PREPRINT ANL/MCS-P664-0597,
MATHEMATICS AND COMPUTER SCIENCE DIVISION, ARGONNE NATIONAL LABORATORY

Application of Interior-Point Methods to Model Predictive Control

Christopher V. Rao*, Stephen J. Wright†, and James B. Rawlings‡

Abstract

We present a structured interior-point method for the efficient solution of the optimal control problem in model predictive control (MPC). The cost of this approach is linear in the horizon length, compared with cubic growth for a naive approach. We use a discrete Riccati recursion to solve the linear equations efficiently at each iteration of the interior-point method, and show that we can expect this recursion to be numerically stable although it was motivated originally by structural rather than numerical considerations. We demonstrate the effectiveness of our approach by solving three process control problems.

1 Introduction

Model predictive control (MPC) is an optimal control-based strategy that uses a plant model to predict the effect of an input profile on the evolving state of the plant. At each step of MPC, an optimal control problem with Bolza objectives is solved and its optimal input profile is implemented until the another plant measurement becomes available. The updated plant information is used to formulate and solve a new optimal control problem—thereby providing feedback from the plant to the model—and the process is repeated. This strategy yields a receding horizon control formulation.

The MPC methodology is appealing to the practitioner because input and state constraints can be explicitly accounted for in the controller. A practical disadvantage is its computational cost, which has tended to limit MPC applications to linear processes with relatively slow dynamics. For such problems, the optimal control problem to be solved at each stage of MPC is a convex quadratic program. While robust and efficient software exists for the solution of unstructured convex quadratic programs, significant improvements can be made by exploiting the structure of the MPC subproblem.

*Department of Chemical Engineering, University of Wisconsin-Madison. rao@bevo.che.wisc.edu

†(Author to whom correspondence should be addressed) Mathematics and Computer Science Division, Argonne National Laboratory. wright@mcs.anl.gov

‡Department of Chemical Engineering, University of Wisconsin-Madison. jbraw@che.wisc.edu.

When input and state constraints are not present, MPC with an infinite horizon is simply the well-known linear quadratic regulator problem. Even when constraints are present, the infinite-horizon MPC problem reduces to a linear quadratic regulator after a certain number of stages (see [3, 18, 22]) and can therefore be recast as a finite-dimensional quadratic program. Since this quadratic program can be large, however, it is important that algorithms be efficient even for long horizons.

Unconstrained linear optimal control problems can be solved by using a discrete Riccati equation. For this approach, the computational cost grows linearly in the horizon length N. A different formulation obtained by eliminating the state variables results in an unconstrained quadratic function whose Hessian is dense, with dimensions that grow linearly in N. The cost of minimizing this quadratic grows cubically with N, making it uncompetitive with the Riccati approach. There is a third option, however—an optimization formulation in which the states are retained as explicit variables and the model equation is retained as a constraint. The optimality conditions for this formulation reveal that the adjoint variables are simply the Lagrange multipliers for the model equation and that the problem can be solved by factoring a matrix whose dimension again grows linearly with N. In this formulation, however, the matrix is banded, with a bandwidth independent of N, so the cost of the factorization grows linearly rather than cubically with N. The discrete Riccati equation can be interpreted as a block factorization scheme for this matrix.

Traditionally, the discrete Riccati equation is obtained by using dynamic programming to solve the unconstrained linear optimal control problem. The essential idea in dynamic programming is to work backwards stagewise through the problem. By tackling the optimization problem in stages, one can redice the optimization problem to a series of simpler subproblems. See Berksekas [2] for further details concerning dynamic programming. In an analogous manner to dynamic programming, block factorization also exploits the multi-staged nature of the optimization problem. The key difference is that the block factorization approach tackles the problem explicitly, whereas dynamic programming tackles the problem semi-implicitly by using Bellman's principle of optimality. It is through the explicit formulation that the block factorization approach retains its inherent structure with the addition of inequality constraints.

When constraints are present, the quadratic program still can be structured in a manner analogous to the unconstrained problem, so that the cost of solving linear equations associated with the problem grows linearly with the horizon length N. This observation has been made by numerous authors, in the context of both active set and interior point methods. Glad and Jonson [8] and Arnold et al. [1] demonstrate that the factorization of a structured Lagrangian in an optimal control problem with a Bolza objective for an active set framework yields a Riccati recursion. Wright [23, 24], Steinbach [21], and Lim et al. [11] investigate the Bolza control problem in an interior-point framework.

In this paper we present an MPC algorithm in which the optimal control subproblems are solved by using this structured formulation. Our work differs from earlier contributions in that the formulation of the optimal control problem is tailored to the MPC application, the interior-point algorithm is based on Mehrotra's algorithm [14] (whose practical efficiency

on general linear and quadratic programming problems is well documented), and the linear system at each interior-point iteration is solved efficiently by a Riccati recursion. We compare our approach with the alternative of using the model equation to eliminate the states, yielding a dense quadratic program in the input variables alone. The two approaches are compared on three large industrial problems.

We now define a few items of notation for use in the remainder of the paper. Given a function $f : [0, \infty) \to [0, \infty)$, we write $f(\delta) = O(\delta)$ if $f(\delta) \leq C\delta$ for some positive constant C and all δ in some region of interest (usually δ very small or very large). We write $f(\delta) = \Omega(\delta)$ if there are positive constants C_1 and C_2 such that $C_1\delta \leq f(\delta) \leq C_2\delta$ for all δ in the region of interest.

We say that a matrix is "positive diagonal" if it is diagonal with positive diagonal elements. The term "nonnegative diagonal" is defined correspondingly. We use SPD as an abbreviation for "symmetric positive definite" and SPSD as an abbreviation for "symmetric positive semidefinite."

2 Model Predictive Control

The fundamental formulation of the linear model predictive controller is the following infinite dimensional convex quadratic program:

$$\min_{u,x} \ \Phi(u, x) = \frac{1}{2} \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T R u_k + \Delta u_k^T S \Delta u_k), \tag{1}$$

subject to the following constraints:

$$x_0 = \hat{x}_j, \qquad x_{k+1} = Ax_k + Bu_k, \tag{2a}$$

$$Du_k \leq d, \qquad G\Delta u_k \leq g, \qquad Hx_k \leq h, \tag{2b}$$

where $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^m$, and $\Delta u_k = u_k - u_{k-1}$. The vector $\hat{x}_j$ represents the current estimate of the state at discrete time j, whereas x_k represents the state at k sampling steps along the future prediction horizon and u_k represents the input at this same time. We assume that Q is an SPSD matrix and that R and S are SPD.

By a suitable adjustment of the origin, the formulation (1), (2) can also account for target tracking and disturbance rejection [15]. If there is a feasible point for the constraints (2), the infinite horizon regulator formulation is stabilizing whenever (A, B) is stabilizable and $(A, Q^{1/2})$ is detectable [20].

By expanding Δu_k, we transform (1),(2) into the following more tractable form:

$$\min_{u,x} \ \Phi(u, x) = \frac{1}{2} \sum_{k=0}^{\infty} (x_k^T Q x_k + u_k^T R u_k + 2x_k^T M u_k), \tag{3}$$

subject to the following constraints:

$$x_0 = \hat{x}_j, \qquad x_{k+1} = Ax_k + Bu_k, \tag{4a}$$

$$Du_k - Gx_k \leq d, \qquad Hx_k \leq h. \tag{4b}$$

The original formulation (1), (2) can be recovered from (3), (4) by making the following substitutions into the second formulation:

$$\hat{x}_j \leftarrow \begin{bmatrix} \hat{x}_j \\ u_{j-1} \end{bmatrix}, \quad x_k \leftarrow \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix}, \quad A \leftarrow \begin{bmatrix} A & 0 \\ 0 & 0 \end{bmatrix}, \quad B \leftarrow \begin{bmatrix} B \\ I \end{bmatrix},$$
$$Q \leftarrow \begin{bmatrix} Q & 0 \\ 0 & S \end{bmatrix}, \quad M \leftarrow \begin{bmatrix} 0 \\ -S \end{bmatrix}, \quad R \leftarrow R + S,$$
$$D \leftarrow \begin{bmatrix} D \\ G \end{bmatrix}, \quad G \leftarrow \begin{bmatrix} 0 & 0 \\ 0 & G \end{bmatrix}, \quad d \leftarrow \begin{bmatrix} d \\ g \end{bmatrix}, \quad H \leftarrow \begin{bmatrix} H & 0 \end{bmatrix}.$$

In the remainder of this section, we address two issues. The first is the replacement of (3), (4) by an equivalent (or similar) finite-dimensional problem, a step necessary for practical computation of the solution. The second issue is replacement of the constraints $Hx_k \leq h$ by so-called soft constraints. Instead of enforcing these conditions strictly, we add terms to the objective that penalize violations of these conditions. This technique is a more appropriate way of dealing with certain constraints from an engineering viewpoint.

2.1 Receding Horizon Regulator Formulation

The key step in reducing (3), (4) to a finite-dimensional problem is the use of a linear control law to determine u_k after a certain time horizon, that is,

$$u_k = Kx_k, \qquad \text{for all } k \geq N. \tag{5}$$

With this added constraint, the states x_k, $k > N$ and the inputs u_k, $k \geq N$ are completely determined by x_N, the state at the end of the prediction horizon.

Two techniques can be used to determine the law (5). The first, due to Rawlings and Muske [16], sets $K = 0$ uniformly in (5) and produces an approximate solution to (3), (4). With this substitution, we have

$$\frac{1}{2}\sum_{k=N}^{\infty}(x_k^T Q x_k + u_k^T R u_k + 2x_k^T M u_k) = \frac{1}{2}\sum_{k=N}^{\infty} x_k^T \bar{Q} x_k. \tag{6}$$

If A is stable, this sum is equal to $(1/2)x_N^T \bar{Q} x_N$, where $\bar{Q}$ is the solution of the matrix Lyapunov equation

$$\bar{Q} - A^T \bar{Q} A = Q. \tag{7}$$

If A is unstable, the sum (6) may be infinite, so we impose a stabilizing constraint to derive any useful information from the solution of the model problem. The Schur decomposition of A can be used to partition A into its stable and unstable subspaces. If this decomposition is

$$A = UTU^T = \begin{bmatrix} U_s & U_u \end{bmatrix} \begin{bmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{bmatrix} \begin{bmatrix} U_s^T \\ U_u^T \end{bmatrix},$$

where the eigenvalues of T_{11} are contained within the unit circle whereas those of T_{22} are contained on or outside of the unit circle, then the orthogonal columns of U span the stable (U_s) and unstable subspaces (U_u) of A [13]. We add the endpoint constraint

$$Fx_N = 0 \qquad (\text{where } F = U_u^T) \tag{8}$$

to ensure that the unstable modes have vanished by stage N. (Since the input u_k is zero for all $k \geq N$, the unstable modes also remain at zero at all subsequent stages.) The evolution of the stable modes on the infinite horizon can be accounted for by solving the following Lyapunov equation for $\bar{Q}$:

$$\bar{Q} - A_s^T \bar{Q} A_s = Q,$$

where $A_s = U_s T_{11} U_s^T$, and replacing the infinite sum with $(1/2)x_N^T \bar{Q} x_N$, as above.

