which the functions g, g_i, g_f are nontrivial in both states and controls, are known to be significantly more difficult to solve than the special cases described above. Algorithms based on nonlinear programming techniques appear to be the most promising. In these algorithms, both states and controls are treated as unknowns and the state equation and "auxiliary" constraints as equality and inequality constraints, respectively. Miele [21] deals mainly with the case in which the auxiliary constraints g in (1.1) are equalities (rather than inequalities) and proposes algorithms of the reduced gradient type, with features added to ensure near-feasibility of all iterates. Polak, Yang and Mayne [28] describe a first-order algorithm which makes use of barrier functions for the inequality constraints. Evtushenko [10, Chapter 6] describes a variety of augmented Lagrangian penalty function methods, in which (1.2) is reduced to an unconstrained problem. Di Pillo, Grippo and Lampariello [4] describe a structured quasi-Newton method for a particular augmented Lagrangian, and take advantage of the same feature which we exploit in this paper: bandedness of the coefficient matrix which is factored at each iteration.

In this paper, we focus on a linear-quadratic version of the discrete problem (1.2). This can be formulated as

$$
\begin{aligned}
&\min_{x_i,u_i} \sum_{i=1}^{N} r_i^T u_i + z_i^T x_i + \frac{1}{2}(x_i^T Q_i x_i + 2x_i^T R_i u_i + u_i^T S_i u_i) \\
&\qquad\qquad + z_{N+1}^T x_{N+1} + \frac{1}{2} x_{N+1}^T Q_{N+1} x_{N+1}, \\
&x_{i+1} = A_i x_i + B_i u_i + s_i, \quad i = 1, \cdots, N, \qquad x_1 \text{ fixed}, \\
&G_i u_i + H_i x_i \le g_i, \quad i = 1, \cdots, N, \qquad H_{N+1} x_{N+1} \le g_{N+1}.
\end{aligned} \tag{1.3}
$$

Sequential quadratic programming algorithms for the nonlinear problem (1.2) give rise to subproblems of this form at each iteration. The algorithm that we propose may be less efficient than other algorithms on special cases of (1.3) (see, for example, comparisons with a gradient projection algorithm for control-bounded problems in Section 6), but we claim that it forms the basis of an efficient general solution procedure for (1.3).

For the continuous problem (1.1), solution of a problem of the form (1.3), or something similar, remains a "core" operation. In the numerical results of Section 6, we restrict attention to simple discretizations of the continuous problems. Higher-order discretizations are possible: for example, Cuthrell and Biegler [2] use collocation at Gauss points to convert (1.1) to a nonlinear programming problem. Many issues arise in the discretization process, particularly when the solution of (1.1) contains singular arcs (and so the first-order necessary conditions give rise to a higher-index differential-algebraic equation). We will not discuss them further here, except to point out that the general methodology of this paper is applicable whenever discretizations with "local support" are used, since these lead to block-banded linear systems of the type described in Section 5.

Our main task here is to show that interior point methods may be useful tools for solving problems of the form (1.3), and that these methods can be embedded in inexact sequential quadratic programming algorithms to solve problems of the form (1.2). If (1.3) arises as a discretization of a continuous problem, alternative algorithms from mathematical programming would seem to be less efficient as N grows very large. For example, the number of iterations required by active set methods (see Fletcher [11, Chapter 10]) could reasonably be expected to be proportional to the number of constraints, that is, $O(N)$. Since each such iteration involves the solution of a certain narrow-banded linear system of dimension $O(N)$, the total complexity would probably be (N^2). Another possibility is to use algorithms of the gradient projection class, but these are difficult to implement when both states and controls are variable, because of the complexity of the feasible set in (1.3). We observe in Section 6 that the number of interior point iterations required to solve (1.3) is often independent of N and is always better than $O(N^{1/2})$. Since the main task is each iteration also involves solution of a linear system with a banded coefficient matrix of dimension $O(N)$, the total amount of work is between $O(N)$ and $O(N^{3/2})$ in practice. Other researchers have also noted that in many cases the iteration count is practically almost independent of N though, as we show in the next section, formal analyses suggest that it should be $O(N^{1/2})$.

Interesting algorithms have recently been proposed by Rockafellar and co-workers [30, 36] for extended linear-quadratic programming, a class of problems that includes discrete-time linear-quadratic optimal control problems. They aim to find the saddle point of a Lagrangian which, for multistage problems such as (1.3), has the property that it is decomposable with respect to the primal variables when the dual variables are fixed, and vice versa. In Zhu and Rockafellar [36], primal-dual steepest descent and conjugate gradient algorithms which take advantage of this structure are used, and linear convergence results are proved. A finite termination results is proved for the conjugate gradient algorithm. These algorithms take advantage of the structure at a higher level than the linear algebra, but the $O(N)$ complexity per iteration is similar to ours. An interesting question is whether structured interior point methods of the type discussed in this paper can be efficiently used to solve the entire extended linear-quadratic programming class.

We assume throughout that a convexity condition holds:

$$\text{(1.4)} \qquad \begin{bmatrix} Q_i & R_i \\ R_i^T & S_i \end{bmatrix} \text{ is positive semidefinite for } i = 1, \cdots, N; \quad Q_{N+1} \text{ is positive semidefinite.}$$

The second-order sufficiency conditions for (1.3) to have an isolated local solution are weaker than this; however, (1.4) holds in practice for many problems.

The remainder of the paper is laid out as follows. In Section 2, we introduce two classes of interior point algorithms for convex quadratic programming. These algorithms are described with respect to a general formulation of the problem (see (2.1)) rather than the specific problem (1.3). Convergence theory for these algorithms is described. Section 3 discusses some of the practical issues that arise in the implementation of these two algorithms, with reference again to the general formulation (2.1). In Section 4, we discuss an inexact sequential quadratic programming algorithm for the general nonlinear programming problem. The convergence analysis can be derived from existing theory for mixed nonlinear complementarity problems, and the stopping criterion for each quadratic subproblem is shown to be easily evaluated. In Section 5, we show how the algorithms described in the preceding three sections can take special advantage of the structure inherent in the problems (1.2) and (1.3), by using linear algebra techniques for banded linear systems as described in Wright [33, 32]. Although we do not deal with it in this paper, the whole approach is conducive to parallel implementation: the task of evaluating of the functions and gradients in (1.2) can clearly be divided between independent processors, while parallel solution of the banded linear system can be carried out by using the techniques discussed in [33].

In the remainder of the paper, superscripts on vector or matrix quantities represent iteration numbers, while subscripts are used either to distinguish different components of a vector, or to distinguish different stages of the optimal control problem, as in (1.3). Subscripts on scalars denote iteration numbers. $\|.\|$ denotes the Euclidean norm, unless otherwise specified.

2. Interior-point algorithms for convex quadratic programming. In this section, we give the general outline of recently developed interior point algorithms for convex quadratic programming. These algorithms usually also apply to linear complementarity problems, and in the descriptions which follow we will make use of the connection between these two classes of problems. It has been a source of some frustration in recent years that the interior point algorithms with desirable theoretical properties (polynomial complexity, superlinear convergence) tend to be slow in practice, while little can be proved about the algorithms that perform exceptionally well. Developments continue to occur at a rapid pace, and the performance gap is closing.

We will outline two interior point methods, one which has polynomial complexity and superlinear convergence and one which tends to be faster in practice but which does not have these nice theoretical properties. Both algorithms can be motivated within a common

simple framework, which we describe after stating the problem and discussing the relationship between the primal and dual formulations.

We assume that the convex quadratic program has the following form:

$$\min_z \ \frac{1}{2} z^T Q z + c^T z, \qquad Az = b, \qquad Cz \leq d, \tag{2.1}$$

where $z \in \mathbb{R}^n$, $b \in \mathbb{R}^{m_e}$, $d \in \mathbb{R}^m$, etc., and Q is positive semi-definite. We assume throughout that (2.1) has an optimal solution $(z, \nu) = (z^*, \nu^*)$, where ν is a vector of slacks for the inequality constraints. The dual of (2.1) is

$$\max_{v,w,y} \ -\frac{1}{2} v^T Q v - b^T w - d^T y, \qquad Qv + A^T w + C^T y + c = 0, \qquad y \geq 0. \tag{2.2}$$

The relationship between problems (2.1) and (2.2) is outlined in the following proposition (see, for example, Monteiro and Adler [23, Propositions 2.1–2.3] and Mangasarian [20, Section 8.2]):

PROPOSITION 2.1.

(i) If (2.1) has an optimal solution (z^*, ν^*), then there exist w^* and y^* such that $(v, w, y) = (z^*, w^*, y^*)$ is an optimal solution for (2.2). Conversely, if (2.2) has an optimal solution, then so does (2.1).

(ii) If (z^*, ν^*) and (v^*, w^*, y^*) are optimal solutions of (2.1) and (2.2), respectively, then $(y^*)^T \nu^* = 0$. Conversely, if (v^*, w^*, y^*) is a feasible solution of (2.2) such that $z = v^*$, $\nu = d - Cv^*$ are feasible in (2.1) and $(y^*)^T (d - Cv^*) = 0$, then $(v^*, d - Cv^*)$ and (v^*, w^*, y^*) are optimal solutions of (2.1) and (2.2), respectively.

