Feasibility Issues in Linear Model Predictive Control

Pierre O. M. Scokaert and James B. Rawlings
Dept. of Chemical Engineering, University of Wisconsin-Madison, 1415 Engineering Drive, Madison, WI 53706

The problem of infeasibility caused by inconsistent hard state constraints in model predictive control is considered. The difficulties that may arise are analyzed, and the inherent trade-off a controller must address at times of infeasibility is shown. From a multiobjective viewpoint, it is demonstrated that the two approaches for handling infeasibility documented in the literature have significant limitations. Two new techniques for handling infeasibility that overcome the difficulties encountered with current approaches are presented.

Introduction

One of the primary reasons for the industrial success of model predictive control (MPC) is that hard constraints can be enforced on the process. We distinguish input constraints from state or output constraints. The input constraints arise from physical limitations of the actuators such as valve saturations, and it is always advantageous to enforce them in the control law. State constraints, on the other hand, are often not due to physical limitations, but rather they are included to maintain the process in a desired operating regime. Furthermore, and more importantly, they often cannot be strictly enforced at all times. In the presence of disturbances, for example, it is not always possible for the controller to enforce the state constraints. The goal of this article is therefore to provide a framework for allowing state constraints in a reliable and tunable control law that enforces these constraints when they are feasible and relaxes them in a clear way when they are not. The need for clear tuning guidelines for relaxing state constraints becomes especially important for large multivariable problems with many different types of state constraints that interact with each other in nonintuitive ways. A current trend in industrial process control is toward such large, multivariable, constrained problems.

Early attempts to incorporate state constraints within finite-horizon MPC were unreliable because state constraints can cause closed-loop instability regardless of how the regulator tuning parameters are chosen (Zafiriou and Marchal, 1991). Garcia and Morshedi (1986) also provide an early discussion of some of the undesirable side effects of satisfying hard constraints in finite-horizon MPC. As discussed in Rawlings and Muske (1993), detecting and resolving infeasibilities on-line becomes the new issue, and that paper presents one simple stabilizing strategy.

We consider the two approaches documented in the literature for achieving feasibility, which we call the minimum-time approach and the soft-constraint approach. We assume the system can be represented with a linear, discrete-time, state-space model

$$x_{t+1} = Ax_t + Bu_t$$

and consider a standard infinite-horizon MPC objective function

$$\phi(x_t, \pi_t) = \sum_{j=t}^{\infty} x'_{j|t} Q x_{j|t} + u'_{j|t} R u_{j|t}$$

in which $x_{j|t}$ denotes prediction at current time t of state at time j in the horizon, so $x_t = x_{t|t}$ is the current state. The parameters $Q \geq 0$ and $R > 0$ are symmetric weighting matrices, with $(Q^{1/2}, A)$ detectable and

$$\pi_t = \{u_{t|t}, u_{t+1|t}, \dots, u_{t+N-1|t}\}$$

Correspondence concerning this article should be addressed to J. B. Rawlings.
Current address of P. O. M. Scokaert: Centre National d'Etudes des Telecommunications, France Telecom, Paris, France.

is the sequence of control moves to be optimized, with N as the control horizon, and $u_{j|t} = 0$ for $j \geq t + N$. The MPC control law is defined by implementing only the first move of this sequence and then re-solving the optimization with the new x_{t+1} measurement as the initial condition at $t+1$. We assume throughout the article that the state is accessible and do not address the problem of output feedback.

Minimal time approach

Rawlings and Muske (1993) discuss a control algorithm that identifies the smallest time, $\kappa(x)$, which depends on the current state x, beyond which the state constraint can be satisfied on an infinite horizon. Prior to time κ, the state constraint is ignored, and the control law enforces the state constraint only after that time. For suitably restricted plants, Rawlings and Muske show that $\kappa(x)$ is finite for bounded x. Furthermore, $\kappa(x)$ can nominally be reduced by at least one at each sample; it follows that in the nominal case, at most $\kappa(x_0)$ state-constraint violations are observed in closed loop. An advantage of the method is that it leads to the earliest possible constraint satisfaction. The control law can also be shown to be nominally stabilizing, and its closed-loop behavior approaches the open-loop predictions for large N. Disadvantages are that transient constraint violations can be large, calculation of $\kappa(x)$ requires the solution of a sequence of linear programs, and the control law can be discontinuous in the state. Discontinuous control laws raise questions because it is difficult to establish that they are stable under perturbations. Stability under perturbations is normally required to establish that small disturbances, which are inevitable in feedback control systems, do not destabilize a nominally stable closed-loop system.

Establishing perturbed stability for the closed-loop system, on the other hand, is a strong result because it guarantees a form of separation of state regulation and estimation: any asymptotically stable state estimator coupled with the MPC regulator remains asymptotically stable (Scokaert et al., 1997).