In the second formulation, discussed by Sznaier and Damborg [22], Chmielewski and Manousiouthakis [3], and Scokaert and Rawlings [18], the input after stage N is parameterized with the classical linear quadratic gain K obtained from the solution of the steady-state Riccati equation. This matrix, used in conjunction with the control law (5), is the solution to the "unconstrained" version of the problem, in which the inequality constraints (4b) do not appear. By using (5), the infinite tail of the sum in (3) can be written as

$$\frac{1}{2}\sum_{k=N}^{\infty}(x_k^T Q x_k + u_k^T R u_k + 2x_k^T M u_k) = \frac{1}{2}\sum_{k=N}^{\infty}(x_k^T Q x_k + x_k^T K^T R K x_k + 2x_k^T M K x_k).$$

This infinite summation can be replaced by the single term $(1/2)x_N^T \bar{Q} x_N$, where $\bar{Q}$ is the solution of the following discrete algebraic Riccati equation:

$$\bar{Q} = Q + A^T \bar{Q} A - (A^T \bar{Q} B + M)(R + B^T \bar{Q} B)^{-1}(B^T \bar{Q} A + M^T). \tag{9}$$

In both formulations, the feedback law (5) is valid only if the constraints (4) are satisfied at *all* stages, including the stages $k \geq N$. Hence, we would like to implement this law only after we reach a state x_N such that the solution generated by the control law (5) and the model equation in (4a) at stages $k \geq N$ satisfies the inequalities (4b) at all such stages. We define a set $\mathcal{X}$ of states for which this property holds, as follows:

$$\mathcal{X} = \{x : H(A - BK)^l x \leq h, \;\; (DK - G)(A - BK)^l x \leq d, \;\; \text{for all } l \geq 0\}.$$

where K is the optimal unconstrained linear control law obtained from the following equation:

$$K = -(R + B^T \bar{Q} B)^{-1}(B^T \bar{Q} A + M^T). \tag{10}$$

If N is chosen so that $x_N \in \mathcal{X}$, then the following finite-dimensional problem is equivalent[1] to (3), (4):

$$\min_{u,x} \; \Phi(u,x) = \frac{1}{2}\sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k + 2x_k^T M u_k) + \frac{1}{2}x_N^T \bar{Q} x_N, \tag{11}$$

[1]The finite-dimensional problem is also a valid approximation to (3), (4) for $K = 0$ when the endpoint constraint $Fx_N = 0$ is added to (12) and $\bar{Q}$ is defined by (7)

subject to

$$x_0 = \hat{x}_j, \qquad x_{k+1} = Ax_k + Bu_k, \qquad k = 0, 1, 2, \ldots, \tag{12a}$$

$$u_k = Kx_k, \qquad k \geq N, \tag{12b}$$

$$Du_k - Gx_k \leq d, \quad Hx_k \leq h, \qquad k = 0, 1, 2, \ldots, N-1, \tag{12c}$$

where $\bar{Q}$ is defined in (9).

The set $\mathcal{X}$ is difficult to characterize explicitly because it is defined by an infinite number of conditions. If the components of h and d are strictly positive, however, and the "unconstrained" model is stabilizable, we can show that $\mathcal{X}$ contains 0 in its interior. Under these circumstances, there is an index N_∞ such that

$$x_k \notin \mathcal{X},\ k < N_\infty, \qquad x_k \in \mathcal{X},\ k \geq N_\infty. \tag{13}$$

Since N_∞ is difficult to determine in practice, we can solve the problem (11),(12) for some fixed value of N and then check that the states and inputs at stages $k \geq N$ continue to satisfy the inequality constraints at subsequent stages. If not, we increase N and repeat the process.

A variety of methods guarantee that the constraints are satisfied on the infinite horizon by checking a finite number of stages $k \geq N$. Scokaert and Rawlings [18] propose constructing an open ball B_x contained within the set $\mathcal{X}$, thereby allowing termination of the search when $x_k \in B_x$ for $k \geq N$. The approximation for $\mathcal{X}$ tends to be conservative, however, since the algorithm is motivated by norm-bounding arguments. A more practical method, given by Gilbert and Tan [6], is to construct an output maximal set. This set provides a priori an upper bound l on number of feasible stages $k \in [N, N+l]$ necessary to guarantee that all of the subsequent stages $k \geq l + N$ are feasible. The drawback of this approach is that the algorithm for constructing the maximal sets is not guaranteed to converge for unbounded feasible regions. A third approach, proposed by Rawlings and Muske [16], is to calculate an upper bound l that depends on the state x_N. Since this method is also motivated by norm-bounding arguments, the bound also tends to be conservative. A more complete discussion of handling constraints on the infinite horizon is given by Meadows et al. [13]. For a compact, convex set of states, an alternative approach that circumvents having to check for constraint violations is given by Chmielewski and Manousiouthakis [3]. By examining the extremal points on the feasible region, they calculate a conservative upper bound on the N required to guarantee that the solution is feasible on the infinite horizon.

We have assumed to this point that there exists a feasible solution with respect to the input and endpoint constraints for the optimal control calculation. In the presence of side constraints (2b), it is no longer true that the constrained regulator stabilizes all possible states even when the stabilizability assumption is satisfied. When stabilization is not possible, the problem (3),(4) is an infeasible optimization problem. (In actual operation, an infeasible solution would signal a process exception condition.)

For the Rawlings-Muske formulation, enforcement of the endpoint constraint (8) often results in an infeasible optimization problem. Feasibility can often be recovered by increasing

the horizon length N, but when the initial state is not stabilizable, the feasible region will continue to be empty for all N. The existence of a feasible N can easily be checked by solving the following linear program [13]:

$$\min_{u,x,r} \; e^T r, \tag{14}$$

(where e is the vector whose entries are all 1) subject to the constraints

$$\begin{gathered} x_{k+1} = Ax_k + Bu_k, \qquad k = 0,1,2,\ldots,N-1, \\ Du_k - Gx_k \le d, \quad Hx_k \le h, \qquad k = 0,1,2,\ldots,N-1, \\ r - Fx_N \ge 0, \quad r + Fx_N \ge 0. \end{gathered} \tag{15}$$

A positive solution to the linear program indicates that a feasible solution does not exist and the horizon length N must be increased. If the feasibility check fails for some user supplied upper bound on the horizon length, then current state is not constrained stabilizable for the specified regulator.

2.2 Feasibility and Soft Constraints

In the formulation of the MPC problem, some state constraints are imposed by physical limitations such as valve saturation. Other constraints are less important; they may represent desired ranges of operation for the plant, for instance. In some situations, no set of inputs and states for the MPC problem may satisfy all of these constraints. Rather than having the algorithm declare infeasibility and return without a result, we prefer a solution that enforces some constraints strictly ("hard constraints"), while relaxing others and replacing them with penalties on their violation ("soft constraints").

Scokaert and Rawlings [19] replace the soft constraints with penalty terms in the objective that are a combination of ℓ_1 norms and squared ℓ_2 norms of the constraint violations. Assuming for simplicity that all state constraints $Hx_k \le h$ in (12) are softened in this way, we obtain the following modification to the objective (11):

$$\min_{u,x,\epsilon} \; \Phi(u,x,\epsilon) = \frac{1}{2}\sum_{k=0}^{N-1}(x_k^T Q x_k + u_k^T R u_k + 2x_k^T M u_k + \epsilon_k^T Z \epsilon_k) + z^T \epsilon_k + \frac{1}{2} x_N^T \bar{Q} x_N, \tag{16}$$

where the constraint violations ϵ_k are defined by the following formulae (which replace $Hx_k \le h$):

$$Hx_k - \epsilon_k \le h, \qquad \epsilon_k \ge 0. \tag{17}$$

It is known that when the weighting on the ℓ_1 terms is sufficiently large (see, for example, Fletcher [5]) and when the original problem (11),(12) has a nonempty feasible region, then local minimizers of problem (11), (12) modified by (16), (17) defined above correspond to local solutions of the unmodified problem (11), (12). The modified formulation has the advantage that it can still yield a solution when the original problem (11), (12) is infeasible.

The lower bound on elements of z needed to ensure that $\epsilon_k = 0$ (that is, the original hard constraints $Hx_k \leq h$ are satisfied by the solution) cannot be calculated explicitly without knowing the solution to the original problem (11), (12), since it depends on the optimal Lagrange multipliers for this problem. A conservative state-dependent upper bound for these multipliers can be obtained by exploiting the Lipschitz continuity of the quadratic program [10]. In practice, it usually is not necessary to guarantee that the soft constraints are exact. Therefore, approximate values of z are sufficient for reasonable controller performance.

In the remainder of the paper, we work with a general form of the MPC problem, which contains all the features discussed in this section: finite horizon, endpoint constraints, and soft constraints. This general form is

$$\min_{u,x,\epsilon} \Phi(u,x,\epsilon) = \sum_{k=0}^{N-1} \frac{1}{2}(x_k^T Q x_k + u_k^T R u_k + 2x_k^T M u_k + \epsilon_k^T Z \epsilon_k) + z^T \epsilon_k + x_N^T \bar{Q}_N x_N, \tag{18}$$

subject to

$$\begin{aligned}
x_0 &= \hat{x}_j, \qquad \text{(fixed)} && \text{(19a)}\\
x_{k+1} &= Ax_k + Bu_k, \qquad k = 0,1,\ldots,N-1, && \text{(19b)}\\
Du_k - Gx_k &\leq d, \qquad k = 0,1,\ldots,N-1, && \text{(19c)}\\
Hx_k - \epsilon_k &\leq h, \qquad k = 1,2,\ldots,N, && \text{(19d)}\\
\epsilon_k &\geq 0, \qquad k = 1,2,\ldots,N, && \text{(19e)}\\
Fx_N &= 0. && \text{(19f)}
\end{aligned}$$

3 The Interior-Point Method

In this section, we describe our interior-point-based approach for solving the MPC problem (18), (19). We start with a geneal description of the interior-point method of choice for linear and convex quadratic programming: Mehrotra's predictor-corrector algorithm. The remaining sections describe the specialization of this approach to MPC, including the use of the Riccati approach to sovle the linear subproblem, handling of endpoint constraints, and hot starting.

3.1 Mehrotra's Predictor-Corrector Algorithm

Active set methods have proved to be efficient for solving quadratic programs with general constraints. The interior-point approach has proved to be an attractive alternative when the problems are large and convex. In addition, this approach has the advantage that the system of linear equations to be solved at each iterate has the same dimension and structure throughout the algorithm, making it possible to exploit any structure inherent in the problem. Moreover, the most widely used interior-point algorithms do not require the user to specify a feasible starting point. From a theoretical viewpoint, interior-point

methods exhibit polynomial complexity, in contrast to the exponential complexity of active-set approaches.

In this section, we sketch an interior-point method for general convex quadratic programming problems and discuss its application to the specific problem (18). A more complete description is given by Wright [25].