Since we have assumed that a primal solution exists, it follows from this Proposition that there is a quartet (z, ν, w, y) such that

$$Qz + A^T w + C^T y + c = 0 \tag{2.3}$$

$$-Az + b = 0 \tag{2.4}$$

$$\nu = -Cz + d \geq 0 \tag{2.5}$$

$$y \geq 0, \tag{2.6}$$

$$y^T \nu = 0. \tag{2.7}$$

If (z, ν, w, y) is feasible with respect to the first four of these conditions (2.3)–(2.6), then

$$\begin{aligned} y^T \nu &= y^T (d - Cz) \\ &= y^T d - z^T (-c - Qz - A^T w) \\ &= y^T d + w^T b + z^T Q z + z^T c, \end{aligned}$$

which is the difference between the primal and dual objective function values, or *duality gap*. Interior point methods maintain feasibility with respect to (2.3)–(2.6) while gradually

reducing the duality gap to zero. In fact, all (y, ν) iterates of these algorithms remain in the *strictly* feasible set defined by

$$\mathcal{F}_+ = \{(y, \nu) \mid \ (z, \nu, w, y) \text{ satisfies (2.3)–(2.6) for some } (z, w), \qquad y > 0, \nu > 0\}.$$

(This fact justifies the use of the term "interior".) The progress of the algorithm towards a solution can be gauged by examining the duality gap $\nu^T y$ itself, or some potential function constructed from it, for example

$$\psi(y, \nu) = \rho_p \ln(y^T \nu) - \sum_{j=1}^{m} \ln(y_j \nu_j), \tag{2.8}$$

where ρ_p is a fixed barrier parameter. Besides being elements of $\mathcal{F}_+$, it is desirable for each iterate to remain in the vicinity of a "central path" defined by

$$\mathcal{C} = \{(\nu, y) \in \mathcal{F}_+ \mid \nu_i y_i = (\nu^T y)/m, \qquad i = 1, \cdots, m\}. \tag{2.9}$$

(Measures of closeness to $\mathcal{C}$ will be discussed later.) Maintaining closeness to the central path helps in retaining the strict interiority property and facilitates the convergence analysis.

In general, each step of the method aims to retain feasibility with respect to (2.3)–(2.6), while moving closer to the central path and/or reducing the duality gap. The latter two aims may be satisfied by performing a Newton-like linearization of the equations

$$\nu_i y_i = \mu, \qquad i = 1, \cdots, m,$$

or, more succinctly,

$$MYe = \mu e, \tag{2.10}$$

where $M = \text{diag}(\nu_1, \nu_2, \cdots, \nu_m)$, $Y = (y_1, y_2, \cdots, y_m)$, $e = (1, 1, \cdots, 1)^T$, and $\mu \geq 0$. If we set $\mu = 0$, then we are clearly aiming to satisfy the complementarity condition (2.7) without regard to the central path, while if we set $\mu = (y^{kT} \nu^k)/m$ where (ν^k, y^k) is the current iterate, then we are aiming to move closer to the "nearest point" on the central path in some sense while not reducing the duality gap. As we see below, μ is often chosen to lie strictly between these two extremes, so that it partially satisfies both aims.

We note here that (2.3)–(2.7) is a linear complementarity problem, in which the coefficient matrix

$$\begin{bmatrix} Q & A^T & C^T \\ -A & 0 & 0 \\ -C & 0 & 0 \end{bmatrix}$$

is positive semi-definite. The "linear complementarity" point of view is useful when discussing convergence of the interior point algorithms, and when describing the choice of an initial point, as we do in the next section.

The interior point framework can now be specified. Suppose an initial point (z^0, ν^0, w^0, y^0) that satisfies (2.3)–(2.6) with strict inequalities in (2.5)–(2.6) is available. At the current iterate (z^k, ν^k, w^k, y^k), we generate the search direction $(\delta z^k, \delta \nu^k, \delta w^k, \delta y^k)$ by solving the following system of equations:

$$\begin{aligned} Q\delta z^k + A^T \delta w^k + C^T \delta y^k &= 0 \\ A\delta z^k &= 0 \\ C\delta z^k + \delta \nu^k &= 0 \\ Y^k \delta \nu^k + M^k \delta y_k &= \mu_k e - Y^k M^k e. \end{aligned} \tag{2.11}$$

Elimination of $\delta \nu^k$ yields the (symmetric, indefinite) system

$$\begin{bmatrix} Q & A^T & C^T \\ A & 0 & 0 \\ C & 0 & -(Y^k)^{-1} M^k \end{bmatrix} \begin{bmatrix} \delta z^k \\ \delta w^k \\ \delta y^k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -\mu_k (Y^k)^{-1} e + M^k e \end{bmatrix}. \tag{2.12}$$

Now, we set $z^{k+1} = z^k + \theta_k \delta z^k$, $\nu^{k+1} = \nu^k + \theta_k \delta \nu^k$, $w^{k+1} = w^k + \theta_k \delta w^k$, and $y^{k+1} = y^k + \theta_k \delta y^k$, where θ_k is chosen to retain feasibility of the new iterate with respect to (2.3)–(2.6), among other things.

Potential reduction algorithms choose μ_k to depend on the duality gap: specifically

$$\mu_k = \frac{\Delta_k}{\rho_k}, \qquad \Delta_k \stackrel{\text{def}}{=} y^{kT} \nu^k, \tag{2.13}$$

where the value of ρ_k will be discussed below. θ_k is then chosen so that $\psi(y^k + \theta_k \delta y^k, \nu^k + \theta_k \delta \nu^k)$ achieves at least certain constant reduction over $\psi(y^k, \nu^k)$, where ψ is defined in (2.8). The following result can be used to demonstrate polynomial complexity of the basic potential reduction algorithm. It is proved by Kojima, Mizuno and Yoshise [17] for the linear complementarity problem in standard form, but can be easily extended to the "mixed" linear complementarity problem (2.3)–(2.7).

THEOREM 2.2. *Suppose that*

(i) The set $\mathcal{C}$ is nonempty

(ii) $\rho_k \equiv \rho_p = m + \sqrt{m}$ in (2.8) and (2.13)

(iii) $m \geq 2$

(iv)

$$\theta_k = 0.4 \frac{\min_{j=1,\cdots,m} \sqrt{y_j^k \nu_j^k}}{\|(Y^k M^k)^{-1/2} (\frac{\Delta_k}{\rho_k} e - Y^k M^k e)\|}.$$

Then the iterates produced by the interior point algorithm satisfy

$$\psi(y^{k+1}, \nu^{k+1}) \leq \psi(y^k, \nu^k) - 0.2, \qquad k = 0, 1, \cdots.$$

COROLLARY 2.3. *Suppose that the assumptions of Theorem 2.2 hold, and let L be any number with $L > -\psi(y^0, \nu^0)/\sqrt{m}$. Then the iterates generated by the algorithm will satisfy $\Delta_k = y^{kT}\nu^k \leq e^{-L}$ for all $k \geq 5\,[\psi(y^0, \nu^0) + \sqrt{m}L]$.*

Proof. The result follows immediately from Theorem 2.2 together with the fact that

$$\psi(y, \nu) \leq -\sqrt{m}L \Rightarrow y^T\nu \leq e^{-L}.$$

■

Polynomial complexity follows if $\psi(y^0, \nu^0) = O(\sqrt{m}L)$, since the time for each iteration is polynomial in the problem size. (Strictly speaking, the standard assumption that all data in the problem (2.1) is rational is also needed.)

The algorithm of Theorem 2.2 can certainly not be expected to yield superlinear convergence, and indeed it is very slow in practice. Other researchers have analyzed algorithms in which the choices of ρ_p, ρ_k and θ_k are relaxed. However, the choices of ρ_k and θ_k that yield the most efficient practical algorithms lie outside the scope of this analysis. We use the following heuristics (which are similar to those utilized by Han, Pardalos and Ye [13]):

initially: $\rho_{\min} \leftarrow m^{1.5}$

the k-th iteration:

$\rho_k \leftarrow \max(\rho_{\min}, 1/\Delta_k)$;
calculate the step $(\delta z^k, \delta \nu^k, \delta w^k, \delta y^k)$;
set $\bar{\theta}_k = \max\{\theta \mid \theta \leq 1,\ y_i^k + \theta \delta y_i^k \geq 0,\ \ \nu_i^k + \theta \delta \nu_i^k \geq 0,\ \ i = 1, \cdots, m\}$,

if $\bar{\theta}_k \leq .5$ **then** $\rho_{\min} \leftarrow .5 * \rho_{\min}$;
if $\bar{\theta}_k = 1$ **then** $\rho_{\min} \leftarrow 2 * \rho_{\min}$;
if $\bar{\theta}_k = 1$ **then** $\theta_k \leftarrow 1$ **else** $\theta_k \leftarrow \beta\bar{\theta}_k$
Take the step of length θ_k and go to next iteration;

Here, the constant β is set to .9995.

In our experience, the choice of ρ_k should be manipulated to be as large as possible, while allowing steplengths θ_k of 1, since these steps were usually observed to produce the largest reductions in the duality gap. The heuristic above was found, after some experimentation, to be quite successful. As noted by Zhang, Tapia and Dennis [35], the choice $\rho_k = 1/\Delta_k$ (which takes effect during the last few iterations) ensures quadratic convergence of the duality gap to zero.

The second algorithm we consider is of the "predictor-corrector" type. For linear programming problems, this algorithm is described by Mizuno, Todd and Ye [22] and Ye, Tapia and Zhang [34]. The analysis is extended to linear complementarity problems by Ji, Potra and Huang [15]. The algorithm makes use of the idea of an "α-neighborhood" of the central

path $\mathcal{C}$, defined as follows:

$$\mathcal{C}(\alpha) = \left\{ (y,\nu) \in \mathcal{F}_+ \;\middle|\; \left\| \frac{MYe}{(y^T\nu/m)} - e \right\| \le \alpha \right\},$$

where $\alpha \in [0,1)$. (Note that $\mathcal{C} = \mathcal{C}(0)$.) It is assumed that the initial point (z^0, ν^0, w^0, y^0) has $(y^0, \nu^0) \in \mathcal{C}(\frac{1}{4})$. To complete the definition of the algorithm, we need only specify the choices for μ_k and θ_k:

(i) For k even, choose $\mu_k = 0$ and θ_k to be the largest value in $(0,1]$ such that

$$(y^k + \theta_k \delta y^k, \nu^k + \theta_k \delta \nu^k) \in \mathcal{C}(\frac{1}{2});$$

(ii) for k odd, choose $\mu_k = (y^{kT}\nu^k/m)$ and $\theta_k = 1$.