Soft-constraint approach

A soft-constraint approach to handle infeasibility is discussed in (Ricker et al., 1988; Zheng and Morari, 1995). In soft-constraint MPC, violations of the state constraints are allowed, but an additional term is introduced in the objective, which penalizes constraint violations. In Zheng and Morari (1995), the added term in the cost is the square of the maximum violation over the horizon, weighted by a constant, S. They formulate the soft-constraint MPC approach as the solution to the following quadratic program:

$$\min_{\pi_t, \epsilon_t} \phi(x_t, \pi_t, \epsilon_t)$$

$$\text{Subject to} \begin{cases} x_{j+1|t} = Ax_{j|t} + Bu_{j|t}, & \\ Hx_{j|t} \leq h + \epsilon_t & j \leq t + \kappa(x_t), \\ Hx_{j|t} \leq h & t + \kappa(x_t) < j, \\ Du_{j|t} \leq d & t \geq j, \\ u_{j|t} = 0 & t + N \leq j, \\ \epsilon_t \geq 0, & \end{cases}$$

where

$$\phi(x_t, \phi_t, \epsilon_t) = \sum_{j=t}^{\infty} x'_{j|t} Q x_{j|t} + u'_{j|t} R u_{j|t} + \epsilon'_t S \epsilon_t.$$

Setting S to zero effectively removes the state constraints; increasing S leads to increasingly ''hard'' constraints. An advantage of the method is that it is computationally cheap, requiring the solution of a single quadratic program. The soft-constraint MPC control is also nominally stabilizing and Lipschitz continuous in the state. Perturbed stability is therefore easily established. Disadvantages are that open- and closed-loop behavior are inherently different, regardless of N, and tuning can be counterintuitive.

Example. The following example highlights the limitations of these two infeasibility approaches. See Scokaert and Rawlings (1997) for further examples. We consider the third-order nonminimum phase system

$$A = \begin{bmatrix} 2 & -1.45 & 0.35 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad C = [-1 \quad 0 \quad 2], \tag{1}$$

for which the output displays inverse response. The controller tuning parameters are $Q = C'C$, $R = 1$, and $N = 20$. The input is unconstrained, the output is constrained between ± 1, and we perform simulations from the initial condition $x_0 = [1.5 \ 1.5 \ 1.5]'$. In all simulation results presented in the article, the solid lines represent the closed-loop results, the dashed lines represent the open-loop predictions made at time 0, and the dotted line shows the output upper constraint level.

The results obtained with the minimal time solution are presented in Figure 1. The open-loop predictions are close to the closed-loop results, so the two curves cannot be distinguished. As mentioned previously, the method leads to a large transient constraint violation. We show later that this violation is unnecessarily large. Constraint violations are not observed after sample 3, however, and the state constraints are enforced in minimal time.

The results obtained with the soft-constraint solution and a weight $S = 20$ are displayed in Figure 2. As discussed previously, the method leads to a large mismatch between open-loop predictions and closed-loop behavior, even with a horizon as large as $N = 20$. This mismatch does not decrease as N increases. With $S = 20$, peak violations are smaller than with the minimal time solution, but constraint violations are experienced in closed-loop until sample 13. Predicted open-loop performance is good, but degrades in the closed-loop using the receding-horizon implementation; the peak constraint violation is identical for both open- and closed-loop and the duration of violations is larger in closed-loop. The loss of performance experienced in closed-loop is caused by the open-/closed-loop mismatch and is the source of the tuning difficulties presented in the soft-constraint approach. To illustrate this point, we present in Figure 3 the results obtained with weights $S = 1$, 10, 50, and 100. For clarity of the plots, we show only closed-loop behavior. The effect of in-

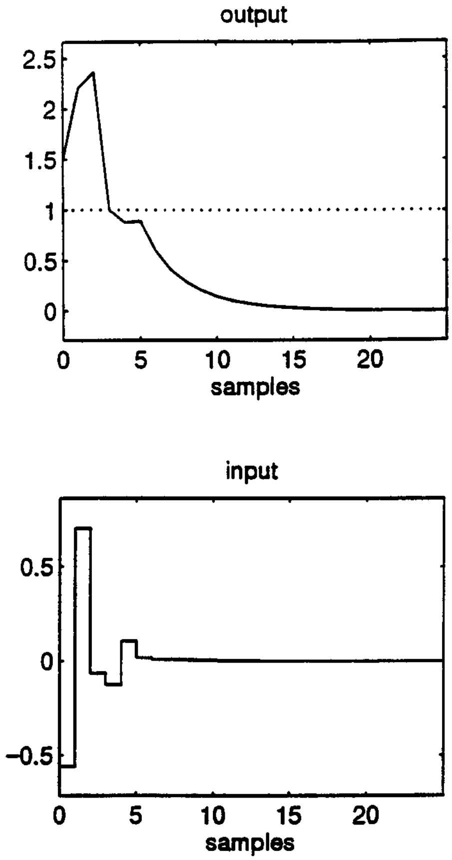

Figure 1. Minimal time solution.

Solid lines: closed-loop; dashed lines: open-loop predictions at time 0; dotted line: output upper constraint.

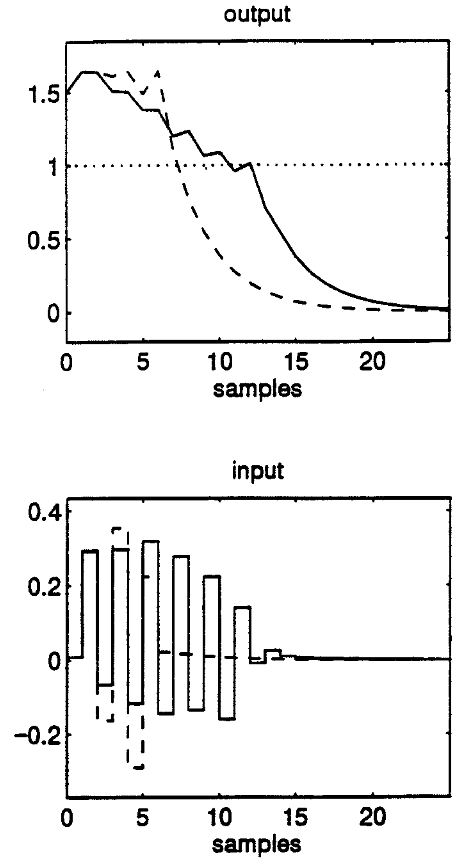

Figure 2. Soft constraint solution.