Consider the following convex quadratic program

$$\min_w \Phi(w) = \frac{1}{2} w^T Q w + c^T w, \quad \text{subject to } Fw = g,\ Cw \le d, \tag{20}$$

where Q is a positive semidefinite symmetric matrix. Let m denote the number of rows in C, the inequality constraint coefficient matrix. The Karush-Kuhn-Tucker (KKT) conditions state that the vector w^* solves this problem if and only if there are vectors π^* and λ^* such that the following conditions are satisfied for $(w, \pi, \lambda) = (w^*, \pi^*, \lambda^*)$:

$$\begin{aligned} Qw + F^T\pi + C^T\lambda + c &= 0, \\ -Fw + g &= 0, \\ -Cw + d &\ge 0, \\ \lambda &\ge 0, \\ \lambda_j(-Cw + d)_j &= 0, \qquad j = 1, 2, \ldots, m. \end{aligned}$$

By introducing a vector t of slacks for the constraint $Cw \le d$, we can rewrite these conditions in a slightly more convenient form:

$$\mathcal{F}(w, \pi, \lambda, t) = \begin{bmatrix} Qw + F^T\pi + C^T\lambda + c \\ -Fw + g \\ -Cw + t + d \\ T\Lambda e \end{bmatrix} = 0, \tag{21a}$$

$$(\lambda, t) \ge 0, \tag{21b}$$

where T and Λ are diagonal matrices defined by

$$T = \mathrm{diag}(t_1, t_2, \ldots, t_m), \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m),$$

and $e = (1, 1, \ldots, 1)^T$ as before.

Primal-dual interior-point methods generate iterates $(w^i, \pi^i, \lambda^i, t^i)$, $i = 1, 2, \ldots$, with $(\lambda^i, t^i) > 0$ that approach feasibility with respect to the conditions (21a) as $i \to \infty$. The search directions are Newton-like directions for the equality conditions in (21a). Dropping the superscript and denoting the current iterate by (w, π, λ, t), we can write the general linear system to be solved for the search direction as

$$\begin{bmatrix} Q & F^T & C^T & \\ -F & & & \\ -C & & & -I \\ & & T & \Lambda \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta\pi \\ \Delta\lambda \\ \Delta t \end{bmatrix} = \begin{bmatrix} r_Q \\ r_F \\ r_C \\ r_t \end{bmatrix}. \tag{22}$$

(Note that the coefficient matrix is the Jacobian of the nonlinear equations (21a).) Different primal-dual methods are obtained from different choices of the right-hand side vector (r_Q, r_F, r_C, r_t). The duality gap μ defined by

$$\mu = \lambda^T t/m \tag{23}$$

is typically used as a measure of optimality of the current point (w, π, λ, t). Most primal-dual interior-point methods ensure that the norm of the function $\mathcal{F}$ defined by (21a) remains bounded by a constant multiple of μ.

We use a variant of Mehrotra's predictor-corrector algorithm [14] to solve (20). This algorithm has proved to be the most effective approach for general linear programs and is similarly effective for convex quadratic programming. The first part of the Mehrotra search direction—the *predictor* or *affine-scaling* step—is simply a pure Newton step for the system (21a), obtained by solving (22) with the following right-hand side:

$$\begin{bmatrix} r_Q \\ r_F \\ r_C \\ r_t \end{bmatrix} = - \begin{bmatrix} Qw + F^T\pi + C^T\lambda + c \\ -Fw + g \\ -Cw + t + d \\ T\Lambda e \end{bmatrix}. \tag{24}$$

We denote the corresponding solution of (22) by $(\Delta w_{\text{aff}}, \Delta\pi_{\text{aff}}, \Delta\lambda_{\text{aff}}, \Delta t_{\text{aff}})$. The second part of the search direction—the *centering-corrector* direction $(\Delta w_{\text{cc}}, \Delta\pi_{\text{cc}}, \Delta\lambda_{\text{cc}}, \Delta t_{\text{cc}})$—is calculated by choosing the centering parameter $\sigma \in [0, 1)$ as outlined below and solving the system (22) with the following right-hand side:

$$\begin{bmatrix} r_Q \\ r_F \\ r_C \\ r_t \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\Delta T_{\text{aff}}\Delta\Lambda_{\text{aff}}e + \sigma\mu e \end{bmatrix}. \tag{25}$$

The following heuristic for choosing the value of σ has proved to be highly effective. We first compute the maximum step length α_{aff} that can be taken along the affine-scaling direction, as follows:

$$\alpha_{\text{aff}} = \arg\max\{\alpha \in [0, 1] \,|\, (\lambda, t) + \alpha(\Delta\lambda_{\text{aff}}, \Delta t_{\text{aff}}) \geq 0\}.$$

The duality gap μ_{aff} attained from this full step to the boundary is

$$\mu_{\text{aff}} = (\lambda + \alpha\Delta\lambda_{\text{aff}})^T(t + \Delta t_{\text{aff}})/m.$$

Finally, we set

$$\sigma = \left(\frac{\mu_{\text{aff}}}{\mu}\right)^3.$$

The search direction is obtained by adding these two components:

$$(\Delta w, \Delta\pi, \Delta\lambda, \Delta t) = (\Delta w_{\text{aff}}, \Delta\pi_{\text{aff}}, \Delta\lambda_{\text{aff}}, \Delta t_{\text{aff}}) + (\Delta w_{\text{cc}}, \Delta\pi_{\text{cc}}, \Delta\lambda_{\text{cc}}, \Delta t_{\text{cc}}). \tag{26}$$

Note that the coefficient matrix in (22) does not need to be refactored to compute the centering-corrector step. Total computational requirements in obtaining the step (26) include one factorization of the matrix in (22), back-substitutions for two different right-hand sides, and a number of matrix-matrix operations.

The distance we move along the direction (26) is defined in terms of the maximum step $\alpha_{\max}$ that can be taken without violating the condition (21b):

$$\alpha_{\max} = \arg\max \{\alpha \in [0,1] \,|\, (\lambda, t) + \alpha(\Delta\lambda, \Delta t) \geq 0\}.$$

The actual steplength α is chosen to be $\gamma\alpha_{\max}$, where γ is a parameter in the range $(0,1)$ chosen to ensure that the pairwise products $\lambda_i t_i$ do not become too unbalanced and the improvement in duality gap does not outpace the improvement in feasibility with respect to the linear constraints in (21a) by too much. The value of γ typically lies between .9 and .9999.

The algorithm does not require the initial point to be feasible, and checks can be added to detect problems for which no feasible points exist. In our case, feasibility of the MPC problem obtained from the Rawlings and Muske formulation with unstable plants can be determined a priori by solving the linear program (14),(15).

Finally, we note that block elimination can be applied to the system (22) to obtain reduced systems with more convenient structures. By eliminating Δt, we obtain the following system:

$$\begin{bmatrix} Q & F^T & C^T \\ -F & & \\ -C & & \Lambda^{-1}T \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta\pi \\ \Delta\lambda \end{bmatrix} = \begin{bmatrix} r_Q \\ r_F \\ r_C + \Lambda^{-1} r_t \end{bmatrix} \stackrel{\text{def}}{=} \begin{bmatrix} \hat{r}_Q \\ \hat{r}_F \\ \hat{r}_C \end{bmatrix}. \tag{27}$$

Since $\Lambda^{-1}T$ is a positive diagonal matrix, we can easily eliminate $\Delta\lambda$ as well to obtain

$$\begin{bmatrix} Q + C^T\Lambda T^{-1} C & F^T \\ -F & 0 \end{bmatrix} \begin{bmatrix} \Delta w \\ \Delta\pi \end{bmatrix} = \begin{bmatrix} r_Q - C^T T^{-1}(\Lambda r_C + r_t) \\ r_H \end{bmatrix}. \tag{28}$$

We conclude with a note on the sizes of elements in t and λ and their effect on elements of the matrices in (27) and (28). In path-following interior-point methods that adhere rigorously to the theory, iterates are confined to a region in which the pairwise products $t_i\lambda_i$ are not too different from each other in size. A bound of the form

$$t_i\lambda_i \geq \gamma\mu \tag{29}$$

is usually enforced, where μ is the average value of $t_i\lambda_i$ (see (23)) and $\gamma \in (0,1)$ is constant, typically $\mu = 10^{-4}$. When the primal-dual solution set for (20) is bounded, we have further that

$$t_i \leq \beta, \quad \lambda_i \leq \beta, \quad i = 1, 2, \ldots, m, \tag{30}$$

for some constant bound $\beta > 0$. It follows immediately from (29) and (30) that

$$\frac{\gamma}{\beta^2}\mu \leq \frac{t_i}{\lambda_i} \leq \frac{\beta^2}{\gamma}\mu^{-1}, \quad \frac{\gamma}{\beta^2}\mu \leq \frac{\lambda_i}{t_i} \leq \frac{\beta^2}{\gamma}\mu^{-1}. \tag{31}$$

Hence, the diagonal elements of the matrices $T^{-1}\Lambda$ and $\Lambda^{-1}T$ lie in the range $[\Omega(\mu), \Omega(\mu^{-1})]$.

Although bounds of the form (29) are not enforced explicitly in most implementations of Mehrotra's algorithm, computational experience shows that they are almost always satisfied in practice. Hence, it is reasonable to assume, as we do in the stability analysis below, that the estimates (31) are satisfied by iterates of our algorithm.

3.2 Efficient MPC Formulation

The optimal control problem (18),(19) has been viewed traditionally as a problem in which just the inputs are variables, while the states are eliminated from the problem by direct substitution using the transition equation (19b) (see, for example, Muske and Rawlings [15]). We refer to this formulation hereafter as the standard method. Unfortunately, the constraint and Hessian matrices in the reduced problem are generally dense, so the computational cost of solving the problem is proportional to N^3. Efficient commercial solvers for dense quadratic programs (such as QPSOL [7]) can then be applied to the reduced problem.

The $O(N^3)$ cost of the standard method is unacceptable because the "unconstrained" version of (18) is known to be solvable in $O(N)$ time by using a Riccati equation or dynamic programming. We are led to ask whether there is an algorithm for the constrained problem (18),(19) that preserves the $O(N)$ behavior. In fact, the interior-point algorithm of the preceding section almost attains this goal. It can be applied to the problem (18),(19) at a cost of $O(N)$ operations per iteration. The rows and columns of the reduced linear systems (27) and (28) can be rearranged to make these matrices *banded*, with dimension $O(N)$ and bandwidth independent of N. Since the number of iterations required by the interior-point algorithm depends only weakly on N in practice, the total computational cost of this approach is only slightly higher than $O(N)$ in practice. In both the active set and interior-point approaches, the dependence of solution time on other parameters, such as input dimension m, control dimension n, and the number of side constraints, is cubic.

Wright [23, 24] describes a scheme in which these banded matrices are explicitly formed and factored with a general banded factorization routine. In the next section, we show that the linear system to be solved at each interior-point iteration can be rearranged to have the same form as an "unconstrained" version of (18), (19), that is, a problem in which the side constraints (19c), (19d) are absent. Hence, a Riccati recursion similar to the technique used for the unconstrained problem can be used to solve this linear system. From a purely linear algebra point of view, this approach is equivalent to a block factorization scheme for the banded coefficient matrix. Because the block elimination scheme restricts the use of pivoting for numerical stability, it is not obvious that the Riccati scheme will produce accurate answers. We alleviate this concern by showing in the next section and in Appendix A that numerical stability of the Riccati approach can be expected under normal circumstances.