Ji, Potra and Huang show that $(y^k, \nu^k) \in \mathcal{C}(\frac{1}{4})$ for all even k and $(y^k, \nu^k) \in \mathcal{C}(\frac{1}{2})$ for all odd k. Their main results are summarized in the following theorem:

THEOREM 2.4. *([15, Theorem 3.1 and Corollary 4.2]) For the predictor-corrector algorithm described above,*

$$\frac{\Delta_{k+2}}{\Delta_k} \le 1 - \frac{1}{4\sqrt{m}}, \qquad k = 0, 2, 4, \cdots.$$

Suppose that $(y^k, \nu^k) \to (y^*, \nu^*)$ as $k \to \infty$, where (z^*, ν^*, w^*, y^*) is a strictly complementary solution of (2.1), that is, one for which exactly one *of y_i^* and ν_i^* is zero for $i = 1, 2, \cdots, m$. Then the convergence is two-step superlinear, that is*

$$\lim_{k \to \infty,\, k \text{ even}} \frac{\Delta_{k+2}}{\Delta_k} = 0.$$

The two main implementation issues for this algorithm are

- The choice of θ_k for odd k. We solve the scalar equation

$$g(\theta) = \left\| (M^k + \theta\delta M^k)(Y^k + \theta\delta Y^k)e - \frac{(y^k + \theta\delta y^k)^T(\nu^k + \theta\delta\nu^k)}{m} e \right\|^2 - \frac{1}{4}\frac{(y^k + \theta\delta y^k)^T(\nu^k + \theta\delta\nu^k)^2}{m^2} = 0$$

  by using a safeguarded search in the interval $[0,1]$ with an algorithm that has local third-order convergence. Typically about three iterations are required. The cost of this step is much less than the cost of solving the system (2.12).

- The choice of an initial point which satisfies the centrality condition

$$(y^0, \nu^0) \in \mathcal{C}(\frac{1}{4}). \tag{2.14}$$

  When no prior information about the starting point is known, the "cold start" device to be discussed in the next section can be used to choose such a point. In some

circumstances, however, a good initial estimate of the solution which does *not* satisfy (2.14) is available. In this case, we perform only corrector iterations (i.e., setting $\mu = (y^T\nu)/m$) with a line search, until a point which satisfies (2.14) is encountered. During this start-up phase, the line search parameter for each step is chosen to approximately minimize

$$g(\theta) = \left\| \frac{(M^k + \theta\delta M^k)(Y^k + \theta\delta Y^k)}{(y^k + \theta\delta y^k)^T(\nu^k + \theta\delta\nu^k)/m} e - e \right\|^2,$$

while retaining $(y + \theta\delta y, \nu + \theta\delta\nu) \in \mathcal{F}_+$.

3. Implementation details for interior point algorithms. We now discuss two implementation issues that arise in both of the interior point method discussed in the previous section, namely, the choice of an initial point which satisfies (2.3)–(2.6) and the efficient and stable solution of the linear system (2.12).

In some situations, choice of a feasible initial point is made easier by the form of the inequality constraints. Perhaps the most common example for problems of the form (1.3) is one in which the only inequality constraints are bounds on the controls. Controls u_i which lie strictly between their bounds can be chosen, and initial values of the states x_i and costates p_i can be obtained by substituting into the state and adjoint equations, respectively. The slacks ν_i are completely determined by the choice of u_i. The remaining unknowns to be chosen are the multipliers y_i^u and y_i^l that correspond to the upper and lower bounds on u_i, respectively. By taking the derivative of the Lagrangian for (1.3) with respect to u_i, we obtain the following equation that must be satisfied by y_i^u and y_i^l:

$$R_i^T x_i + S_i u_i + r_i - B_i^T p_{i+1} + y_i^u - y_i^l = 0, \qquad i = 1, 2, \cdots, N. \tag{3.1}$$

Since, for centrality, it is also desirable to have the products $\nu_i^u y_i^u$ and $\nu_i^l y_i^l$ as close as possible to each other for $i = 1, 2, \cdots, m$, we choose some "target" value Δ for these products, and then pick y_i^u and y_i^l that satisfy the simple optimization problem

$$\min_{y_i^u,\, y_i^l} \ \frac{\|\nu_i^u y_i^u - \Delta\|^2}{\nu_i^u} + \frac{\|\nu_i^l y_i^l - \Delta\|^2}{\nu_i^l},$$
$$y_i^u \geq \frac{\Delta}{100\nu_i^u}, \qquad y_i^l \geq \frac{\Delta}{100\nu_i^l}, \qquad y_i^u, y_i^l \text{ satisfy (3.1)}.$$

Although this heuristic often produces near-central points, it is sometimes necessary to apply a few centering steps to obtain a point which satisfies (2.14).

When the choice of a feasible point is not so easy (as is often the case when both control and state constraints are present) the problem can be artificially augmented by one extra ν and y component, in such a way that a feasible choice can be made trivially. Monteiro and Adler [23] and Kojima, Mizuno and Yoshise [18] show how to construct feasible initial points that are on or near the central path by introducing artificial variables into convex quadratic

programs and linear complementarity problems, respectively. Here, we describe a scheme for choosing a near-central point which is tailored to the form of (2.3)–(2.6). A modification of this scheme is useful when a good initial estimate of the solution is available; we describe this also.

We introduce a (large, positive) number $\tilde{L} \in I\!R$, whose actual magnitude will be discussed below. Using $\tilde{L}$, define the quantities

$$\begin{aligned} e &= (1, 1 \cdots, 1)^T, \\ a_1 &= -\frac{1}{\tilde{L}} \left[c + \frac{\tilde{L}}{m^2} C^T e \right], \\ a_2 &= -\frac{b}{\tilde{L}}, \\ \hat{c} &= \tilde{L} \left[\frac{1}{m^2} + \frac{1}{m} \right], \end{aligned}$$

and consider the following augmented version of (2.3)–(2.7):

$$Qz + A^T w + C^T y + a_1 \hat{y} + c = 0, \tag{3.2}$$

$$-Az + a_2 \hat{y} + b = 0, \tag{3.3}$$

$$\nu = -Cz + e\hat{y} + d \geq 0, \tag{3.4}$$

$$\hat{\nu} = -a_1^T z - a_2^T w - e^T y + \hat{c} \geq 0, \tag{3.5}$$

$$y \geq 0, \tag{3.6}$$

$$\hat{y} \geq 0, \tag{3.7}$$

$$y^T \nu + \hat{y}\hat{\nu} = 0. \tag{3.8}$$

A little computation shows that the following point is feasible with respect to all but the last (complementarity) condition:

$$z^0 = 0, \quad w^0 = 0, \quad \nu^0 = \tilde{L}e + d, \quad y^0 = \frac{\tilde{L}e}{m^2}, \quad \hat{y}^0 = \tilde{L}, \quad \hat{\nu}^0 = \frac{\tilde{L}}{m^2}.$$

This point can be placed arbitrarily close to the central path by a sufficiently large choice of $\tilde{L}$, since

$$\hat{y}^0 \hat{\nu}^0 = \frac{\tilde{L}^2}{m^2}, \quad y_i^0 \nu_i^0 = \frac{\tilde{L}^2}{m^2} + \frac{\tilde{L}}{m^2} d_i, \quad \text{for } i = 1, \cdots, m.$$

By viewing (3.2)–(3.8) as a linear complementarity problem, we now show that the solutions of this augmented problem correspond to solutions of (2.3)–(2.7), for $\tilde{L}$ sufficiently large. (3.2)–(3.8) can be stated as the mixed linear complementarity problem

$$\begin{bmatrix} M_{11} & M_{12} \\ -M_{12}^T & M_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 0 \\ t_2 \end{bmatrix}, \quad s_2 \geq 0, \quad t_2 \geq 0, \quad s_2^T t_2 = 0, \tag{3.9}$$

where

$$s_1 = \begin{bmatrix} z \\ w \end{bmatrix}, \quad s_2 = \begin{bmatrix} y \\ \hat{y} \end{bmatrix}, \quad r_1 = \begin{bmatrix} c \\ b \end{bmatrix}, \quad r_2 = \begin{bmatrix} d \\ \hat{c} \end{bmatrix}, \quad t_2 = \begin{bmatrix} \nu \\ \hat{\nu} \end{bmatrix},$$
$$M_{11} = \begin{bmatrix} Q & A \\ -A^T & 0 \end{bmatrix}, \quad M_{12} = \begin{bmatrix} C^T & a_1 \\ 0 & a_2 \end{bmatrix}, \quad M_{22} = \begin{bmatrix} 0 & e \\ -e^T & 0 \end{bmatrix}.$$

Clearly, M_{11} and M_{22} are positive semi-definite. The following Lemma shows that any two solution triples for (3.9) must be complementary:

LEMMA 3.1. *Let (s_1^*, s_2^*, t_2^*) and (s_1, s_2, t_2) be solutions of (3.9). Then $s_2^{*T} t_2 = s_2^T t_2^* = 0$.*