$S = 20$. Solid lines: closed-loop; dashed lines: open-loop predictions at time 0; dotted line: output upper constraint.

creasing S is expected to be a "hardening" of the state constraint. We find instead that although the peak constraint violation does decrease as S increases, the return to the feasible region becomes more sluggish, resulting in "large" violations for longer time. Because this behavior is not predicted in the open-loop simulation, the effect of closing the loop is unclear. For increasing values of S, performance further degrades; we show in Figure 4 the result of a simulation with $S = 1000$. In the limit, as $S \to \infty$, the duration of violations tends to infinity, which must be considered an undesirable feature of the control law.

The oscillations are due to the nonminimum phase behavior and the matrix A having imaginary eigenvalues. The violent oscillations are due to minimizing ϵ, the maximum constraint violation. At time $k = 0$, let ϵ_0 denote the maximum violation. If we look at the open-loop prediction, we see that the system decays to the origin in roughly 20 time steps. The system also violates the constraint for roughly 10 time steps with a constant violation of ϵ_0. At time $k = 1$, the controller is able to determine a maximum violation $\epsilon_1 < \epsilon_0$. Because of the large penalty, reducing this maximum violation is the prime objective of the controller. If we again look at the open-loop prediction, we see the system again decays to the origin in roughly 20 time steps (now 21 time steps in the closed loop). Also, the system violates the constraint for roughly 10 time steps with a constant violation of the ϵ_1 (now 11 time steps in the closed loop). The problem is that ϵ_1 is just barely less than ϵ_0, so the eventual decay to the origin is always being postponed. Likewise, the violation of the constraint is perpetually being extended.

In this simple example, the poor performance with some choices of S is easy to observe and understand; with more complex plants, however, the results obtained may appear counterintuitive. These issues traditionally have been viewed as tuning considerations, and the user has been required to select a weight S that yields acceptable performance. However, the interpretation of S is not clear. For small S, the constraints are not enforced with any vigor, even when they are feasible. For large S, large constraint violations can be experienced for long periods of time. The choice of an intermediate value for S appears to be the only appropriate one. To complicate matters further, S and N interact, so that a choice of S that gives good performance with a given horizon can become undesirable when N increases. Therefore reliable design guidelines for S are not available, causing a serious tuning difficulty, in view of the poor performance with some choices of S.

Having presented the difficulties associated with current approaches, the remainder of the article is organized as follows. In the following section we present two new approaches to the infeasibility problem. First we discuss the inherent trade-off that must be resolved by any controller when state

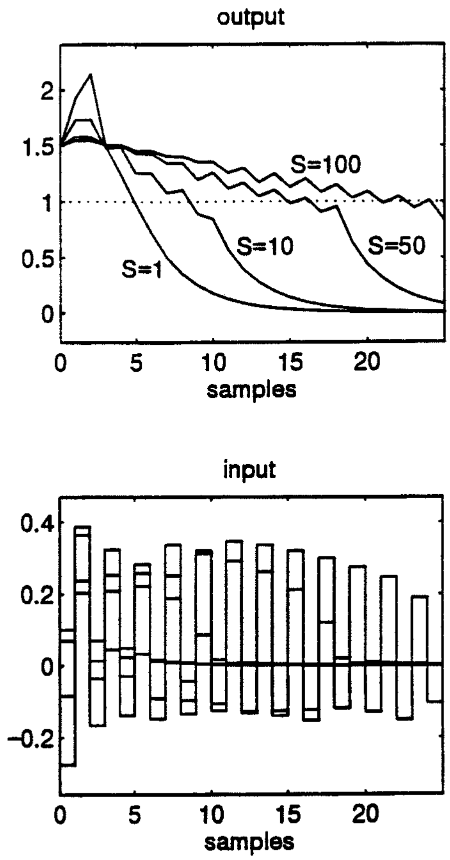

Figure 3. Soft constraint solution.

$S = 1$, 10, 50, and 100. Solid lines: closed-loop; dotted line: output upper constraint.

constraints are inconsistent and the control problem is infeasible. This multiobjective framework motivates the two new approaches: an optimal minimal-time approach and a new soft-constraint approach. We also discuss the use of exact soft constraints, which are penalty functions that enforce the hard constraints when the problem is feasible. Finally, concluding remarks are made in the third section. Simulations are used throughout the article to illustrate the main points.

New Results

Multiobjective nature of infeasibility problems

The minimal-time and soft-constraint approaches discussed previously are sharply contrasting solutions to the feasibility question: the first minimizes the duration of constraint violations, regardless of their size; the second minimizes the size of the violations, regardless of their duration. Each of these approaches may be appropriate for different processes. In general, however, both the duration and size of constraint violations are important, which leads to our interpretation of the feasibility question as a multiobjective problem.

When all the state constraints cannot be satisfied, it is normally desired to minimize the predicted violations in some way. Different measures of violation can be used; here, we consider the "size" of constraint violations as one measure and "duration" of the constraint violations as another. We note that alternative measures could be equally appropriate, but these two allow us to demonstrate the multiobjective nature of reducing state-constraint violations, which is our main purpose.

In many plants, the simultaneous minimization of the size and duration of the state-constraint violations is not a conflicting objective. The optimal way to handle infeasibility is then simply to minimize both size and duration; regulator performance may then be optimized, subject to the "optimally" relaxed state constraints.