Suppose that the interior-point algorithm of Section 3.1 is applied to the problem (18),(19). We use λ_k, ζ_k, and η_k to denote the Lagrange multipliers for the constraints (19c), (19d), and (19e), respectively. We rearrange the linear system (27) to be solved at each iteration of the interior-point method by "interleaving" the variables and equations according to stage

index. That is, the primal and dual variables for stage 0 are listed before those for stage 1, and so on. For this ordering, the rows of the system (27) that correspond to stage k are as follows:

$$\begin{bmatrix} \cdots & Q & M & -G^T & A^T & & & & & \\ & M^T & R & D^T & B^T & & & & & \\ & -G & D & -\Sigma_k^D & & & & & & \\ & A & B & & & & & & -I & \\ & & & & & -\Sigma_{k+1}^{\epsilon} & & -I & & \\ & & & & & & -\Sigma_{k+1}^{H} & -I & H & \\ & & & & & -I & -I & Z & & \\ & & & & -I & & H^T & & Q & \cdots \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta x_k \\ \Delta u_k \\ \Delta \lambda_k \\ \Delta p_{k+1} \\ \Delta \xi_{k+1} \\ \Delta \eta_{k+1} \\ \Delta \epsilon_{k+1} \\ \Delta x_{k+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ r_k^x \\ r_k^u \\ r_k^\lambda \\ r_{k+1}^p \\ r_{k+1}^\xi \\ r_{k+1}^\eta \\ r_{k+1}^\epsilon \\ r_{k+1}^x \\ \vdots \end{bmatrix}. \tag{32}$$

The diagonal matrices Σ_k^D, Σ_k^ϵ, and Σ_k^H, which correspond to $\Lambda^{-1}T$ in the general system (27), are defined by

$$\Sigma_k^D = (\Lambda_k)^{-1} T_k^\lambda, \quad \Sigma_k^\epsilon = (\Xi_k)^{-1} T_k^\xi, \quad \Sigma_k^H = (\mathcal{H}_k)^{-1} T_k^\eta, \tag{33}$$

where Λ_k, Ξ_k, and $\mathcal{H}_k$ are the diagonal matrices whose diagonal elements are the Lagrange multipliers λ_k, ζ_k, and η_k, while T_k^λ, T_k^ξ, and T_k^η are likewise diagonal matrices constructed from the slack variables associated with the constraints (19c), (19d), and (19e), respectively. The final rows in this linear system are

$$\begin{bmatrix} \cdots & Q & M & -G^T & A^T & & & & & \\ & M^T & R & D^T & B^T & & & & & \\ & -G & D & -\Sigma_{N-1}^D & & & & & & \\ & A & B & & & & & & -I & \\ & & & & & -\Sigma_N^{\epsilon} & & -I & & \\ & & & & & & -\Sigma_N^{H} & -I & H & \\ & & & & & -I & -I & Z & & \\ & & & & -I & & H^T & & \bar{Q}_N & F^T \\ & & & & & & & & F & \end{bmatrix} \begin{bmatrix} \vdots \\ \Delta x_{N-1} \\ \Delta u_{N-1} \\ \Delta \lambda_{N-1} \\ \Delta p_N \\ \Delta \xi_N \\ \Delta \eta_N \\ \Delta \epsilon_N \\ \Delta x_N \\ \Delta \beta \end{bmatrix} = \begin{bmatrix} r_{N-1}^x \\ r_{N-1}^u \\ r_{N-1}^\lambda \\ r_{N-1}^p \\ r_{N-1}^\xi \\ r_{N-1}^\eta \\ r_{N-1}^\epsilon \\ r_N^x \\ r^\beta \end{bmatrix}, \tag{34}$$

where β denotes the Lagrange multiplier for the endpoint constraint (19f).

By eliminating the Lagrange multipliers λ_k, ζ_k, and η_k from the systems (32) and (34),

we derive the following analog of the compact system (28):

$$\begin{bmatrix} R_0 & B^T & & & & & & & \\ B & & -I & & & & & & \\ & -I & Q_1 & M_1 & A^T & & & & \\ & & M_1^T & R_1 & B^T & & & & \\ & & A & B & & -I & & & \\ & & & & -I & Q_2 & M_2 & A^T & \\ & & & & & M_2^T & R_2 & B^T & \\ & & & & & A & B & \ddots & \ddots \\ & & & & & & & \ddots & Q_N & F^T \\ & & & & & & & & F & \end{bmatrix} \begin{bmatrix} \Delta u_0 \\ \Delta p_0 \\ \Delta x_1 \\ \Delta u_1 \\ \Delta p_1 \\ \Delta x_2 \\ \Delta u_2 \\ \vdots \\ \Delta x_N \\ \Delta \beta \end{bmatrix} = \begin{bmatrix} \tilde{r}_0^u \\ \tilde{r}_0^p \\ \tilde{r}_1^x \\ \tilde{r}_1^u \\ \tilde{r}_1^p \\ \tilde{r}_2^x \\ \tilde{r}_2^u \\ \vdots \\ \tilde{r}_N^x \\ r^\beta \end{bmatrix}, \tag{35}$$

where

$$\begin{array}{rcll} R_k & = & R + D^T(\Sigma_k^D)^{-1}D & k = 0, \ldots, N-1, \\ M_k & = & M - G^T(\Sigma_k^D)^{-1}D & k = 1, \ldots, N-1, \\ Z_k & = & Z + (\Sigma_k^\epsilon)^{-1} + (\Sigma_k^H)^{-1} & k = 1, \ldots, N, \\ Q_k & = & Q + H^T(\Sigma_k^H)^{-1}H + G^T(\Sigma_k^D)^{-1}G - H^T(\Sigma_k^H Z_k \Sigma_k^H)^{-1}H & k = 1, \ldots, N-1, \\ Q_N & = & \bar{Q}_N + H^T(\Sigma_N^H)^{-1}H - H^T(\Sigma_N^H Z_N \Sigma_N^H)^{-1}H, & \end{array} \tag{36}$$

and

$$\begin{array}{rcll} \tilde{r}_k^u & = & r_k^u + D^T(\Sigma^D)^{-1}r_k^\lambda, & k = 0, \ldots, N-1, \\ \tilde{r}_k^p & = & r_k^p, & k = 0, \ldots, N-1, \\ \tilde{r}_k^z & = & r^\epsilon - (\Sigma_k^\epsilon)^{-1}r_k^\xi - (\Sigma_k^H)^{-1}r_k^\eta, & k = 1, \ldots, N, \\ \tilde{r}_k^x & = & r_k^x + -G^T(\Sigma_k^D)^{-1}r_k^\lambda + H^T(\Sigma_k^H)^{-1}r_k^\eta + H^T(\Sigma_k^H Z_k)^{-1}\tilde{r}_k^z, & k = 1, \ldots, N-1, \\ \tilde{r}_N^x & = & r_N^x + H^T(\Sigma_N^H)^{-1}r_N^\eta + H^T(\Sigma_N^H Z_N)^{-1}\tilde{r}_N^z. & \end{array} \tag{37}$$

This matrix has the same form as the KKT matrix obtained from the following problem in which the only constraint (apart from the model equation and initial state) is a final point condition:

$$\min_{u,x} \ \Phi(u,x) = \sum_{k=0}^{N-1} \frac{1}{2}(x_k^T Q_k x_k + u_k^T R_k u_k + 2x_k^T M_k u_k) + \frac{1}{2}x_N^T Q_N x_N, \tag{38}$$

subject to

$$x_0 = \hat{x}_j, \qquad \text{(fixed)}, \tag{39a}$$

$$x_{k+1} = Ax_k + Bu_k, \qquad k = 0, 1, \ldots, N-1, \tag{39b}$$

$$Fx_N = 0. \tag{39c}$$

Note that R_k is SPD, because it is an SPSD modification of the SPD matrix R. To show that Q_k is SPSD, we show that the modification applied to the SPSD matrices Q and $\bar{Q}_N$ in (36) is itself SPSD. It suffices to show that the sum

$$H^T(\Sigma_k^H)^{-1}H - H^T(\Sigma_k^H Z_k \Sigma_k^H)^{-1}H \tag{40}$$

is SPSD for all k. By writing this sum as

$$H^T(\Sigma_k^H)^{-1/2}\left[I-(\Sigma_k^H)^{1/2}Z_k^{-1}(\Sigma_k^H)^{1/2}\right](\Sigma_k^H)^{-1/2}H,$$

we note that positive definiteness of (40) follows if we can show that the diagonal matrix in brackets has positive diagonal elements. Using the definition of Z_k above, together with diagonality of Z and Σ_k^ϵ, we have that

$$I-(\Sigma_k^H)^{1/2}Z_k^{-1}(\Sigma_k^H)^{1/2}=I-\left(\Sigma_k^H Z+\Sigma_k^H\Sigma_k^\epsilon+I\right)^{-1}. \tag{41}$$

Since Z, Σ_k^H, and Σ_k^ϵ are all diagonal with positive diagonal elements, the matrix (41) has the required property, so that Q_k is symmetric positive semidefinite as claimed.

Since the matrix in (35) is banded with dimension $O(N(m+n))$ and bandwidth $O(m+n)$, the computational cost of factoring is $O(N(m+n)^3)$. This estimate is linear in N, unlike the naive dense implementation for which the cost grows cubically in N.

3.3 Block Elimination Solution: No Endpoint Constraints

We can improve the efficiency of the algorithm by applying a block factorization scheme to (35) in place of the elimination scheme for general banded matrices. In this section, we consider the case in which endpoint constraints are not present in the problem (so that the quantities F, $\Delta\beta$, and r^β do not appear in (35)). We describe a block elimination scheme and show that it yields a Riccati recursion.