Proof.

$$\begin{aligned}(s_2 - s_2^*)^T (t_2 - t_2^*) &= (s_2 - s_2^*)^T \left[-M_{12}^T (s_1 - s_1^*) + M_{22}(s_2 - s_2^*) \right] \\ &= (s_2 - s_2^*)^T M_{22} (s_2 - s_2^*)^T - (s_1 - s_1^*)^T M_{12} (s_2 - s_2^*) \\ &= (s_2 - s_2^*)^T M_{22} (s_2 - s_2^*)^T + (s_1 - s_1^*)^T M_{11} (s_1 - s_1^*).\end{aligned}$$

Now, since $s_2^T t_2 = s_2^{*T} t_2^* = 0$,

$$0 = s_2^{*T} t_2 + s_2^T t_2^* + (s_2 - s_2^*)^T M_{22} (s_2 - s_2^*) + (s_1 - s_1^*)^T M_{11} (s_1 - s_1^*).$$

Each quantity on the right-hand side is non-negative, so the result follows. ∎

It remains to specify the required magnitude of $\tilde{L}$. This is made precise in the following result:

THEOREM 3.2. *Suppose that (2.1) has an optimal solution (z^*, ν^*), with corresponding dual vectors (w^*, y^*). Suppose further that $\tilde{L}$ is chosen large enough that*

$$\hat{c} > a_1^T z^* + a_2^T w^* + e^T y^*. \tag{3.10}$$

Then

$$\begin{aligned} z = z^*, \quad w = w^*, \quad y = y^*, \quad \nu = \nu^* \\ \hat{\nu}^* = \hat{c} - (a_1^T z^* + a_2^T w^* + e^T y^*) > 0, \quad \hat{y}^* = 0, \end{aligned} \tag{3.11}$$

is a solution of the augmented problem (3.2)–(3.6).

Let z, w, y, ν, $\hat{y}$, $\hat{\nu}$ satisfy (3.2)–(3.6). Then $\hat{y} = 0$, and hence (z, ν, w, y) is a solution of (2.1),(2.2).

Proof. The first statement is easily verified by substituting in the conditions (3.2)–(3.8). To prove the second statement, we use Lemma 3.1. Since the vectors $(\nu^*, \hat{\nu}^*)$ constructed in the first part of the theorem must be complementary to the pair $(y, \hat{y})$ which is assumed to exist in the second part (that is, $y^T \nu^* + \hat{y}\hat{\nu}^* = 0$), we have that $\hat{y} = 0$. It follows that (z, ν, w, y) satisfies the conditions (2.3)–(2.7) and, therefore, is a solution of (2.1),(2.2). ∎

In our implementation of the predictor-corrector algorithm, we start by setting $\tilde{L} = 1000\sqrt{n + m_e + m}\|d\|$. If after a number of iterations we still have $\hat{\nu} < \hat{y}$, then $\tilde{L}$ is multiplied

by 10, and the process is repeated. For the primal-dual algorithm, we do not place too much emphasis on starting close to the central path. We start by setting $\tilde{L} = \max(1, \|d\|)$ and, if $\hat{\nu} < \hat{y}$ after a few iterations, then $\hat{c}$ is increased (as suggested by Theorem 3.2) and the problem is re-initialized.

When good initial guesses of the solution of (2.1),(2.2) are available , we can modify the strategy above to take advantage of the available information. This situation can arise when the linear-quadratic problem arises as a subproblem in the sequential quadratic programming algorithm for nonlinear programming. It can also arise in a multilevel method, where the result of solving the problem on a coarse grid is being used as a starting point for a finer grid.

Suppose that the initial guess $(\bar{z}, \bar{\nu}, \bar{w}, \bar{y})$ has $\bar{\nu} > 0$, $\bar{y} > 0$ and $d - C\bar{z} > 0$, and that

$$\begin{aligned} c + Q\bar{z} + A^T\bar{w} + C^T\bar{y} &\approx 0 \\ b - A\bar{z} &\approx 0 \\ d - C\bar{z} - \bar{\nu} &\approx 0. \end{aligned}$$

Choosing some threshold criterion ϵ_L with $0 < \epsilon_L \ll 1$, we define a vector $\bar{e} \in I\!R^m$ such that

$$\bar{e}_i = \begin{cases} 1 & \text{if } 0 < \bar{y}_i \leq \epsilon_L \bar{\nu}_i \\ 0 & \text{otherwise.} \end{cases}$$

Then $\|\bar{e}^T\bar{y}\| \leq \epsilon_L \|\bar{\nu}\|_1$ is also small. Now, choose $\tilde{L} > 0$ which is similar in magnitude to $\|c + Q\bar{z} + A^T\bar{w} + C^T\bar{y}\|$ and $\|b - A\bar{z}\|$, and set

$$\begin{aligned} a_1 &= -\frac{1}{\tilde{L}}(c + Q\bar{z} + A^T\bar{w} + C^T\bar{y}), \\ a_2 &= -\frac{1}{\tilde{L}}(b - A\bar{z}). \end{aligned}$$

Choose $\hat{c}$ such that

$$\hat{\nu}^0 = \hat{c} - a_1^T\bar{z} - a_2^T\bar{w} - \bar{e}^T\bar{y} \approx \frac{100}{\tilde{L}m}\sum_{i=1}^{m} \bar{\nu}_i\bar{y}_i.$$

Replacing e by $\bar{e}$, we see that the following is a feasible initial point for (3.2)–(3.7):

$$\begin{gathered} z^0 = \bar{z} \quad w^0 = \bar{w}, \quad y^0 = \bar{y}, \\ \nu^0 = d - C\bar{z} + \tilde{L}\bar{e}, \quad \hat{y}^0 = \tilde{L}, \quad \hat{\nu}^0 = \hat{c} - a_1^T\bar{z} - a_2^T\bar{w} - \bar{e}^T\bar{y}. \end{gathered}$$

Moreover, since

$$\hat{y}^0\hat{\nu}^0 = \frac{100}{m}\sum_{i=1}^{m} \nu_i^0 y_i^0,$$

this starting point is not too far from the central path provided that $(\bar{\nu}, \bar{y})$ is not too far away.

This "hot start" technique can be modified for the case in which $(d - C\bar{z})_i$ is slightly non-positive for $i = 1, \cdots, m$ (a situation which arises frequently in practice). When this occurs, $\bar{e}_i$ is usually zero; our strategy is to replace this zero value with a positive value, chosen so that

$$(d + C\bar{z} + \tilde{L}\bar{e})_i = \frac{\tilde{L}}{m}.$$

Finally, we turn to two of the issues that arise in the solution of the linear system (2.12).

When the interior point techniques described thus far are applied to the problem (1.3), the variables and equations can be ordered so that the matrix in (2.12) has a banded structure, and hence the system (2.12) can be solved in $O(N(n_s + n_c)^3)$ operations. However, if augmentation of the problem is necessary as in (3.3)–(3.8), the banded structure is destroyed. This problem can be overcome by forming and factorizing a Schur complement. Specifically, a simple row and column reordering of the augmented version of (2.12) yields:

$$\left[\begin{array}{ccc|c} Q & A^T & C^T & a_1 \\ A & 0 & 0 & -a_2 \\ C & 0 & -(Y^k)^{-1}M^k & -e \\ \hline a_1^T & a_2^T & e^T & -\frac{\hat{\nu}^k}{\hat{y}^k} \end{array}\right] \left[\begin{array}{c} \delta z^k \\ \delta w^k \\ \delta y^k \\ \hline \delta \hat{y}^k \end{array}\right] = \left[\begin{array}{c} 0 \\ 0 \\ -\frac{\Delta_k}{\rho_k}(Y^k)^{-1}e + M^k e \\ \hline -\frac{\Delta_k}{\rho_k}(\hat{y}^k)^{-1} + \hat{\nu}^k \end{array}\right]. \tag{3.12}$$

An elementary argument shows that (3.12) can be solved at the cost of one factorization of the coefficient matrix in (2.12) (the $(1,1)$ block in (3.12)) and two forward- and back-substitution with the resulting factors (each involving a different right-hand side), plus a few vector inner products.

Finally, we turn to the numerical issue of factorizing the matrix in (2.12), given that some of its diagonals $-\nu_i^k/y_i^k$ are unbounded as $k \to \infty$. This problem can be overcome by a simple scaling procedure, which we sketch here. Define a diagonal scaling matrix D^k by

$$D_{ii}^k = \min(1, \sqrt{y_i^k/\nu_i^k}).$$

If "strict complementarity" holds (See Theorem 2.4 for the definition) then as $k \to \infty$,

$$\begin{array}{rcl} \nu_i^* = (d - Cz^*)_i = 0 & \Rightarrow & \lim_k \nu_i^k/y_i^k = 0; \\ \nu_i^* = (d - Cz^*)_i > 0 & \Rightarrow & \lim_k \nu_i^k/y_i^k = \infty. \end{array}$$

Suppose, without loss of generality, that the rows of C and d can be partitioned such that

$$C = \left[\begin{array}{c} C_A \\ C_I \end{array}\right], \quad d = \left[\begin{array}{c} d_A \\ d_I \end{array}\right], \quad C_A z^* = d_A, \quad C_I z^* < d_I.$$

Then, D^k can be partitioned accordingly as

$$D^k = \left[\begin{array}{cc} D_A^k & 0 \\ 0 & D_I^k \end{array}\right] = \left[\begin{array}{cc} I & 0 \\ 0 & D_I^k \end{array}\right] \quad \text{for } k \text{ sufficiently large.}$$

The matrix we actually factorize is then

$$\begin{bmatrix} I & & \\ & I & \\ & & D^k \end{bmatrix} \begin{bmatrix} Q & A^T & C^T \\ A & 0 & 0 \\ C & 0 & -(Y^k)^{-1}M^k \end{bmatrix} \begin{bmatrix} I & & \\ & I & \\ & & D^k \end{bmatrix} \rightarrow \begin{bmatrix} Q & A^T & (C_A)^T & 0 \\ A & 0 & 0 & 0 \\ C_A & 0 & 0 & 0 \\ 0 & 0 & 0 & -I \end{bmatrix}.$$

The latter matrix is then as well-conditioned as the data in the equality-constrained quadratic program

$$\min_z \frac{1}{2} z^T Q z + c^T z, \qquad Az = b, \quad C_A z = d_A$$

will allow. A similar scaling can be applied to the additional diagonal element in the augmented system (3.12).