Unfortunately, not all infeasibilities are as easily resolved. In some cases, such as nonminimum phase plants, a reduction in the size of violation can only be obtained at the cost of a large increase in duration of the violation, and vice versa. The optimization of constraint violations then becomes a multiobjective problem. For a given system and horizon N, the Pareto optimal-size/-duration curves can be plotted for different initial conditions, as in Figure 5. The user must then

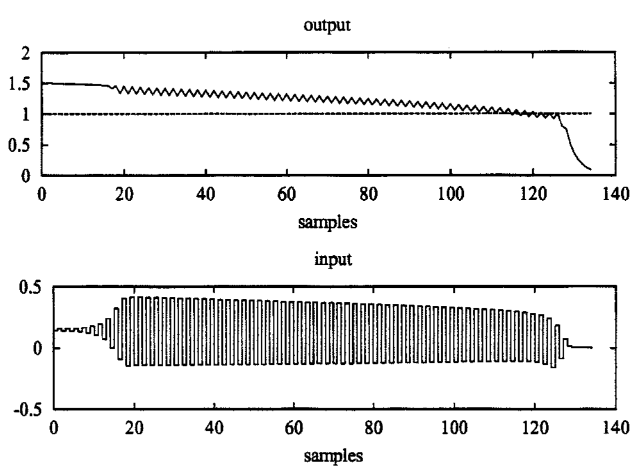

Figure 4. Soft constraint approach.

$S = 1{,}000$. Solid lines: closed-loop; dotted line: output upper constraint.

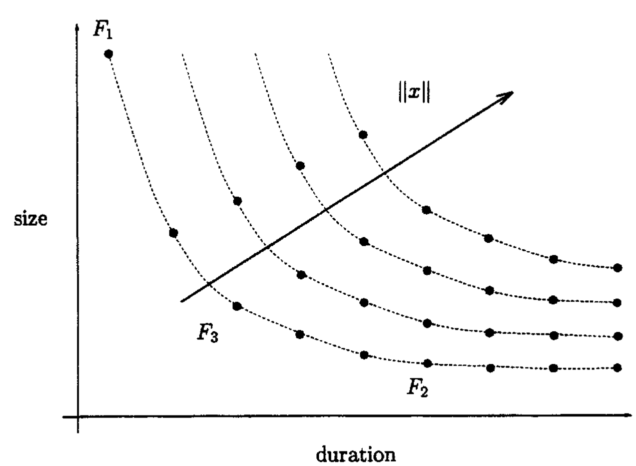

Figure 5. Pareto optimal constraint violation size/ duration curves for varying initial state *x*.

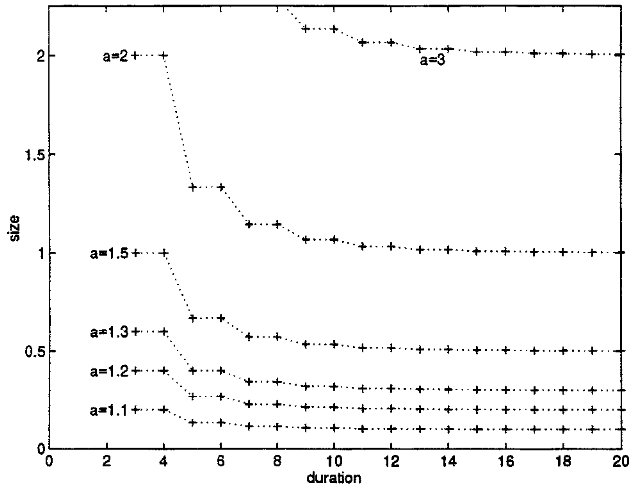

Figure 6. Pareto optimal size/duration curves for varying initial conditions: $x_0 = a[1\ 1\ 1]'$, with a = 1.1, 1.2, 1.3, 1.5, 2.3.

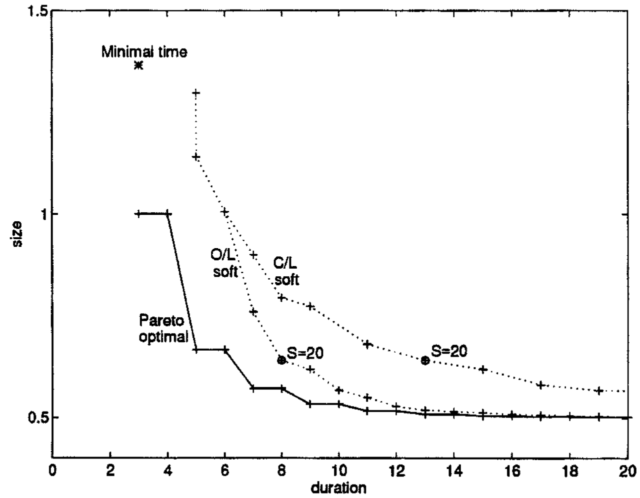

Figure 7. Comparison of Pareto optimal performance and performance under current solution methods.

decide where in the size/duration plane the plant should operate at times of infeasibility. Desired operation may lie on the Pareto optimal curve, because points below this curve cannot be attained and points above it are inferior, in the sense that they correspond to larger size and/or duration than are required.