For simplicity, we rewrite the system (35) for the case of no endpoint constraints as follows:

$$\begin{bmatrix} R_0 & B^T & & & & & & & \\ B & & -I & & & & & & \\ & -I & Q_1 & M_1 & A^T & & & & \\ & & M_1^T & R_1 & B^T & & & & \\ & & A & B & & -I & & & \\ & & & & -I & Q_2 & M_2 & A^T & \\ & & & & & M_2^T & R_2 & B^T & \\ & & & & & A & B & \ddots & \ddots \\ & & & & & & & \ddots & \bar{Q}_N \end{bmatrix} \begin{bmatrix} \widehat{\Delta u}_0 \\ \widehat{\Delta p}_0 \\ \widehat{\Delta x}_1 \\ \widehat{\Delta u}_1 \\ \widehat{\Delta p}_1 \\ \widehat{\Delta x}_2 \\ \widehat{\Delta u}_2 \\ \vdots \\ \widehat{\Delta x}_N \end{bmatrix} = \begin{bmatrix} \tilde{r}_0^u \\ \tilde{r}_0^p \\ \tilde{r}_1^x \\ \tilde{r}_1^u \\ \tilde{r}_1^p \\ \tilde{r}_2^x \\ \tilde{r}_2^u \\ \vdots \\ \tilde{r}_N^x \end{bmatrix}. \tag{42}$$

Our scheme yields a set of matrices $\Pi_k \in \mathbb{R}^{n\times n}$ and vectors $\pi_k \in \mathbb{R}^n$, $k = N, N-1, \ldots, 1$, such that the following relationship holds between the unknown vectors $\widehat{\Delta p}_{k-1}$ and $\widehat{\Delta x}_k$ in (42):

$$-\widehat{\Delta p}_{k-1}+\Pi_k\widehat{\Delta x}_k=\pi_k, \qquad k=N,N-1,\ldots,1. \tag{43}$$

We can see immediately from (42) that (43) is satisfied for $k = N$ if we define

$$\Pi_N=\bar{Q}_N, \qquad \pi_N=\tilde{r}_N^x. \tag{44}$$

The remaining quantities Π_k and π_k can be generated recursively. If (43) holds for some k, we can combine this equation with three successive block rows from (42) to obtain the following subsystem:

$$\begin{bmatrix} -I & Q_{k-1} & M_{k-1} & A^T & \\ & M_{k-1}^T & R_{k-1} & B^T & \\ & A & B & 0 & -I \\ & & & -I & \Pi_k \end{bmatrix} \begin{bmatrix} \widehat{\Delta p}_{k-2} \\ \widehat{\Delta x}_{k-1} \\ \widehat{\Delta u}_{k-1} \\ \widehat{\Delta p}_{k-1} \\ \widehat{\Delta x}_k \end{bmatrix} = \begin{bmatrix} \tilde{r}_{k-1}^x \\ \tilde{r}_{k-1}^u \\ \tilde{r}_{k-1}^p \\ \pi_k \end{bmatrix}. \tag{45}$$

Elimination of $\widehat{\Delta p}_{k-1}$ and $\widehat{\Delta x}_k$ yields

$$\begin{bmatrix} -I & Q_{k-1} + A^T\Pi_k A & A^T\Pi_k B + M_{k-1} \\ 0 & B^T\Pi_k A + M_{k-1}^T & R_{k-1} + B^T\Pi_k B \end{bmatrix} \begin{bmatrix} \widehat{\Delta p}_{k-2} \\ \widehat{\Delta x}_{k-1} \\ \widehat{\Delta u}_{k-1} \end{bmatrix} = \begin{bmatrix} \tilde{r}_{k-1}^x + A^T\Pi_k\tilde{r}_{k-1}^p + A^T\pi_k \\ \tilde{r}_{k-1}^u + B^T\Pi_k\tilde{r}_{k-1}^p + B^T\pi_k \end{bmatrix}. \tag{46}$$

Finally, elimination of $\widehat{\Delta u}_{k-1}$ yields the equation

$$-\widehat{\Delta p}_{k-2} + \Pi_{k-1}\widehat{\Delta x}_{k-1} = \pi_{k-1}. \tag{47}$$

where

$$\begin{aligned} \Pi_{k-1} &= Q_{k-1} + A^T\Pi_k A - \\ &\qquad (A^T\Pi_k B + M_{k-1})(R_{k-1} + B^T\Pi_k B)^{-1}(B^T\Pi_k A + M_{k-1}^T), && (48a) \\ \pi_{k-1} &= \tilde{r}_{k-1}^x + A^T\Pi_k\tilde{r}_{k-1}^p + A^T\pi_k - \\ &\qquad (A^T\Pi_k B + M_{k-1})(R_{k-1} + B^T\Pi_k B)^{-1}(\tilde{r}_{k-1}^u + B^T\Pi_k\tilde{r}_{k-1}^p + B^T\pi_k). && (48b) \end{aligned}$$

The equation (48a) is the famous discrete Riccati equation for time-varying weighting matrices.

The solution of (42) can now be obtained as follows. By combining (43) for $k = 1$ with the first two rows of (42), we obtain

$$\begin{bmatrix} R_0 & B^T & \\ B & & -I \\ & -I & \Pi_1 \end{bmatrix} \begin{bmatrix} \widehat{\Delta u}_0 \\ \widehat{\Delta p}_0 \\ \widehat{\Delta x}_1 \end{bmatrix} = \begin{bmatrix} \tilde{r}_0^u \\ \tilde{r}_0^p \\ \pi_1 \end{bmatrix}, \tag{49}$$

which can be solved for $\widehat{\Delta u}_0$, $\widehat{\Delta x}_1$, and $\widehat{\Delta p}_0$. Values of $\widehat{\Delta u}_k$, $\widehat{\Delta p}_k$, and $\widehat{\Delta x}_k$ at subsequent stages can be obtained by substituting into (46) and (45) for successively higher values of k. The computational cost of the entire process is $O(N(m+n)^3)$.

The stability of this approach may seem questionable. For one thing, the submatrices R_k, M_k, Q_k, and Z_k contain elements of widely varying magnitude, because the diagonal matrices Σ_k^D, Σ_k^H, and Σ_k^ϵ typically contain both very large and very small elements. Second, the

elimination ordering in this scheme is determined in part by structural rather than numerical considerations, leaving open the possibility of significant element growth as the factorization progresses, a phenomenon that often leads to instability. (Note however that pivoting for numerical stability can occur "internally," during the factorization of $(R_{k-1} + B^T \Pi_k B)$ in (48a) and (48b) for $k = N, N-1, \ldots, 2$.)

If we analyze (48a) by estimating norms of the each of the right-hand side components, we find that $\|\Pi_k\| = O(\mu^{k-N-1})$, indicating severe blowup (and attendant loss of information) as k is decreased. A more careful analysis indicates, however, that we can reasonably expect the estimate $\|\Pi_k\| = O(\mu^{-1})$ to hold independently of k. This observation is consistent with our computational experience. Our discussion of these claims is quite technical, and we relegate it to Appendix A.

3.4 Endpoint Constraints

When endpoint constraints are present in the problem, they must be accounted for by additional recursions. In the language of linear algebra, our approach is to partition the coefficient matrix in (35) in the form

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{12}^T & T_{22} \end{bmatrix},$$

where

$$T_{11} = \begin{bmatrix} R_0 & B^T & & & & \\ B & & -I & & & \\ & -I & Q_1 & M_1 & A^T & \\ & & M_1^T & R_1 & B^T & \\ & & A & B & \ddots & \ddots \\ & & & & \ddots & \bar{Q}_N \end{bmatrix}, \; T_{12} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ F^T \end{bmatrix}, \; T_{22} = 0. \tag{50}$$

We partition the right-hand side and solution of (35) correspondingly and rewrite the system as

$$\begin{bmatrix} T_{11} & T_{12} \\ T_{12}^T & T_{22} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix},$$

where $r_2 = r^\beta$ and $y_2 = \Delta\beta$. By simple manipulation, assuming that T_{11} is nonsingular, we obtain

$$[T_{22} - T_{12}^T T_{11}^{-1} T_{12}] y_2 = r_2 - T_{12}^T T_{11}^{-1} r_1, \tag{51a}$$

$$y_1 = T_{11}^{-1} r_1 - T_{11}^{-1} T_{12} y_2. \tag{51b}$$

We calculate the vector $T_{11}^{-1}r_1$ by using the approach of Section 3.3. The other major operation is to find $T_{11}^{-1}T_{12}$, which we achieve by solving the following system:

$$\begin{bmatrix} R_0 & B^T & & & & & & & \\ B & & -I & & & & & & \\ & -I & Q_1 & M_1 & A^T & & & & \\ & & M_1^T & R_1 & B^T & & & & \\ & & A & B & & -I & & & \\ & & & & -I & Q_2 & M_2 & A^T & \\ & & & & & M_2^T & R_2 & B^T & \\ & & & & & A & B & \ddots & \ddots \\ & & & & & & & \ddots & \bar{Q}_N \end{bmatrix} \begin{bmatrix} \Phi_0^u \\ \Phi_0^p \\ \Phi_1^x \\ \Phi_1^u \\ \Phi_1^p \\ \Phi_2^x \\ \vdots \\ \Phi_N^x \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ F^T \end{bmatrix}. \tag{52}$$

The structure of this system is identical to (42) except that the right-hand side is now a matrix instead of a vector. As in the preceding section, we seek $n \times n_f$ matrices Ψ_k, $k = N, N-1, \ldots, 1$ (where n_f is the number of rows in F) such that the following relationship holds between Φ_{k-1}^p and Φ_k^x satisfying (52):

$$-\Phi_{k-1}^p + \Pi_k \Phi_k^x = \Psi_k, \qquad k = N, N-1, \ldots, 1. \tag{53}$$

(Note that Π_k in (53) are identical to the matrices of this name in the preceding section, since these matrices depend only on the coefficient matrix.) The familiar argument yields the following recursion for Ψ_k:

$$\begin{aligned} \Psi_N &= F^T, \\ \Psi_{k-1} &= A^T\Psi_k - (A^T\Pi_k B + M_{k-1})(R_{k-1} + B^T\Pi_k B)^{-1}B^T\Psi_k, \quad k = N, N-1, \ldots, 2. \end{aligned}$$

We solve (52) by using a similar technique to the one used for (42).

We now recover the solution of (35) via (51). By substituting from (50) and (52), we find that

$$\begin{aligned} T_{22} - T_{12}^T T_{11}^{-1} T_{12} &= -(\Phi_N^x)^T F^T, \\ r_2 - T_{12}^T T_{11}^{-1} r_1 &= r^\beta - F\Delta x_N, \end{aligned}$$

so that $y_2 = \Delta\beta$ can be found directly by substituting into (51a). We recover the remainder of the solution vector from (51b) by noting that

$$T_{11}^{-1}r_1 - T_{11}^{-1}T_{12}y_2 = \begin{bmatrix} \widehat{\Delta u}_0 \\ \widehat{\Delta p}_0 \\ \widehat{\Delta x}_1 \\ \widehat{\Delta u}_1 \\ \widehat{\Delta p}_1 \\ \widehat{\Delta x}_2 \\ \vdots \\ \widehat{\Delta x}_N \end{bmatrix} - \begin{bmatrix} \Phi_0^u \\ \Phi_0^p \\ \Phi_1^x \\ \Phi_1^u \\ \Phi_1^p \\ \Phi_2^x \\ \vdots \\ \Phi_N^x \end{bmatrix} \Delta\beta.$$

In the implementation, the recurrences for computing Π_k, Ψ_k, and π_k take place simultaneously, as do the recurrences needed for solving the systems (42) and (52). The additional cost associated with the n_f endpoint constraints is $O(N(m+n)^2 n_f)$. When $n_f < n$—a necessary condition for (35) to have a unique solution—the cost of solving the full system (35) is less than double the cost of solving the subsystem (42) alone by the method of the preceding section.