4. Inexact Sequential Quadratic Programming. Consider now the general nonlinear programming problem

$$\min\ L(z), \qquad h(z) = 0, \qquad g(z) \leq 0, \tag{4.1}$$

where $z \in I\!R^n$, $h \in I\!R^{m_e}$, $g \in I\!R^m$. We write the Lagrangian for (4.1) as

$$\mathcal{L}(z, w, y) = L(z) + w^T h(z) + y^T g(z).$$

If z^* is a solution of (4.1), there exist w^* and y^* such that

$$\frac{\partial \mathcal{L}}{\partial z}(z^*, w^*, y^*) = 0, \qquad h(z^*) = 0, \qquad g(z^*) \leq 0, \qquad y^* \geq 0, \qquad (y^*)^T g(x^*) = 0. \tag{4.2}$$

These are first-order necessary conditions. Without loss of generality, $g(z^*)$ and y^* can be partitioned as

$$g(z^*) = \begin{bmatrix} g_+(z^*) \\ g_0(z^*) \\ g_-(z^*) \end{bmatrix}, \qquad y^* = \begin{bmatrix} y_+^* \\ y_0^* \\ y_-^* \end{bmatrix},$$

where

$$\begin{aligned} g_+(z^*) &= 0, & y_+^* &> 0, \\ g_0(z^*) &= 0, & y_0^* &= 0, \\ g_-(z^*) &< 0, & y_-^* &= 0. \end{aligned}$$

In addition to (4.2), we assume that the following *strong second order sufficiency conditions* hold:

$$\begin{aligned} &\text{For all } \delta_z \neq 0 \text{ with} \qquad \delta_z^T \frac{d\,h}{d\,z}(z^*) = 0, \qquad \delta_z^T \frac{d\,g_+}{d\,z}(z^*) = 0, \\ &\qquad\qquad \text{we have } \delta_z^T \left[\frac{\partial^2 \mathcal{L}}{\partial z^2}(z^*, w^*, y^*) \right] \delta_z > 0. \end{aligned} \tag{4.3}$$

We also assume that the linear independence condition

$$\begin{bmatrix} \frac{d\,h}{d\,z}^T(z^*) \\ \frac{d\,g_+}{d\,z}^T(z^*) \\ \frac{d\,g_0}{d\,z}^T(z^*) \end{bmatrix} \quad \text{has full row rank} \tag{4.4}$$

applies. Given (4.2), (4.3) and (4.4), Robinson [29, Theorem 4.1] shows that (z^*, w^*, y^*) is a *regular solution* of the mixed nonlinear complementarity problem (NLCP)

$$\begin{aligned} \frac{dL}{dz} + \frac{d\,h}{d\,z}^T w + \frac{d\,g}{d\,z}^T y = 0, \\ h(z) = 0, \\ g(z) \le 0, \qquad y \ge 0, \qquad y^T g(z) = 0. \end{aligned} \tag{4.5}$$

("Regularity" is referred to as "strong regularity" in [29].)

A common variant of the sequential quadratic programming algorithm for (4.1) obtains a new iterate $(z^{j+1}, w^{j+1}, y^{j+1})$ from the current iterate (z^j, w^j, y^j) by solving the quadratic program

$$\min \frac{1}{2}\delta_z^T Q^j \delta_z + \delta_z^T c^j, \qquad A^j\delta_z = -h(z^j), \qquad C^j\delta_z \le -g(z^j), \tag{4.6}$$

where

$$Q^j = \frac{\partial^2 \mathcal{L}}{\partial z^2}(z^j, w^j, y^j), \qquad c^j = \frac{dL}{dz}(z^j), \qquad A^j = \frac{d\,h}{d\,z}^T(z^j), \qquad C^j = \frac{d\,g}{d\,z}^T(z^j).$$

Then z^{j+1} is set to $z^j + \delta_z$ and w^{j+1} and y^{j+1} are set to the Lagrange multipliers at the solution of (4.6). The subproblem (4.6) can be restated as a linear complementarity problem

$$\begin{aligned} Q^j\delta_z + (A^j)^T w + (C^j)^T y = -c^j \\ -A^j\delta_z = h(z^j) \\ -C^j\delta_z \ge g(z^j), \qquad y \ge 0, \qquad y^T(C^j\delta_z + g(z^j)) = 0. \end{aligned} \tag{4.7}$$

When Q^j is positive semidefinite, (4.6) has the form (2.1) and the coefficient matrix on the left hand side of (4.7) is positive semidefinite. Pang [25] suggests the following algorithm for solving (4.5) and hence (4.1):

initially:

choose (z^0, w^0, y^0), set $j = 0$

each iteration:

find (δ_z, w, y) such that

$$\|H^j(\delta_z, w, y)\| \le \eta_j \left\| \begin{bmatrix} \frac{dL}{dz}(z^j) + \frac{dh}{dz}(z^j)w^j + \frac{dg}{dz}(z^j)y^j \\ h(z^j) \\ \min(y^j, -g(z^j)) \end{bmatrix} \right\|, \tag{4.8}$$

where

$$H^j(\delta_z, w, y) = \begin{bmatrix} Q^j\delta_z + (A^j)^T w + (C^j)^T y + c^j \\ h(z^j) + A^j\delta_z \\ \min(y, -C^j\delta_z - g(z^j)) \end{bmatrix}; \tag{4.9}$$

set $(z^{j+1}, w^{j+1}, y^{j+1}) = (z^j + \delta_z, w, y)$;
$j \leftarrow j + 1$.

Here, η_j is a scalar to be specified below. Note that $H^j(\delta_z, w, y) = 0$ at an exact solution of (4.6). When Q^j is positive semidefinite, we can use the algorithm of Section 2 to obtain an approximate solution to (4.6), terminating the inner iterations when the criterion (4.8) is satisfied.

The following convergence result can be obtained by making straightforward modifications to Theorem 1 of Pang [25]. Its proof is omitted.

THEOREM 4.1. *Suppose that (z^*, w^*, y^*) is a solution triple for (4.1), that L, h and g are twice continuously differentiable in a neighborhood of z^* and that conditions (4.2), (4.3) and (4.4) are satisfied. Then there is a constant $\eta > 0$ such that*

(i) if $\eta_j \le \eta$ for all j, and if the initial point (z^0, w^0, y^0) is sufficiently close to (z^*, w^*, y^*), then the algorithm (4.8) produces a sequence $\{(z^j, w^j, y^j)\}$ which converges to (z^*, w^*, y^*).

(ii) If, in addition to the assumptions in (i), we have that $\lim_{j\to\infty} \eta_j = 0$, then the convergence of $\{(z^j, w^j, y^j)\}$ is superlinear.

(iii) Suppose, in addition to the assumptions in (i), that the second derivatives of L, h and g satisfy Hölder continuity conditions

$$\begin{aligned} \|\nabla^2 L(z_1) - \nabla^2 L(z_2)\| &\le c_L \|z_1 - z_2\|^\gamma, \\ \|\nabla^2 g_i(z_1) - \nabla^2 g_i(z_2)\| &\le c_g \|z_1 - z_2\|^\gamma, \qquad i = 1, \cdots m, \\ \|\nabla^2 h_i(z_1) - \nabla^2 h_i(z_2)\| &\le c_h \|z_1 - z_2\|^\gamma, \qquad i = 1, \cdots m_e, \end{aligned}$$

for all z_1, z_2 in a neighborhood of z^*, where $\gamma \in (0, 1]$ and c_L, c_g and c_h are positive constants. Then if

$$\eta_j = O\left(\left\| \begin{bmatrix} \frac{dL}{dz}(z^j) + \frac{dh}{dz}(z^j)w^j + \frac{dg}{dz}(z^j)y^j \\ h(z^j) \\ \min(y^j, -g(z^j)) \end{bmatrix} \right\|^\gamma \right),$$

the convergence of $\{(z^j, w^j, y^j)\}$ to (z^*, w^*, y^*) has Q-order $(1 + \gamma)$.

Remarks.

1. In Pang [25], only non-mixed NLCP are considered. For mixed NLCP (i.e., those with equality relations as well as inequalities) the definition of H^j has to be modified as described above, but the same techniques can be used in the convergence analysis.
2. The value of η is specified more precisely in [25]. It depends on the regularity properties of (4.5) at the point (z^*, w^*, y^*) and the size of the neighborhood in which the iterates (z^j, w^j, y^j) are constrained to lie.
3. Pang proves a result like Theorem 4.1 (iii) only for the case $\gamma = 1$. The extension to $\gamma \in (0, 1]$ is immediate.
4. A variant of Theorem 4.1 can be proved for the case in which only the initial z is accurate (i.e., $\|z^0 - z^*\|$ is small, but $\|y^0 - y^*\|$ and $\|w^0 - w^*\|$ may be large). The details are tedious, and we will not pursue this option further.