In some applications, the best operation is to have constraint violations of the smallest possible duration. For instance, if the quality of the product is unacceptable and must be discarded during times of constraint violation, optimal operation is to minimize duration. Assume that the current state of the process corresponds to the bottom curve in Figure 5. In this situation, the controller should steer operation toward point F_1. In other applications, it may be critical to minimize the size of constraint violations, so that process shutdown or other exception conditions do not occur. Desirable control behavior is then close to F_2. It should be noted in Figure 5 that unconditional minimization of the size of the constraint violation is undesirable, because that leads to a nonzero violation of infinite duration. In many applications, both the size and duration of violations are important. Then F_3 may be the best compromise for the controller to aim for.

In Figure 6, we present the Pareto optimal-size/-duration curves for the plant in Eq. 1 for $N = 20$ and the initial states $x_0 = a[1\ 1\ 1]'$, with a = 1.1, 1.2, 1.3, 1.5, 2, and 3. The figure was constructed by bounding the size of violation, and then solving the optimal control problem to minimize the duration of the violation. For this plant, the size and duration measures are the peak constraint violation and the time after which the constraints are satisfied, respectively. As discussed previously, we find that the duration of constraint violations can only be decreased at the cost of larger violations, and vice versa. We also observe that, after a certain point, the decrease in size becomes negligible with increase in duration. Finally, we observe that, in this example and for the initial states considered, the size of constraint violations cannot be made arbitrarily small. The limitation on the minimal violation size is a function of N, and the use of larger horizons leads to Pareto optimal curves that are similar to the ones presented, but translated down slightly.

We now concentrate on the initial condition used in the previous simulations, $x_0 = [1.5\ 1.5\ 1.5]'$, and illustrate the drawbacks of both previously discussed infeasibility solutions. In Figure 7, the Pareto optimal-size/-duration curves are displayed as the solid line. The compromise obtained with the minimal-time solution is displayed as the star in the upper left corner of the figure. The minimal-time approach results in a constraint violation of 1.37, and we observe that the duration of constraint violations can be kept at 3, with a maximum violation of only 1.0005, which confirms our earlier claim that the minimal time approach is suboptimal. The results obtained with the soft-constraint approach are also displayed for a range of weights S. The upper dotted line represents the closed-loop results obtained with a range of weights S from 0.1 to 65. The lower dotted line represents the open-loop predictions made with weights ranging from 0.1 to 3000. The circled points on the curves represent the results with $S = 20$, as used for the simulation in Figure 2. The mismatch between open-loop predictions and closed-loop results is clear. The open-loop results are suboptimal for all values of S, though decreasingly so as S increases. Closed-loop performance is worse than open-loop, however, yielding points that are more inferior. Furthermore, closed-loop behavior only approaches Pareto optimal performance slowly as S increases. For all values of S, performance is inferior, especially in closed loop, in the sense that the size of constraint violations can be decreased with no adverse effect on duration. For large values, increases in S lead, in the closed loop, to decreasing improvements in size of violations at increasing duration cost. After a certain point, increases in S lead to increasingly suboptimal performance.

Having analyzed the difficulties that may arise with infeasibility approaches from the multiobjective optimization perspective, we propose two new approaches that deal more effectively with infeasibility. First we consider the minimal-time approach, which we refine to give Pareto optimal performance. We then discuss an alternative soft-constraint MPC law that does not suffer the disadvantages of the method described in Ricker et al. (1988) and Zheng and Morari (1995).

Optimized minimal-time approach

The minimal-time approach requires the calculation of the minimal horizon, $\kappa(x)$, beyond which the hard state constraint can be satisfied. This calculation normally requires some form of search. For instance, a simple procedure is to consider the time for which the "state-unconstrained" solution trajectories violate the state constraint, which gives an initial upper bound on $\kappa(x)$. Starting from this bound, κ is decreased until the last value for which the state constraints are feasible after horizon κ. Calculation of the state-unconstrained trajectories requires solution of a quadratic program. Each feasibility check requires the solution of a linear program.

The optimized minimal-time approach we consider arises from the consecutive solution of a feasibility and a regulation optimization. The essential improvement over the previous minimal-time approach is that the regulation problem must meet the smallest violation determined in the feasibility problem. Given $\kappa(x)$ and a vector $s > 0$ of constraint violation weights, the control law is obtained by receding-horizon implementation of the optimal open-loop control, given by the solution of the following problem.

Problem 1: Optimized Minimal-Time MPC

$$\min_{\pi_t} \phi(x_t, \pi_t)$$

$$\text{Subject to} \begin{cases} x_{j+1|t} = Ax_{j|t} + Bu_{j|t} & \\ Hx_{j|t} \le h + \epsilon_t^* & t \le j < t + \kappa(x_t), \\ Hx_{j|t} \le h & t + \kappa(x_t) < j, \\ Du_{j|t} \le d & t \le j, \\ u_{j|t} = 0 & t + N \le j, \end{cases}$$

where ϵ_t^* comes from solution of

$$\min_{\pi_t, \epsilon_t} s'\epsilon_t$$

$$\text{Subject to} \begin{cases} x_{j+1|t} = Ax_{j|t} + Bu_{j|t} & \\ Hx_{j|t} \le h + \epsilon_t & t < j < t + \kappa(x_t), \\ Hx_{j|t} \le h & t + \kappa(x_t) \le j, \\ Du_{j|t} \le d & t \le j, \\ u_{j|t} = 0 & t + N \le j, \\ \epsilon_t \ge 0. & \end{cases}$$

The regulation and feasibility optimizations are Nm and $(Nm + p)$-dimensional, respectively.

We return to the plant in Eq. 1 and illustrate the use of the optimized minimal-time approach. The results are presented in Figure 8. We observe that the state constraint is enforced in minimal time, as with the original minimal-time solution. However, the peak transient constraint violation is also reduced to the lowest value consistent with the minimal-time solution. The peak violation is 1.0005 and the duration of violations is 3. Operation therefore occurs on the Pareto optimal curve of Figure 7, at its leftmost point. This behavior is a property of the method.