3.5 Hot Starting

Model predictive control solves a sequence of similar optimal control problems in succession. If the model is accurate and disturbances are modest, the solution of one optimal control problem can be shifted one time step forward to yield a good approximation to the solution of the next problem in the sequence. Unfortunately, an approximate solution of this type is not a suitable starting guess for the interior-point method, since it usually lies at the boundary of the feasible region. Points close to the so-called central path are much more suitable. In the notation of Section 3.1, the characteristics of such points are that their pairwise products $\lambda_i t_i$ are similar in value for $i = 1, 2, \ldots, m$ and that the ratio of the KKT violations in (21a)—measured by $\mathcal{F}(z, \pi, \lambda, t)$—to the duality gap μ is not too large. We can attempt to find near-central points by bumping components of the "shifted" starting point off their bound. (In the notation of Section 3.1, we turn the zero value of either t_i or λ_i into a small positive value.) A second technique is to use a shifted version of one of the earlier interior-point iterates from the previous problem. Since the interior-point algorithm tends to follow the central path, and since the central path is sensitive to data perturbations only near the solution, this strategy generally produces an iterate that is close to the central path for the new optimal control subproblem.

In the presence of new disturbances, the previous solution has little relevance to the new optimal control problem. A starting point can be constructed from the unconstrained solution, or we can perform a cold start from a well-centered point, as is done to good effect in linear programming codes (see Wright [25, Chapter 10]).

4 Computational Results

To gauge the effectiveness of the structured interior-point approach, we tested it against a naive quadratic programming approach, in which the states x_k are eliminated from the problem (18),(19) by using the model equation (19b). A reduced problem with unknowns u_k, $k = 0, 1, \ldots, N-1$ and ϵ_k, $k = 1, 2, \ldots, n$ is obtained. The reduction in dimension is accompanied by filling in of the constraint matrices and the Hessian of the objective. The resulting problem is solved with the widely used code QPSOL [7], which implements an active set method by using dense linear algebra calculations.

We compared these two approaches on three common applications of the model predictive control methodology.

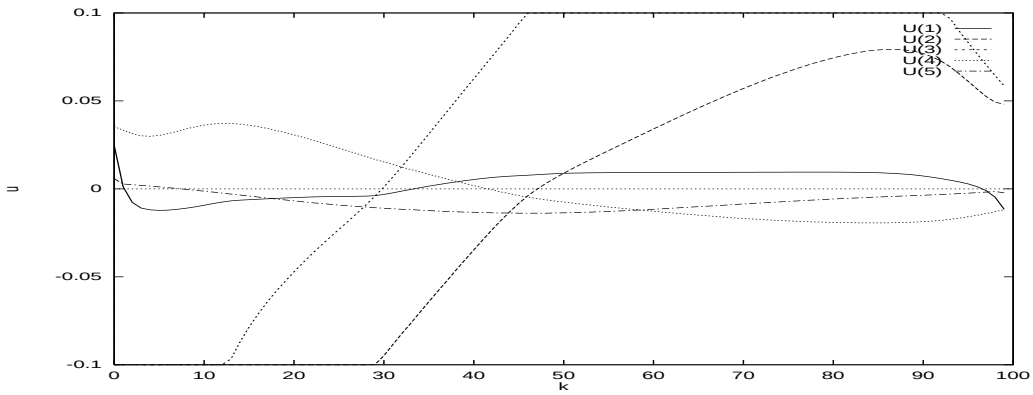

Figure 1: Input Profile at $t = 0$ for Example 1

Example 1: Copolymerization Reactor. Congalidis et al. [4] presented the following normalized model for the copolymerization of methyl methacrylate (MMA) and vinyl acetate (VA) in a continuous stirred tank reactor:

$$G(s) = \begin{bmatrix} \frac{0.34}{0.85s+1} & \frac{0.21}{0.42s+1} & \frac{0.50(0.50s+1)}{12s^2+0.4s+1} & 0 & \frac{6.46(0.9s+1)}{0.07s^2 0.3s+1} \\ \frac{-0.41}{2.41s+1} & \frac{0.66}{1.51s+1} & \frac{-0.3}{1.45s+1} & 0 & \frac{-3.72}{0.8s+1} \\ \frac{0.30}{2.54s+1} & \frac{0.49}{1.54s+1} & \frac{-0.71}{1.35s+1} & \frac{-0.20}{2.71s+1} & \frac{-4.71}{0.008s^2+0.41s+1} \\ 0 & 0 & 0 & 0 & \frac{1.02}{0.07s^2+0.31s+1} \end{bmatrix}. \tag{54}$$

The normalized inputs into the system are the flow of monomer MMA (u_1), flow of monomer VA (u_2), flow of the initiator (u_3), flow of the transfer agent (u_4), and the temperature of the reactor jacket (u_5). The normalized outputs of the systems are the polymer production rate (y_1), mole fraction of MMA in the polymer (y_2), average molecular weight of the polymer (y_3), and reactor temperature (y_4). The model was realized in observer canonical form and discretized with a sample period of 1.

The normalized inputs were constrained to be within 10% of their nominal operating steady–state values The tuning parameters were chosen to be $Q = C^T C$ (C is the measurement matrix obtained from the state space realization), $R = I/10$, $N = 100$. The controller was simulated with the following state disturbance:

$$[x_0]_j = 0.02 * \sin j.$$

Because of the very slow dynamics of the reactor, the Rawlings and Muske formulation was used for this example. Figure 1 shows the optimal control profile normalized with the upper bounds on the input constraints for $t = 0$.

Example 2: Gage Control of a Polymer Film Process. We considered the gage (cross-directional control) of a 26-lane polymer film process with 26 actuators. We used the following model for our simulation:

$$A = 0.9I, \qquad B = (I - A) * K,$$

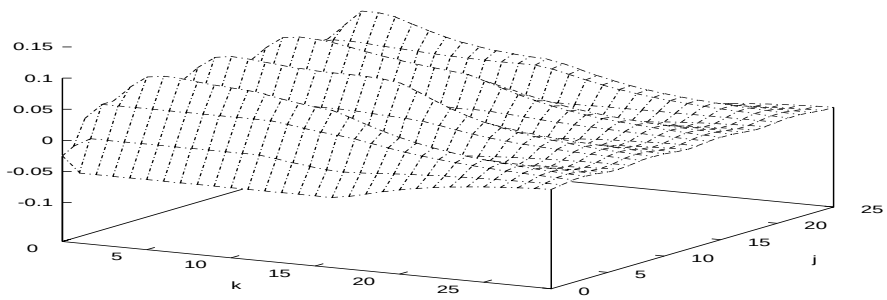

Figure 2: Input Profile at $t = 0$ for Example 2

where the steady-state gain matrix K was extrapolated from data obtained from a 3M polymer film pilot plant. The state $[x]_j$ denotes the deviated film thickness in the jth lane, and the input $[u]_j$ denotes the deviated position of the jth actuator.

The actuators were constrained between the values of 0.1 and -0.1, while the velocity of the actuators was constrained between the values of 0.025 and -0.025. Since a large difference between actuator positions can create excess stress on the die, we imposed the following restriction on the change in input from stage to stage:

$$|[u]_j - [u]_{j-1}| < 0.05, \qquad j = 1, 2, \ldots, m.$$

We chose the tuning parameters to be

$$Q = I, \quad R = I, \quad S = I.$$

We chose a horizon of $N = 30$ to guarantee that the constraints were satisfied on the infinite horizon. Figure 2 shows the calculated optimal input profiles.

Example 3: Evaporator. Ricker et al. [17] presented the following model for an evaporation process in a kraft pulp mill:

$$G(s) = \begin{bmatrix} \frac{1}{30s+1} & 0 \\ \frac{648s}{(30s+1)(20s+1)} & \frac{2.7(-6s+1)}{(20s+1)(5s+1)} \\ \frac{-90s}{(30s+1)(30s+1)} & \frac{-0.1375(-4s+1)}{(30s+1)(2.6s+1)} \end{bmatrix}. \tag{55}$$

The normalized outputs of the process are the feed level (y_1), product concentration (y_2), and product level (y_3). The normalized inputs for the the process are the feed level setpoint (u_1) and the steam flow (u_2). The process was sampled every 0.5 minutes.

Both inputs were constrained to lie in the range $[-0.2, 0.2]$, while the three outputs were constrained to lie in $[-0.05, 0.05]$. A bound of 0.05 was also imposed on the input velocity. The controller was tuned with

$$Q = I, \quad R = I, \quad Z = I, \quad N = 60.$$

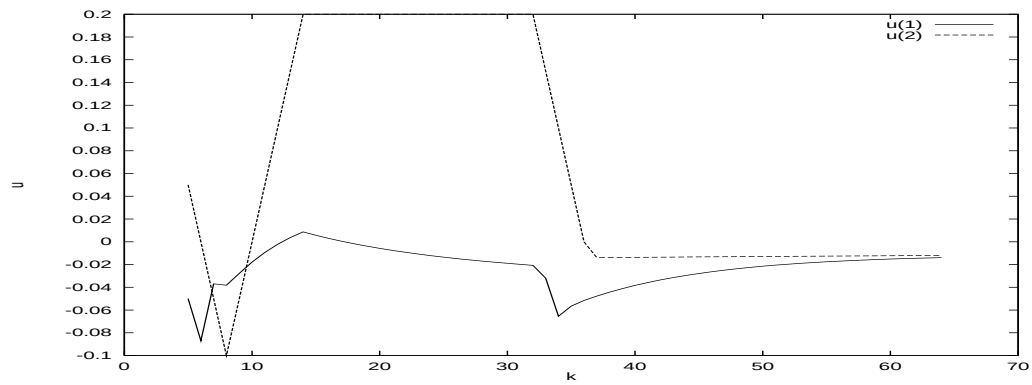

Figure 3: Input Profile at $t = 0$ for Example 3

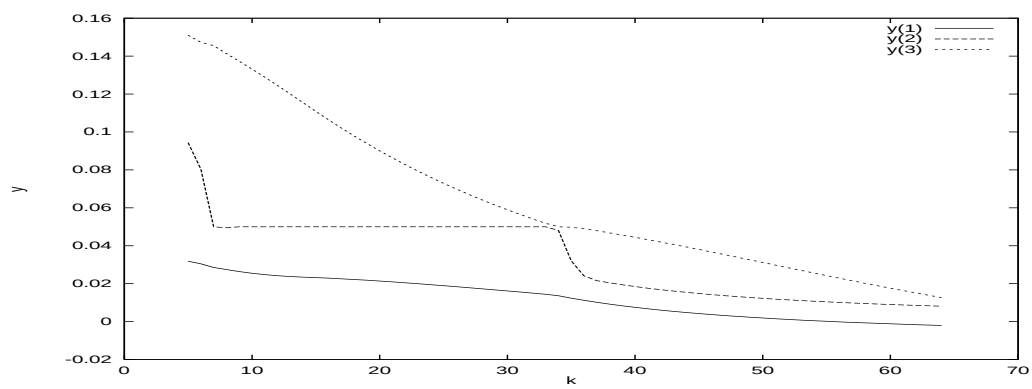

Figure 4: Predicted Output Profile at $t = 0$ for Example 3

A constant l_1 penalty of 1000 was sufficient to force the soft constraints to hold. We simulated the controller with the following state disturbance:

$$[x_0]_j = \sin(j) + \cos(j).$$

Figure 3 shows the calculated optimal input profile, while Figure 4 shows the predicted output profile.