As already noted, we can use the algorithm of Section 2 to solve (4.6), after making the identifications $b = -h(z^j)$, $d = -g(z^j)$, $Q = Q^j$, $c = c^j$, etc, and $z = \delta_z$. The following Lemma gives an alternative expression for the quantity $\|H^j(\delta_z, w, y)\|$ which is needed in 4.8:

LEMMA 4.2. *If the algorithm of Section 2 is used to solve (4.6) with the identifications described above and if the augmentation (3.2)–(3.8) is used, then at each (inner) iterate* (δ_z^k, w^k, y^k)*, we have*

$$\|H^j(\delta_z^k, w^k, y^k)\| = \left\| \begin{bmatrix} (\hat{y}^k/\tilde{L}) \left[c^j + \frac{\tilde{L}}{m^2} (C^j)^T e \right] \\ (\hat{y}^k/\tilde{L}) h(z^j) \\ \min\left(-C^j \delta_z^k - g(z^j), y^k \right) \end{bmatrix} \right\|.$$

Proof. The result follows immediately from (3.2)–(3.4) and the definitions of a_1 and a_2 if we observe that all iterates of the algorithm of Section 2 are feasible with respect to all constraints in (3.2)–(3.7). ∎

5. Tailoring the algorithm to problems (1.2) and (1.3). We turn now to the optimal control problem (1.3), and describe the special structure of the system (2.12) for this problem. We suppose that $x_i \in \mathbb{R}^{n_s}$, $i = 1, \cdots, N+1$ and $u_i \in \mathbb{R}^{n_c}$, $i = 1, \cdots, N$. For simplicity of exposition, assume that the number of auxiliary constraints at each stage is fixed at m. By introducing adjoint variables p_i, $i = 2, \cdots, N+1$, which are Lagrange multipliers for the state equation, and slack vectors ν_i, $i = 1, \cdots, N+1$ and Lagrange multiplier vectors y_i, $i = 1, \cdots, N+1$ for the auxiliary constraints, we arrive at a set of optimality conditions

that correspond to (2.3)–(2.7):

$$
\begin{aligned}
Q_i x_i + R_i u_i + z_i - A_i^T p_{i+1} + p_i + H_i^T y_i &= 0, \quad i = 2, \cdots, N; \\
Q_{N+1} x_{N+1} + z_{N+1} + p_{N+1} + H_{N+1}^T y_{N+1} &= 0; \\
R_i^T x_i + S_i u_i + r_i - B_i^T p_{i+1} + G_i^T y_i &= 0, \quad i = 1, \cdots, N; \\
x_{i+1} - A_i x_i - B_i u_i &= s_i, \quad i = 1, \cdots, N; \\
G_i u_i + H_i x_i + \nu_i &= g_i, \quad i = 1, \cdots, N; \\
H_{N+1} x_{N+1} + \nu_{N+1} &= g_{N+1}; \\
\nu_i \geq 0, \qquad y_i \geq 0, \quad i &= 1, \cdots, N+1; \\
\nu_i^T y_i &= 0, \quad i = 1, \cdots, N+1.
\end{aligned} \tag{5.1}
$$

The states and controls x_i and u_i correspond to the primal variables z_i in Section 2, while the p_i correspond to w. We now order the variables in the system (2.12) as follows:

$$(\delta u_1, \delta y_1, \delta p_2, \delta x_2, \delta u_2, \delta y_2, \cdots, \delta u_N, \delta y_N, \delta p_{N+1}, \delta x_{N+1}, \delta y_{N+1}).$$

If we order the equations (5.1) similarly, the coefficient matrix in (2.12) has the form:

$$
\begin{bmatrix}
S_1 & G_1^T & -B_1^T & & & & & & & & & & \\
G_1 & -V_1^k & 0 & & & & & & & & & & \\
-B_1 & 0 & 0 & I & & & & & & & & & \\
 & & I & Q_2 & \ddots & & & & & & & & \\
 & & & \ddots & \ddots & & & & & & & & \\
 & & & & & I & & & & & & & \\
 & & & & I & Q_i & R_i & H_i^T & -A_i^T & & & & \\
 & & & & & R_i^T & S_i & G_i^T & -B_i^T & & & & \\
 & & & & & H_i & G_i & -V_i^k & 0 & & & & \\
 & & & & & -A_i & -B_i & 0 & 0 & I & & & \\
 & & & & & & & & I & Q_{i+1} & \ddots & & \\
 & & & & & & & & & \ddots & \ddots & & \\
 & & & & & & & & & & & I & \\
 & & & & & & & & & & I & Q_{N+1} & H_{N+1}^T \\
 & & & & & & & & & & & H_{N+1} & -V_{N+1}^k
\end{bmatrix}, \tag{5.2}
$$

where $V_i^k = \mathrm{diag}((\nu_i^k)_1/(y_i^k)_1, \cdots, (\nu_i^k)_1/(y_i^k)_m)$. This matrix has dimension $N(2n_s + n_c + m) + m$, and bandwidth $(2n_s + n_c + m - 1)$, and hence can be factorized in $O(N(n_s + n_c + m)^3)$ operations. Back- and forward-substitution with the resulting triangular factors takes $O(N(2n_s + n_c + m)^2)$ operations.

For the nonlinear problem (1.2), application of sequential quadratic programming (SQP) gives rise to problems of the form (1.3). This is described in Wright [32] for the case in which auxiliary constraints are absent. The Lagrangian for (1.2) is

$$\mathcal{L}(x, u, p, y) = \bar{\mathcal{L}}(x, u, p) + \sum_{i=1}^{N} y_i^T g_i(x_i, u_i) + y_{N+1}^T g_f(x_{N+1})$$

$$\text{where} \quad \bar{\mathcal{L}}(x,u,p) = \sum_{i=1}^{N} L_i(x_i,u_i) + \phi_N(x_{N+1}) + \sum_{i=1}^{N} p_{i+1}^T (x_{i+1} - f_i(x_i,u_i)).$$

At the current SQP iterate (x,u,p,y), we define

$$Q_i \stackrel{\text{def}}{=} \frac{\partial^2 \mathcal{L}}{\partial x_i^2}, \quad R_i \stackrel{\text{def}}{=} \frac{\partial^2 \mathcal{L}}{\partial x_i \partial u_i}, \quad S_i \stackrel{\text{def}}{=} \frac{\partial^2 \mathcal{L}}{\partial u_i^2},$$
$$z_i \stackrel{\text{def}}{=} \frac{\partial \bar{\mathcal{L}}}{\partial x_i}, \quad r_i \stackrel{\text{def}}{=} \frac{\partial \bar{\mathcal{L}}}{\partial u_i}, \quad s_i \stackrel{\text{def}}{=} \frac{\partial \bar{\mathcal{L}}}{\partial p_{i+1}} \quad g_i \stackrel{\text{def}}{=} -g_i(x_i,u_i),$$
$$A_i \stackrel{\text{def}}{=} \frac{\partial f_i}{\partial x_i}, \quad B_i \stackrel{\text{def}}{=} \frac{\partial f_i}{\partial u_i}, \quad H_i \stackrel{\text{def}}{=} \frac{\partial g_i}{\partial x_i}, \quad G_i \stackrel{\text{def}}{=} \frac{\partial g_i}{\partial u_i},$$

and solve the resulting linear-quadratic problem to obtain the SQP step.

6. Computational results. We present computational results on some problems from the literature. In each case, the problems are discretizations of continuous-time problems. An Euler discretization of problem (1.1) can be obtained by dividing the time interval into N equal intervals, with $t_i = (i-1)/N$, $i = 1, \cdots, N+1$, and finding $x_i \approx x(t_i)$ and $u_i \approx u(t_i)$ by solving

$$\min_{x_i,u_i} \frac{1}{N} \sum_{i=1}^{N} L(x_i,u_i,t_i) + \phi_f(x_{N+1}),$$
$$x_{i+1} = x_i + \frac{1}{N} f(x_i,u_i,t_i), \quad x_1 = x_{\text{init}};$$
$$g(x_i,u_i,t_i) \le 0, \quad g_f(x_{N+1}) \le 0.$$

A solution that is accurate to $O(N^{-1})$ is obtained. A midpoint- or trapezoidal-rule discretization of (1.1), which is accurate to $O(N^{-2})$, has the form (1.2) in some circumstances. We use such a discretization for Example 4 below.

For the inexact SQP algorithm, we use termination criteria that are consistent with (4.8),(4.9) and Theorem 4.1, that is

$$\epsilon_k \stackrel{\text{def}}{=} \left\{ \sum_{i=1}^{N} \left[\left\| \frac{\partial \mathcal{L}}{\partial x_{i+1}} \right\|^2 + \left\| \frac{\partial \mathcal{L}}{\partial u_i} \right\|^2 + \left\| \frac{\partial \mathcal{L}}{\partial p_{i+1}} \right\|^2 + \| \min(y_i, -g_i(x_i,u_i)) \|^2 \right] \right.$$
$$\left. + \| \min(y_{N+1}, g_{N+1}(x_{N+1})) \|^2 \right\}^{1/2} \le 10^{-10} \sqrt{N(2n_s + n_c + m)}$$

(where all quantities are evaluated at the k-th SQP iterate, and the $\min(.,.)$ function applies componentwise.) If the problem is linear-quadratic, this criterion is used to terminate the first (and only) SQP iteration. Otherwise, the interior point algorithm that is used to solve the $(k+1)$-st SQP subproblem is terminated after j iterations if

$$\| H^j(\delta x_i, \delta u_i, p_i, y_i) \| \le \text{mid}(\epsilon_{\min}, \epsilon_k^2, \epsilon_{\max}),$$

where

$$\epsilon_{\max} = 10^{-2}\sqrt{N(2n_s + n_c + m)} \quad \text{and} \quad \epsilon_{\min} = 10^{-10}\sqrt{N(2n_s + n_c + m)}.$$

Here, H^j is defined as in (4.9) to be a measure of inexactness for the linear-quadratic subproblem. According to Theorem 4.1, quadratic convergence should be observed if this criterion is used.