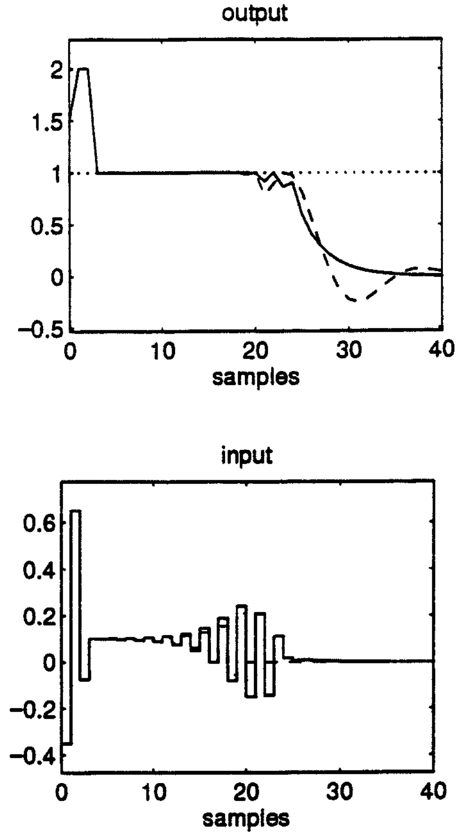

Figure 8. Optimized minimal time solution.

A side effect of the minimization of the transient constraint violations is that the output rides its constraint for several samples, before returning inside the feasible region. It is instructive to note that the time at which the output moves off the constraint is 20, the value of N. As discussed previously the minimal violation size, 1.0005, is a function of this horizon; further reductions of the violation size can be obtained by increasing N, and this induces the output to ride its constraint for a longer period. The mismatch between the open-loop predictions for horizons greater than N and the actual closed-loop behavior at these times is inherent to the method and does not disappear with large N. However, this mismatch does not lead to performance degradation in closed-loop, as observed in Figure 8. Also, it is simple to modify Problem 1 so that the inherent open-/closed-loop mismatch is removed. These points are discussed later in this article.

Performance. One property of interest of minimal-time approaches is that, when the state constraints are feasible, the algorithm enforces them, allowing no violation $[\kappa(x) = 0]$. Thus, when feasibility is not a problem, the control algorithm behaves exactly as the hard-constraint MPC formulation. Thus, minimal-time approaches resolve the problem of infeasibility with no adverse effect on performance at times of feasibility, which is an important feature of the method because the occurrence of infeasibility is regarded as an unusual occurrence.

An interesting property specific to the optimized minimal time approach is that performance is Pareto optimal, with a minimal constraint violation size that is consistent with the fastest return of the state to its allowed region and the horizon N. The method gives satisfaction of the constraints in minimal time, with a guarantee in the nominal case of constraint satisfaction after $\kappa(x_0)$ samples.

Open-/Closed-Loop Comparison. The minimization of the constraint violation ϵ_t can induce mismatch between the open-loop predictions and the closed-loop behavior. This problem is caused by the fact that, at the initial time of infeasibility, the controller uses the N available controls to minimize the size of constraint violations. As the simulation proceeds, the state is returned to its allowed region and more inputs become available to further optimize regulation performance. In the nominal case, $\kappa(x)$ decreases at every sample, until it reaches zero. Therefore, the control reverts to the hard constraint law after $\kappa(x_0)$ samples; the mismatch between open- and closed-loop then effectively disappears for large N.

Stability. Minimal-time approaches have guaranteed nominal exponential stability. The stability proof involves establishing the optimal regulator cost as a Lyapunov function for the closed-loop system and noting that κ decreases by one at each time.

Although simulations with simple examples indicate that the optimized minimal-time control is Lipschitz continuous in the state, this property is difficult to prove. Because $\kappa(x)$ is integer valued, $\kappa(x)$ is not Lipschitz continuous in x, and it is difficult to establish continuity of ϵ_t^*, and consequently of the optimal control. Because discontinuity of the control invalidates known perturbed stability results, perturbed stability for minimal-time solutions remains an open question. This shortcoming is one of the more serious weaknesses of the method.

Soft constraints

In the soft-constraint approach, only the input constraints are directly enforced. Violation of the state constraints are allowed, but an additional term is introduced into the objective function that penalizes a measure of the constraint violations; it has been usual to choose the peak predicted violation over the horizon as this measure (Ricker et al., 1988; Zheng and Morari, 1995). As discussed previously, this choice introduces mismatch between open-loop predictions and closed-loop behavior, which can lead to poor performance and tuning difficulties. As an alternative that does not suffer these problems, we consider in this section an alternative formulation of soft-constraint MPC, where rather than penalizing the peak violation, we penalize total sum of the violation at each time step along the prediction horizon.

Here we concentrate on the use of combined quadratic- and linear-term soft constraints. As we discuss later, by penalizing the sum of constraint violations at each time step, this MPC formulation satisfies the principle of optimality and removes the counterintuitive behavior seen in Figures 3 and 4. With $\epsilon_{j|t}$ denoting the predicted constraint violations, we consider the following objective function

$$\phi(x_t, \pi_t, \sigma_t) = \sum_{j=t}^{\infty} x'_{j|t} Q x_{j|t} + u'_{j|t} R u_{j|t} + \epsilon'_{j|t} S \epsilon_{j|t} + s' \epsilon_{j|t},$$

where

$$\sigma_t = \{\epsilon_{t+1|t}, \epsilon_{t+2|t}, \dots\}, \qquad (2)$$

S is a diagonal matrix of positive weights, and s is a vector with positive entries.