The computational times required by the structured interior-point approach and the naive quadratic programming approach are shown in Table 1. Our platform was a DEC Alphastation 250, and the times were obtained with the Unix `time` command. For the chosen (large) values of the horizon parameter N, the structured interior-point method easily outperforms the naive quadratic programming approach. For the latter approach, we do not include the time required to eliminate the states. These times were often quite significant, but they are calculated offline.

Table 1: Computational Times (sec)

Example	Structured Interior-Point	Naive Quadratic Programming
1	3.80	23.78
2	20.33	276.91
3	2.01	25.32

5 Concluding Remarks

We conclude with three brief comments on the structured interior-point method for MPC. The first is that the structured method presented is also directly applicable to the dual problem of MPC, the constrained receding horizon estimation problem. In fact, the estimation problem will provide greater justification for structured approach because long horizons N are desirable. However, we did not investigate applying the structured optimization approach because the theory for linear constrained receding horizon estimators is still in its infancy.

The second comment is that we can extend the structured method to nonlinear MPC by applying the approach of this paper to the linear-quadratic subproblems generated by sequential quadratic programming. Wright [23], Arnold et al. [1], and Steinbach [21] all apply this technique to discrete-time optimal control problems. While some theory for nonlinear MPC is available, the questions of robust implementation and suitable formulation of nonlinear MPC have not been resolved. See Mayne [12] for a discussion of the some of the issues.

The third comment concerns time delays, which occur when more than one sampling period elapses before an input u_k affects the state of the system. In the simplest case, we can rewrite the state equation (4a) as

$$x_{k+1} = Ax_k + Bu_{k-d}, \tag{56}$$

for the case in which the delay is d sampling periods. The natural infinite-dimensional LQR objective function for this case is

$$\Phi = \frac{1}{2}\sum_{k=0}^{\infty}(x_k^T Q x_k + u_k^T R u_k + 2x_{k+d}^T M u_k), \tag{57}$$

where the cross-penalty terms relates u_{k-d} and x_k. Since the first $(d+1)$ state vectors $x_0, x_1, \ldots, x_d$ are independent of the inputs, the decision variables in the optimization problem are $x_{d+1}, x_{d+2}, \ldots$ and $u_0, u_1, \ldots$. By defining

$$\tilde{x}_k = x_{k+d}, \qquad k = 0, 1, 2, \ldots,$$

and removing constant terms from (57), the objective function and state equation become

$$\begin{gathered}\Phi' = \tfrac{1}{2}\textstyle\sum_{k=0}^{\infty}(\tilde{x}_k^T Q \tilde{x}_k + u_k^T R u_k + 2\tilde{x}_k^T M u_k),\\ \tilde{x}_0 = x_d, \quad \tilde{x}_{k+1} = A\tilde{x}_k + Bu_k, \quad k = 0, 1, 2, \ldots.\end{gathered}$$

These formulae have the same form as (3) and (4).

If no additional constraints of the form (4b) are present, a Riccati equation may be used to solve (58), (58) directly, as in Section 2.1. If state constraints of the form $Hx_k \leq h$ or "jump" constraints of the form $G\Delta u_k \leq g$ (as in (1)) are present, we can still apply constraint softening (Section 2.2) and use the approaches of Rawlings and Muske [16] and Scokaert and Rawlings [18] described in Section 2.1 to obtain finite-dimensional versions of (58), (58). The techniques of Section 3.1 can then be used to solve the problem efficiently.

Difficulties may arise, however, when multiple time delays are present, since these may reduce the locality of the relationships between the decision variables and lead to significant broadening of the bandwidths of the matrices in (32) and (35). A process in which two time delays are present (of d_1 and d_2 sampling intervals) can be described by a state equation of the following form:

$$\begin{bmatrix} x^1_{k+1} \\ x^2_{k+1} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x^1_k \\ x^2_k \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} u^1_{k-d_1} \\ u^2_{k-d_2} \end{bmatrix}.$$

A problem with these dynamics can be solved by augmenting the state vector x_k with the input variables $u_{k-d_1}, u_{k-d_1-1}, \ldots, u_{k-d_2+1}$ (assuming that $d_2 > d_1$) and applying the technique for a single time delay outlined above. Alternatively, the KKT conditions for the original formulation can be used directly as the basis of an interior-point method. The linear system to be solved at each interior-point iteration will contain not only diagonal blocks of the form in (32), but also a number of blocks at some distance from the diagonal. Some rearrangement to reduce the overall bandwidth may be possible, but expansion of the bandwidth by an amount proportional to $(d_2 - d_1)$ is inevitable.

Of course, we can also revert to the original approach of eliminating the states $x_0, x_1, \ldots$ from the problem to obtain a problem in which the inputs $u_0, u_1, \ldots$ alone are decision variables. The cost of this approach, too, is higher than in the no-delay case, because the horizon length N usually must be increased to incorporate the effects of the delayed dynamics. One could postulate that certain processes would be effectively handled by the standard approach while others would be effectively handled by the structured approach. Perhaps the only solution is to exercise engineering judgment to decouple the multiple time scales of the process and consider a set of singularly perturbed control problems. This issue remains unresolved and is a topic of current research.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38, and a grant from Aspen Technology Inc. The authors also acknowledge the support of the industrial members of the Texas-Wisconsin Modeling and Control Consortium.

A Stability of the Riccati Scheme

We return to the solution of the system (35) by the Riccati substitution approach of Section 3.3. In this appendix, we present an informal argument to back up our computational observation that the matrices Π_k arising in the Riccati iteration (48a) do not blow up unduly as k decreases from N to 1.

The singular value decomposition (SVD) of a general real matrix W has the form

$$W = U\Sigma V^T, \tag{58}$$

where U and V are orthonormal matrices and Σ is a nonnegative diagonal matrix. (The diagonals of Σ are the singular values.) When W is symmetric, its SVD has the special form

$$W = U\Sigma U^T,$$

where U is orthogonal and the diagonals of Σ are the eigenvalues of W. Consequently, Σ is nonnegative diagonal if W is SPSD, while Σ is positive diagonal if W is SPD.

A cursory analysis of (48a) does not allay our concern about possible blowup during the Riccati iteration. From the estimates (31), we have that the diagonal elements in Σ_k^D, Σ_k^H, and Σ_k^ϵ lie in the range $[\Omega(\mu), \Omega(\mu^{-1})]$. From the definitions (36), our assumption that R is symmetric positive definite, and the argument above for positive semidefiniteness of Q_k, it follows that

(i) R_k is SPD with eigenvalues in the range $[\Omega(1), \Omega(\mu^{-1})]$;

(ii) the singular values of M_k, $k = 1, \ldots, N-1$ lie in the range $[0, \Omega(\mu^{-1})]$; and

(iii) the matrices Q_k, $k = 1, \ldots, N-1$ and $\bar{Q}_N$ are SPSD with eigenvalues in the range $[0, \Omega(\mu^{-1})]$.

In particular, we have from (44) that Π_N is SPSD with eigenvalues in the range $[0, \Omega(\mu^{-1})]$. By considering the first step of recursion ($k = N$ in (48a)), we find that

$$\begin{aligned} \|Q_{k-1} + A^T\Pi_N A\| &= O(\mu^{-1}), \\ \|A^T\Pi_N B + M_{N-1}\| &= O(\mu^{-1}), \\ \|(R_{N-1} + B^T\Pi_N B)^{-1}\| &= O(1), \end{aligned}$$

so we obtain the estimate

$$\|\Pi_{N-1}\| = O(\mu^{-2}).$$

Continuing inductively in k, we obtain the general estimate $\|\Pi_k\| = O(\mu^{k-N-1})$, which indicates blowup of the kind that destroys accuracy of the solution.

A more refined analysis shows that we can reasonably expect $\|\Pi_k\| = O(\mu^{-1})$ independently of k, indicating stability. We use an informal style of stability analysis here, which suffices to demonstrate the "usual" behavior of the algorithm. (A rigorous analysis would be

overly technical and pessimistic.) We use a recursive argument to show that if $\Pi_k = O(\mu^{-1})$ (with its "large" eigenvalues of size $\Omega(\mu^{-1})$ well separated from the smaller values of size $O(1)$), then under certain assumptions the next matrix Π_{k-1} in the Riccati recurrence will have the same properties. Given Π_k with the specified properties, we can write its SVD as

$$\Pi_k = U_k \Lambda_k U_k^T = U_{k;-1} \Lambda_{k;-1} U_{k;-1}^T + U_{k;0} \Lambda_{k;0} U_{k;0}^T, \tag{59}$$

where U_k is orthogonal, Λ_k is nonnegative diagonal, and $\Lambda_{k;-1}$ contains the diagonals of size $\Omega(\mu^{-1})$ while $\Lambda_{k;0}$ contains the $O(1)$ diagonals.

Consider the matrix $R_{k-1} + B^T \Pi_k B$. Using the definition of R_{k-1} in (36), we partition the diagonal weighting matrix Σ_{k-1}^D into an $\Omega(\mu)$ component $\Sigma_{k-1;1}^D$ and an $\Omega(\mu^{-1})$ component $\Sigma_{k-1;-1}^D$ (see (31)) and partition D correspondingly to obtain

$$D^T (\Sigma_{k-1}^D)^{-1} D = D_{k-1;-1}^T (\Sigma_{k-1;-1}^D)^{-1} D_{k-1;-1} + D_{k-1;1}^T (\Sigma_{k-1;1}^D)^{-1} D_{k-1;1}. \tag{60}$$

Using (36) and (59), we can write

$$\begin{aligned} R_{k-1} + B^T \Pi_k B &= R + D^T (\Sigma_k^D)^{-1} D + B^T U_k \Lambda_k U_k^T B \\ &= \tilde{R}_{k-1} + \left[D_{k-1;1}^T \,|\, B^T U_{k;-1} \right] \begin{bmatrix} (\Sigma_{k-1;1}^D)^{-1} & 0 \\ 0 & \Lambda_{k;-1} \end{bmatrix} \begin{bmatrix} D_{k-1;1} \\ U_{k;-1}^T B \end{bmatrix}, \end{aligned} \tag{61}$$

where

$$\tilde{R}_{k-1} = R + \left[D_{k-1;-1}^T \,|\, B^T U_{k;0} \right] \begin{bmatrix} (\Sigma_{k-1;-1}^D)^{-1} & 0 \\ 0 & \Lambda_{k;1} \end{bmatrix} \begin{bmatrix} D_{k-1;-1} \\ U_{k;0}^T B \end{bmatrix}.$$

Since R is SPD and $(\tilde{R}_{k-1} - R)$ is SPSD with size $O(\mu)$, it follows that all eigenvalues of $\tilde{R}_{k-1}$ are $\Omega(1)$. Meanwhile, we can write the SVD of the second term in (61) as

$$\left[D_{k-1;1}^T \,|\, B^T U_{k;-1} \right] \begin{bmatrix} (\Sigma_{k-1;1}^D)^{-1} & 0 \\ 0 & \Lambda_{k;-1} \end{bmatrix} \begin{bmatrix} D_{k-1;1} \\ U_{k;-1}^T B \end{bmatrix} = \hat{V}_{k-1;-1} \hat{\Lambda}_{k-1;-1}^R \hat{V}_{k-1;-1}^T, \tag{62}$$

where $\hat{V}_{k-1}$ is an orthonormal matrix that satisfies

$$\text{range}\left(\left[D_{k-1;1}^T \,|\, B^T U_{k;-1} \right] \right) = \text{range}(\hat{V}_{k-1;-1}). \tag{63}$$

We assume that $[D_{k-1;1}^T \,|\, B^T U_{k;-1}]$ has full rank for all k, in which case all diagonals of the singular value matrix $\hat{\Lambda}_{k-1;-1}^R$ are $\Omega(\mu^{-1})$ in size.