We tested both interior point methods discussed in Sections 2 and 3. As discussed earlier, our implementation of the predictor-corrector method is guaranteed to converge, while our implementation of the potential reduction method is not. The latter is faster in practice, however.

Although the motivation for the algorithms described in this paper has been problems with state constraints and mixed state/control constraints, we first report the results of tests on two problems in which the only constraints are bounds on the controls. Efficient algorithms for this problem are well known, e.g., the two-metric gradient projection algorithms of Bertsekas [1] and Dunn [7] and the second-order DDP algorithms of Jacobson and Mayne [14]. We compare the interior point algorithms with the implementation of the Bertsekas algorithm described in Wright [32], on three examples from the literature. These are Euler discretizations of the following three continuous-time problems:

Example 1: (Bertsekas [1]) $n_s = 2$, $n_c = 1$, $m = 2$.

$$\min \int_0^1 6u(t)^2 + 2x_1(t)^2 + x_2(t)^2 \, dt, \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + u \end{bmatrix},$$
$$|u(t)| \le B, \qquad x_1(0) = 15, \qquad x_2(0) = 5,$$

where B is a positive constant to be specified below. This is a linear-quadratic problem, and hence only one SQP iteration is resquired. The optimal control tends to be on its bound for only a small part of the interval $[0, 1]$.

Example 2: Same as Example 1, but with objective function

$$\min \int_0^1 2x_1(t)^2 + x_2(t)^2 \, dt$$

This problem has a bang-bang solution. When $B = 1$, there is a single switching point from $u(t) = -1$ to $u(t) = 1$ at about $t = .512$. When $B = 2$, the switch from $u(t) = -2$ to $u(t) = 2$ occurs at about $t = .500$.

Example 3: (Jacobson and Mayne [14, Section 2.4.7]) $n_s = 2$, $n_c = 1$, $m = 2$.

$$\min \int_0^{2.5} x_1(t)^2 + u(t)^2 \, dt, \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + 1.4x_2 - .14u^3 + 4u \end{bmatrix},$$
$$|u(t)| \le B, \qquad x_1(0) = -5, \qquad x_2(0) = -5,$$

where, again, $B > 0$ will be specified later. This problem has nonlinear dynamics, and so more than one SQP iteration will usually be required.

In each case, the feasible starting point $u(t) \equiv 0$ is used, with the other quantities being chosen according to the discussion in Section 3. Apart from taking advantage of the feasible initial point and the bandedness, the interior point algorithms did not otherwise exploit the special structure of the constraints during the factorization of the coefficient matrix (5.2), though this would have increased the efficiency. We preferred to use the same codes for all test examples, to demonstrate the versatility of the approach, rather than to customize the algorithm for each class of problems.

Results for these problems are shown in Tables 6.1–6.3. We make the following observations:

- The data reported in each table consists of the number of iterations required for each algorithm, and the amount of CPU time required on a Solbourne 5E/900 workstation. For the interior point algorithms on the nonlinear problems, the number of SQP iterations ("major iterations") is reported along with the total number of interior point iterations ("minor iterations"). For the linear quadratic problems, only one SQP iteration is needed, so we report just the number of interior point iterations. Each SQP iteration requires the evaluation of the objective and constraint functions and their first and second derivatives. Each interior point iteration requires the setup and factorization of a matrix of the form (5.2).
- Iteration counts for the predictor-corrector algorithm do not include the corrector iterations which are sometimes necessary when a "hot start" is used. These corrector iterations are needed to move the initial point closer to the central path, and there are typically only a few of them.
- for the nonlinear problems, the "hot start" strategy of Section 3 was used on all but the first major iteration.
- The factor of 4 to 6 performance difference between the potential reduction algorithm and the predictor-corrector algorithm was typical for all examples that we tried.
- Although the two-metric gradient projection algorithm was clearly fastest for all examples on which it worked, it frequently failed for certain values of the bound B. We believe that this is because of near-degeneracy: at the borderline between regions in which the bound is active and regions in which it is inactive, there are a few components whose active/inactive status is difficult to determine. In the language of Section 2, one of y_i and ν_i is zero and the other is very small. This phenomenon did not appear to affect the performance of the interior point methods.
- no results are given for the two-metric gradient projection algorithm for Example 2. Our implementation of this algorithm always used exact second derivatives in the computation of the second-order part of the step. Global convergence is not assured unless a more complicated strategy is used, and indeed our implementation always failed to make any progress from the initial point on these examples. (Essentially,

TABLE 6.1

Results for Example 1 with $N = 1000$. Second column shows number of active constraints at the solution. For each of the three algorithms, number of iterations and CPU time in seconds on a Solbourne workstation is reported. When $B = 1.4$, two-metric gradient projection terminated prematurely after 5 iterations when $B = 1.40$ with a near-optimal objective function value.

B	# active	two-metric GP iters/time	potential reduction IP/time	predictor-corrector IP/time
1.00	183	3/1.69	16/12.3	76/59.0
1.25	126	4/2.15	16/12.3	72/55.8
1.40	94		17/13.0	74/58.7

TABLE 6.2

Results for Example 2 with $N = 1000$. Problem has a bang-bang solution.

B	potential reduction IP/time	predictor-corrector IP/time
1.0	15/11.7	74/58.6
2.0	13/11.5	72/57.7

for bang-bang problems, the two-metric gradient projection algorithm should reduce to a first-order gradient projection algorithm.)

We turn now to some examples with state and mixed control/state inequality constraints, for which the interior point algorithms were specifically devised. We report results for the following two linear-quadratic examples:

Example 4: $n_s = 4$, $n_c = 1$.

$$\min \sum_{i=1}^{4} x_i^2(4.2),$$

$$\begin{aligned}
\dot{x}_1 &= -0.5x_1 + 5x_2 \\
\dot{x}_2 &= -5x_1 - 0.5x_2 + u \\
\dot{x}_3 &= -0.6x_3 + 10x_4 \\
\dot{x}_4 &= -10x_3 - 0.6x_4 + u \\
x_i(0) &= 10, \qquad i = 1, \ldots, 4, \\
|u(t)| \le 1, &\qquad x_i(4.2) \le 1, \qquad i = 1, \ldots, 4.
\end{aligned}$$

Versions of this problem have been discussed by a number of authors, including Jacobson and Mayne [14, page 85] (who exclude the terminal inequality constraints) and Longsdon [19]. The problem has a bang-bang solution with eight switching times. It is solved in [19] by using a discrete nonlinear programming formulation, and in [14] by using a second-order method

TABLE 6.3

Results for Example 3 with $N = 1000$. Number of active constraints at the solution is reported in the second column. For the interior point algorithms, the number of SQP (major) iterations, number of interior point (minor) iterations, and CPU time on a Solbourne workstation is reported. The two-metric gradient projections algorithm terminates with the objective function at 115% of its final value when $B = 1.0$, and with objective function at 107% of its final value when $B = 0.5$

B	# active	two-metric GP iters/time	potential reduction SQP/IP/time	predictor-corrector SQP/IP/time
1.0	408	fails	6/20/18.5	7/144/135.
0.5	640	fails	5/22/18.8	6/146/127.
2.0	218	7/3.68	7/20/18.1	8/116/116.

TABLE 6.4

Results for Example 4 with different values of the discretization parameter N. Runge-Kutta integration with the final $u(t)$ was used to obtain the final function value. CPU times are in seconds on a Cray Y-MP.

N	final f	potential reduction IP/time	predictor-corrector IP/time
500	1.00423296	15/3.54	80/18.5
2500	1.00357374	17/20.3	110/127.

for problems with control bounds and bang-bang solutions. We discretize the problem by approximating the ODEs by a midpoint rule.

Example 5: Same objective function, dynamics and initial conditions as Example 1, but with constraints:

$$u(t) \leq 0.2, \qquad 3u(t) + 2x_2(t) \leq 4.$$

Results are shown in Tables 6.4 and 6.6. The computations for Example 4 were performed on a Cray Y-MP, which was observed in our experiments to be only about six times faster than the Solbourne workstation. The final function value of $f = 1.00357374$ obtained for the larger version of Example 4 compares with a final value of $f = 1.00347$ obtained by Longsdon [19], who required over 6 hours of CPU time on a VAX 6320 to compute his solution. A comparison of the two solutions appears in Table 6.5. For Example 5, the CPU time required for the potential reduction algorithm is observed to be very nearly proportional to N, while the slow growth in the number of iterations in the predictor-corrector algorithm produces a slightly greater observed complexity. Table 6.7 shows that the number of active constraints at the solution is also $O(N)$, as we would expect.

Finally, we describe experience with two nonlinear problems. These are

Example 6: Van der Pol problem, with state constraint. $n_s = 2$, $n_c = 1$.

$$\min \int_0^5 x_1(t)^2 + x_2(t)^2 + u(t)^2 \, dt, \quad \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} (1 - x_2^2)x_1 - x_2 + u \\ x_1 \end{bmatrix},$$

TABLE 6.5
Comparison of control profiles for computed solutions of Example 4.