The reason we penalize the weighted l_1 norm of $\epsilon_{j|t}$ with the term $s'\epsilon_{j|t}$ is to allow the use of exact penalties. The inclusion of the quadratic penalty is to allow for added flexibility. The quadratic penalty also leads to a well-posed quadratic program (positive definite Hessian). We can also achieve Pareto optimality with a linear penalty due to the exactness of the soft constraint. One disadvantage with very large linear penalties is that it is tough to gauge the competing effect with other penalties. Also, a poorly scaled objective function can cause numerical difficulties. Therefore, the objective effectively penalizes the sum of both linear and quadratic measures of constraint violations; tuning S and s allows us to adjust the relative importance of the two terms in the cost.

The soft-constraint MPC law arises by receding-horizon implementation of the following optimal control problem.

Problem 2: Soft-Constraint MPC

$$\min_{\pi_t, \sigma_t} \phi(x_t, \pi_t, \sigma_t)$$

$$\text{Subject to} \begin{cases} x_{j+1|t} = Ax_{j|t} + Bu_{j|t} & \\ Hx_{j|t} \le h + \epsilon_{j|t} & t < j, \\ \epsilon_{j|t} \ge 0 & t < j, \\ Du_{j|t} \le d & t \le j, \\ u_{j|t} = 0 & t + N \le j. \end{cases}$$

Although this optimization has infinite dimension, it can be shown that the restriction that $\epsilon_{j|t} = 0$, after a suitably large finite horizon, is not suboptimal and therefore leads to the solution to Problem 2. See Meadows et al. (1995) for similar results.

As in the soft-constraint scheme discussed in the Introduction, increases in S lead to ''hardening'' of the state constraints. The measure we use leads, however, to more intuitive results, as we now illustrate. We consider the plant in Eq. 1 again and illustrate the use of the soft-constraint approach that arises by receding-horizon implementation of the solution to Problem 2. For simplicity, we first consider simulations in which $s = 0$, so that the sum of only the quadratic measure of violations is penalized. The use of $S = 20$ leads to the results presented in Figure 9. Both open and closed-loop results are displayed, though they are too close to be distinguished.

For comparison with the previous soft-constraint approach presented in the Introduction we present in Figure 10 closed-loop results with a range of weights, $S = 1$, 10, 50, and 100. We observe that the tuning difficulties experienced with

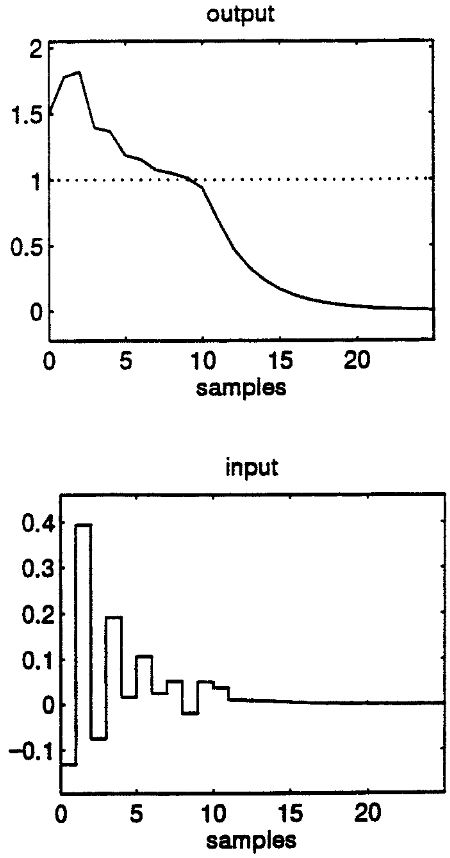

Figure 9. Least-squares soft constraint solution.

$S = 20$. Solid lines: closed-loop; dashed lines: open-loop predictions at time 0; dotted line: output upper constraint.

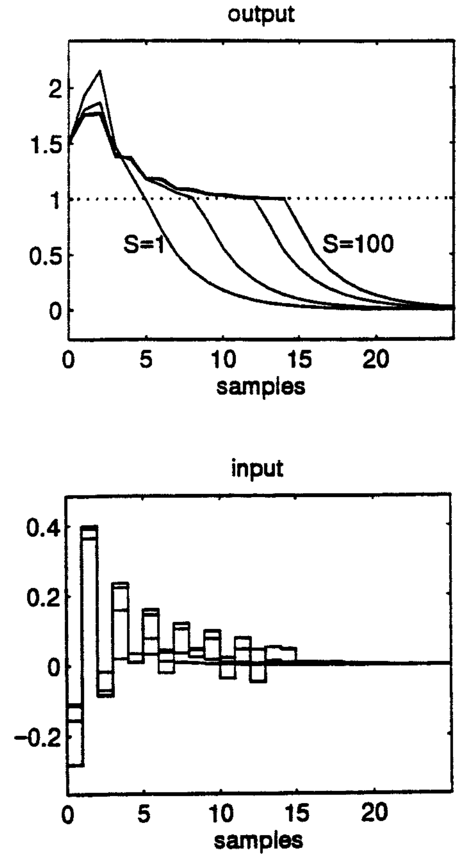

Figure 10. Least-squares soft constraint solution.