It follows from (62) and (58) that there is an orthogonal matrix $\hat{W}_{k-1}$ such that

$$\hat{V}_{k-1;-1} (\hat{\Lambda}_{k-1;-1}^R)^{1/2} \hat{W}_{k-1}^T = \left[D_{k-1;1}^T \,|\, B^T U_{k;-1} \right] \begin{bmatrix} (\Sigma_{k-1;1}^D)^{-1/2} & 0 \\ 0 & \Lambda_{k;-1}^{1/2} \end{bmatrix}. \tag{64}$$

For convenience, we use $\hat{V}_{k-1;0}$ to denote the orthonormal matrix that spans the null space of $\hat{V}_{k-1;-1}^T$; that is, $[\hat{V}_{k-1;-1} \,|\, \hat{V}_{k-1;0}]$ is orthonormal.

We can view the full matrix in (61) as a positive definite perturbation of an SPSD matrix whose nonzero singular values are all $\Omega(\mu^{-1})$ in size. A standard eigenspace perturbation result (see, for example, Golub and Van Loan [9, Theorem 7.2.4]) then implies that for μ sufficiently small, we have

$$R_{k-1} + B^T \Pi_k B = [V_{k-1;-1} \,|\, V_{k-1;0}] \begin{bmatrix} \Lambda_{k-1;-1} & 0 \\ 0 & \Lambda_{k-1;0} \end{bmatrix} \begin{bmatrix} V_{k-1;-1}^T \\ V_{k-1;0}^T \end{bmatrix}, \tag{65}$$

where there are matrices P_{-1} and P_0 such that

$$\|P_{-1}\| = O(\mu), \quad \|P_0\| = O(\mu), \tag{66}$$

$$V_{k-1;-1} = (\hat{V}_{k-1;-1} + \hat{V}_{k-1;0} P_{-1})(I + P_{-1}^T P_{-1})^{-1/2}, \tag{67}$$

$$V_{k-1;0} = (\hat{V}_{k-1;0} + \hat{V}_{k-1;-1} P_0)(I + P_0^T P_0)^{-1/2}. \tag{68}$$

In addition, we have that $\Lambda_{k-1;-1}$ is positive diagonal with all diagonals of size $\Omega(\mu^{-1})$, while $\Lambda_{k-1;0}$ is positive diagonal with all diagonals of size $\Omega(1)$. Using (68) and orthonormality of $\hat{V}_{k-1;-1}$, we have that

$$\hat{V}_{k-1;-1}^T V_{k-1;0} = P_0 (I + P_0^T P_0)^{-1/2},$$

and therefore from (66) we obtain

$$\|\hat{V}_{k-1;-1}^T V_{k-1;0}\| = O(\mu). \tag{69}$$

We can use the SVD (65) to immediately write the inverse of the matrix in question as

$$(R_{k-1} + B^T \Pi_k B)^{-1} = [V_{k-1;-1} \,|\, V_{k-1;0}] \begin{bmatrix} \Lambda_{k-1;-1}^{-1} & 0 \\ 0 & \Lambda_{k-1;0}^{-1} \end{bmatrix} \begin{bmatrix} V_{k-1;-1}^T \\ V_{k-1;0}^T \end{bmatrix}, \tag{70}$$

where from the discussion above we have

$$\|\Lambda_{k-1;-1}^{-1}\| = O(\mu), \qquad \|\Lambda_{k-1;0}^{-1}\| = O(1). \tag{71}$$

We turn now to another term in (48a), namely, the matrix $B^T \Pi_k A + M_{k-1}^T$. From (36), we have

$$B^T \Pi_k A + M_{k-1}^T = M^T - D^T (\Sigma_{k-1}^D)^{-1} G + B^T \Pi_k A.$$

By decomposing similarly to (61), we obtain

$$B^T \Pi_k A + M_{k-1}^T = \tilde{M}_{k-1}^T + \left[D_{k-1;1}^T \,|\, B^T U_{k;-1} \right] \begin{bmatrix} (\Sigma_{k-1;1}^D)^{-1} & 0 \\ 0 & \Lambda_{k;-1} \end{bmatrix} \begin{bmatrix} -G_{k-1;1} \\ U_{k;-1}^T A \end{bmatrix}, \tag{72}$$

where

$$\tilde{M}_{k-1}^T = \left[D_{k-1;-1}^T \,|\, B^T U_{k;0} \right] \begin{bmatrix} (\Sigma_{k-1;-1}^D)^{-1} & 0 \\ 0 & \Lambda_{k;1} \end{bmatrix} \begin{bmatrix} -G_{k-1;-1} \\ U_{k;0}^T A \end{bmatrix} + M^T.$$

By substituting (64) into (72), we obtain

$$\begin{aligned} B^T\Pi_k A + M_{k-1}^T &= \tilde{M}_{k-1}^T + \hat{V}_{k-1;-1}(\hat{\Lambda}_{k-1;-1}^R)^{1/2}\hat{W}_{k-1}^T \begin{bmatrix} -(\Sigma_{k-1;1}^D)^{-1/2}G_{k-1;1} \\ \Lambda_{k;-1}^{1/2}U_{k;-1}^T A \end{bmatrix} \\ &= \tilde{M}_{k-1}^T + \hat{V}_{k-1;-1}O(\mu^{-1}). \end{aligned} \tag{73}$$

It follows from (69) that

$$V_{k-1;0}^T(B^T\Pi_k A + M_{k-1}^T) = V_{k-1;0}^T\tilde{M}_{k-1}^T + V_{k-1;0}^T\hat{V}_{k-1;-1}O(\mu^{-1}) = O(1), \tag{74}$$

while from orthonormality of $V_{k-1;-1}$ and $\hat{V}_{k-1;-1}$, we have

$$V_{k-1;-1}^T(B^T\Pi_k A + M_{k-1}^T) = V_{k-1;-1}^T\tilde{M}_{k-1}^T + V_{k-1;-1}^T\hat{V}_{k-1;-1}O(\mu^{-1}) = O(1) + O(\mu^{-1}). \tag{75}$$

By combining (70) with (74) and (75), and using the estimates (71), we obtain

$$(A^T\Pi_k^T B + M_{k-1})(R_{k-1} + B^T\Pi_k B)^{-1}(B^T\Pi_k A + M_{k-1}^T) = O(1) + O(\mu^{-1}). \tag{76}$$

It is easy to see that the first term in (48a), namely, $Q_{k-1} + A^T\Pi_k A$, is also an SPSD matrix with eigenvalues in the range $[0, \Omega(\mu^{-1})]$. We conclude that

$$\|\Pi_{k-1}\| = \Omega(\mu^{-1}), \tag{77}$$

as claimed.

References

[1] E. Arnold, P. Tatjewski, and P. Wolochowicz. Two methods for large-scale nonlinear optimization and their comparison on a case study of hydropower optimization. *Journal of Optimization Theory and Applications*, 81(2):221–248, 1994.

[2] D. P. Bertsekas. *Dynamic Programming.* Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1987.

[3] D. Chmielewski and V. Manousiouthakis. On constrained infinite-time linear quadratic optimal control. *System & Control Letters*, 29:121–129, 1996.

[4] J. B. Congalidis, J. R. Richards, and W. H. Ray. Modeling and control of a copolymerization reactor. In *Proceedings of the American Control Conferance*, pages 1779–1793, Seattle, Washington, June 1986.

[5] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, New York, 1987.

[6] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, September 1991.

[7] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. User's guide for SOL/QPSOL: A Fortran package for quadratic programming, technical report SOL 83-12. Technical report, Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1983.

[8] T. Glad and H. Jonson. A method for state and control constrained linear quadratic control problems. In *Proceedings of the 9th IFAC World Congress*, Budapest, Hungary, July 1984.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, Maryland, 1983.

[10] W. W. Hager. Lipschitz continuity for constrained processes. *SIAM Journal on Control and Optimization*, 17(3):321–338, 1979.

[11] A. Lim, J. Moore, and L. Faybusovich. Linearly constrained LQ and LQG optimal control. In *Proceedings of the 13th IFAC World Congress*, San Francisco, California, July 1996.

[12] D. Q. Mayne. Nonlinear model predictive control: An assessment. In J. C. Kantor, editor, *Chemical Process Control – V*. CACHE, 1996.

[13] E. S. Meadows, K. R. Muske, and J. B. Rawlings. Implementable model predictive control in the state space. In *Proceedings of the 1995 American Control Conference*, pages 3699–3703, 1995.

[14] S. Mehrotra. On the implementation of a primal-dual interior point method. *SIAM Journal on Optimization*, 2:575–601, 1992.

[15] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[16] J. B. Rawlings and K. R. Muske. Stability of constrained receding horizon control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, October 1993.

[17] N. L. Ricker, T. Subrahmanian, and T. Sim. Case studies of model-predictive control in pulp and paper production. In T. J. McAvoy, Y. Arkun, and E. Zafiriou, editors, *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, pages 13–22, Oxford, 1988. Pergamon Press.

[18] P. O. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. Accepted for publication in *IEEE Transactions on Automatic Control*, December 1995.

[19] P. O. Scokaert and J. B. Rawlings. Feasibility issues in model predictive control. Submitted to *IEEE Transactions on Automatic Control*, January 1996.

[20] P. O. Scokaert and J. B. Rawlings. Infinite horizon linear quadratic control with constraints. In *Proceedings of the 13th IFAC World Congress*, San Francisco, California, July 1996.

[21] M. C. Steinbach. A structured interior point SQP method for nonlinear optimal control problems. In R. Burlirsch and D. Kraft, editors, *Computational Optimal Control*, pages 213–222. Birkhäuser, 1994.

[22] M. Sznaier and M. J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Proceedings of the 26th Conference on Decision and Control*, pages 761–762, 1987.

[23] S. J. Wright. Interior-point methods for optimal control of discrete-time systems. *Journal of Optimization Theory and Applications*, 77:161–187, 1993.

[24] S. J. Wright. Applying new optimization algorithms to model predictive control. In J. C. Kantor, editor, *Chemical Process Control-V*. CACHE, 1996.

[25] S. J. Wright. *Primal-Dual Interior-Point Methods*. SIAM Publications, Philadelphia, Penn., 1997.