Longsdon [19] ($f = 1.00347$)		Interior Point, $N = 2500$ ($f = 1.00357374$)	
Switching time	u	Switching time	u
0.0	-1.0	0.0	-1.0
0.11198	1.0	0.1100	1.0
0.89979	-1.0	0.8980	-1.0
1.36428	1.0	1.3650	1.0
2.16960	-1.0	2.1680	-1.0
2.62063	1.0	2.6200	1.0
3.43619	-1.0	3.4356	-1.0
3.87530	1.0	3.8774	1.0

TABLE 6.6
Results for Example 5 with different values of the discretization parameter N. CPU times are in seconds on a Solbourne workstation.

N	potential reduction IP/time	predictor-corrector IP/time
100	20/2.29	108/11.2
1000	20/23.1	172/192.
10000	19/221.	328/3693.

TABLE 6.7
Active constraints for Example 5

N	100	1000	10000
active $u_i = .2$ components	18	146	1431
active $3u_i + 2x_{i,2} = 4$ components	2	21	207

TABLE 6.8

Results for Example 6 with different values of the discretization parameter N. CPU times are in seconds on a Solbourne workstation.

N	# active	SQP iters	potential reduction inner iters	CPU time
100	42	5	8,11,4,6,2	2.46
1000	415	5	1,6,6,7,7	21.8
10000	4096	4	5,14,16,10	361.

TABLE 6.9

Results for Example 7 with different values of the discretization parameter N. CPU times are in seconds on a Solbourne workstation.

N	# active	SQP iters	potential reduction inner iters	CPU time
100	43	4	8,5,8,1	1.10
1000	428	4	9,8,18	17.4
10000	4294	3	14,16,28	297.

$$x_1(t) \leq -0.3, \qquad t \in [2.5, 5], \qquad x_1(0) = 0, \qquad x_2(0) = 1.$$

From the given starting point, the convexity condition (1.4) does not hold at the solution, but we still observe convergence.

Example 7: (Di Pillo et. al. [4, Example 2]) $n_s = n_c = 1$, $m = 1$

$$\min \int_0^1 x(t)^2 + u(t)^2 \, dt, \qquad \dot{x} = x^2 - u,$$
$$x(t) \geq 0.9, \qquad x(1) \leq 1, \qquad x(0) = 1.$$

We show results for SQP with potential reduction inner iterations in Tables 6.8 and 6.9. In these examples, the computation appears to scale at between $O(N)$ and $O(N^{3/2})$.

7. Conclusions. We have described the use of interior point techniques in discrete-time optimal control problems. The computational results we report in Section 6 for the lienar-quadatic problems appear to be similar in efficiency to those reported by Zhu and Rockafellar [36], who performed computations with randomly generated examples on similar workstation equipment.

Our approach bears some similarity to one described by Ohno [24]. Ohno treats the first-order stationarity conditions for (1.2), and the complementarity conditions $y_i^T g_i(x_i, u_i) = 0$ as a system of nonlinear equations, which are then solved by a differential dynamic programming method that is similar to Newton's method. This is similar to performing only "predictor" iterations in the predictor-corrector algorithm, and we not would expect the esulting algorithm to be robust. In fact, Ohno observes that his algorithm requires a very good initial guess in order to converge. Moreover, as discussed above, our algorithm can be used as the "inner loop" of a more sophisticated algorithm in which approximate first and second derivatives and global convergence strategies can be employed.

REFERENCES

[1] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM Journal on Control and Optimization, 20 (1982), pp. 221–246.

[2] J. E. Cuthrell and L. T. Biegler, *Simultaneous optimization and solution methods for batch reactor control profiles*, Computers chem. Engng., 13 (1989), pp. 49–62.

[3] V. F. Demyanov and A. M. Rubinov, *Approximate Methods in Optimization Problems*, American Elsevier, New York, 1970.

[4] G. Di Pillo, L. Grippo, and F. Lampariello, *A class of structured quasi-Newton algorithms for optimal control problems*, in IFAC Conference on Applications of Nonlinear Programming to Optimization and Control, Proceedings, 1983.

[5] J. C. Dunn, *Global and asymptotic convergence rate estimates for a class of projected gradient processes*, SIAM Journal on Control and Optimization, 19 (1981), pp. 368–400.

[6] ——, *On the convergence of projected gradient processes to singular critical points*, Journal of Optimization Theory and Applications, 55 (1987), pp. 203–216.

[7] ——, *A projected Newton method for minimization problems with nonlinear inequality constraints*, Numerische Mathematik, 53 (1988), pp. 377–409.

[8] J. C. Dunn and D. P. Bertsekas, *Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems*, Journal of Optimization Theory and Applications, 63 (1989), pp. 23–38.

[9] E. R. Edge and W. F. Powers, *Function space quasi-Newton algorithms for optimal control problems with bounded controls and singular arcs*, Journal of Optimization Theory and Applications, 20 (1976), pp. 455–479.

[10] Y. G. Evtushenko, *Numerical Optimization Techniques*, Optimization Software, New York, 1985.

[11] R. Fletcher, *Practical Methods of Optimization (Second Edition)*, John Wiley and Sons, 1987.

[12] E. M. Gafni and D. P. Bertsekas, *Two-metric projection methods for constrained optimization*, SIAM Journal on Control and Optimization, 22 (1984), pp. 936–964.

[13] C.-G. Han, P. Pardalos, and Y. Ye, *Computational aspects of an interior point algorithm for quadratic programming problems with box constraints*, in Large-Scale Numerical Optimization, T. F. Coleman and Y. Li, eds., SIAM Publications, Philadelphia, PA, 1990, pp. 92–112.

[14] D. H. Jacobson and D. Q. Mayne, *Differential Dynamic Programming*, American Elsevier, New York, 1970.

[15] J. Ji, F. Potra, and S. Huang, *A predictor-corrector method for linear complementarity problems with polynomial complexity and superlinear convergence*, Tech. Rep. 18, Department of Mathematics, The University of Iowa, Iowa City, Iowa, August 1991.

[16] C. T. Kelley and E. W. Sachs, *A pointwise quasi-Newton method for unconstrained optimal control problems*, Numerische Mathematik, 55 (1989), pp. 159–176.

[17] M. Kojima, S. Mizuno, and A. Yoshise, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Tech. Rep. Research Report B–217, Department of Information Sciences, Tokyo Institute of Technology, 1988.

[18] ——, *A polynomial-time algorithm for a class of linear complementarity problems*, Mathematical Programming, 44 (1989), pp. 1–26.

[19] J. S. Longsdon, *Efficient determination of optimal control profiles for differential algebraic systems*, PhD thesis, Department of Chemical Engineering, Carnegie-Mellon University, 1990.

[20] O. L. Mangasarian, *Nonlinear Programming*, McGraw-Hill, New York, 1969.

[21] A. Miele, *Recent advances in gradient algorithms for optimal control problems*, Journal of Optimization Theory and Applications, 17 (1975), pp. 361–430.

[22] S. Mizuno, M. Todd, and Y. Ye, *On adaptive-step primal-dual interior point algorithms for linear programming*, Tech. Rep. 944, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY, 1990. To appear in Mathematics of Operations Research.

[23] R. D. C. Monteiro and I. Adler, *Interior path-following primal-dual algorithms. part II: Convex*

quadratic programming, Mathematical Programming, 44 (1989), pp. 43–66.

[24] K. Ohno, *A new approach to differential dynamic programming for discrete-time systems*, IEEE Transactions on Automatic Control, AC-23 (1978), pp. 37–47.

[25] J.-S. Pang, *Inexact Newton methods for the nonlinear complementarity problem*, Mathematical Programming, 36 (1986), pp. 54–71.

[26] J. F. A. d. O. Pantoja and D. Q. Mayne, *Sequential quadratic programming algorithm for discrete optimal control problems with control inequality constraints*, International Journal of Control, 53 (1991), pp. 823–836.

[27] E. Polak, *Computational Methods in Optimization*, Academic Press, New York, 1970.

[28] E. Polak, T. H. Yang, and D. Q. Mayne, *A method of centers based on barrier functions for solving optimal control problems with continuum state and control constraints*, in Proceedings of the 29th Conference on Decision and Control, December 1990, pp. 2327–2332.

[29] S. M. Robinson, *Strongly regular generalized equations*, Mathematics of Operations Research, 11 (1980), pp. 43–62.

[30] R. T. Rockafellar and R. J. Wets, *Generalized linear-quadratic problems of deterministic and stochastic optimal control in discrete time*, SIAM Journal on Control and Optimization, 28 (1990), pp. 810–822.

[31] S. J. Wright, *Partitioned dynamic programming for optimal control*, Tech. Rep. MCS–P173–0890, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, September 1990. To appear in SIAM J. Optimization.

[32] ———, *Solution of discrete-time optimal control problems on parallel computers*, Parallel Computing, 16 (1990), pp. 221–238.

[33] ———, *A parallel algorithm for banded linear systems*, SIAM Journal on Scientific and Statistical Computing, 12 (1991), pp. 824–842.

[34] Y. Ye, R. A. Tapia, and Y. Zhang, *A superlinearly convergent $o(\sqrt{n}l)$-iteration algorithm for linear programming*, Tech. Rep. TR91–22, Department of Mathematical Sciences, Rice University, Houston, TX, 1991.

[35] Y. Zhang, R. A. Tapia, and J. E. Dennis, *On the superlinear and quadratic convergence of primal-dual interior point linear programming algorithms*, Tech. Rep. TR90–6, Department of Mathematical Sciences, Rice University, 1990.

[36] C. Zhu and R. T. Rockafellar, *Primal-dual projected gradient algorithms for extended linear quadratic programming*, tech. rep., Department of Applied Mathematics, University of Washington, Seattle, Washington, August 1990.