$S = 1$, 10, 50, and 100. Solid lines: closed-loop; dashed lines: open-loop prediction at time 0; dotted line: output upper constraint.

the previous soft-constraint law do not arise with this choice of violation measure. With the new approach, increases in S result in reductions in the sum of the squares of violations and have no adverse effect on the constraint violations experienced in closed loop. Tuning is therefore intuitive and the sum of the squares of constraint violations approaches optimality for large S. In the limit, as S increases, the behavior of the control has a well-behaved limit; with $S = 1000$, for instance, we obtain the results shown in Figure 11, which compare favorably with Figure 4.

We note that performance is not Pareto optimal in the sense of the subsection on the multiobjective nature of infeasibility problems, because the control law does not attempt to minimize the peak constraint violation. As mentioned previously, however, the use of the peak constraint violation as the measure of size of violations is somewhat arbitrary. An equally valid measure is the one we use in the soft-constraint cost, the sum of the weighted squares of $\epsilon_{j|t}$. With this measure, we obtain a new set of Pareto optimal curves, and the soft-constraint control law leads to a performance that is increasingly close to Pareto optimal, for large S.

To conclude, we present simulations with nonzero s. In Figure 12, we display the results obtained with $s = 0.4$, 4, 40, 400, 4,000, and 40,000. For clarity of the plots, only the closed-loop behavior is shown. No noticeable performance changes are observed for s greater than 40,000. We find that increasing s results in increasing peak violations and decreasing duration. Large values of s also cause the output to ride its constraint for a large period of time. For a given horizon N, the performance obtained has a well-defined limit as s increases.

Exact Soft Constraints: Choice of S and s. The use of soft constraints provides an attractive solution to the problem of infeasibility, in which the control optimization is guaranteed to have a solution and remains unchanged, regardless of the feasibility of the hard state constraints. A resulting disadvantage of the method is that, even when the state constraints can be satisfied, the soft-constraint law may not enforce the state constraints and unnecessary violations may result. This feature of the control algorithm is undesirable and does not arise, for instance, with the minimal-time approach.

A desirable solution to infeasibility enforces the state constraints when they are feasible and only relaxes them when necessary. This type of scheme can be obtained by using the solution of the soft-constraint algorithm only when the hard-constraint MPC law leads to an infeasible optimization. The use of exact penalties, leading to what we call exact soft constraints, provides an alternative solution. Fletcher (1987) provides a summary of exact penalty methods in optimization problems. De Oliveira and Biegler discuss using l_1, quadratic, and l_∞ norm-stage cost measures of state-constraint penalties in finite-horizon MPC (de Oliveira and Biegler, 1994). Niko-

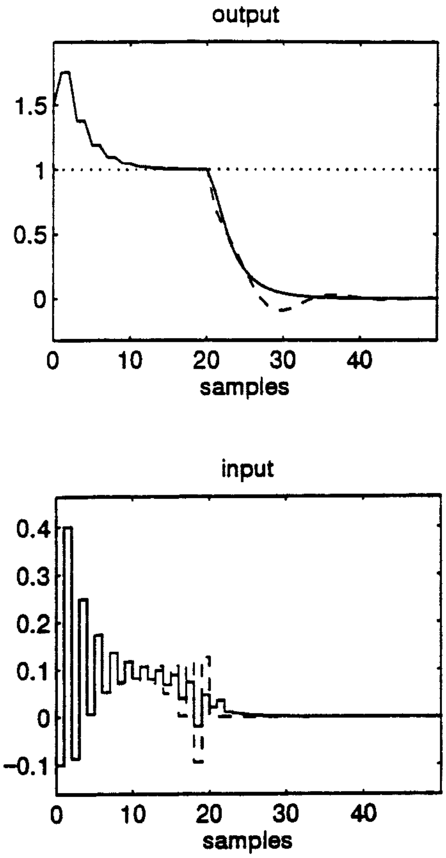

Figure 11. Least-squares soft constraint solution.

$S = 1000$. Solid lines: closed-loop; dashed lines: open-loop predictions at time 0; dotted line: output upper constraint.

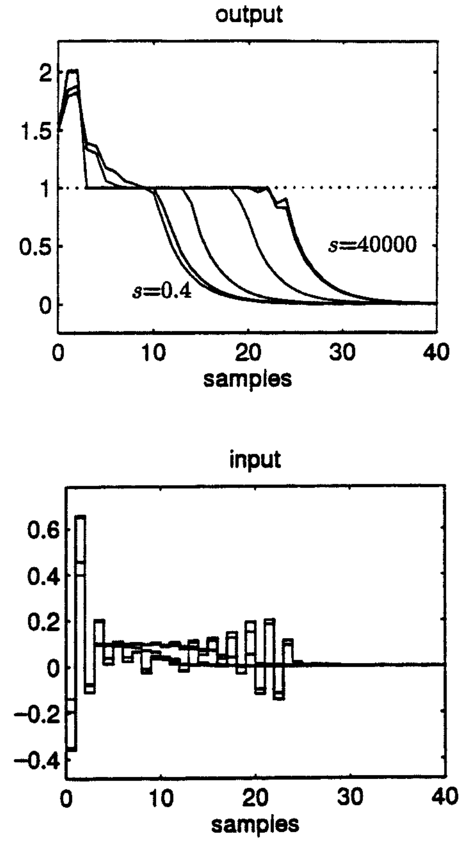

Figure 12. Least-squares soft constraint solution.

$S = 20$, $s = 0.4$, 4, 40, 400, 4000, and 40,000. Solid lines: closed-loop; dotted line: output upper constraint.

laou and coworkers discuss soft state constraints using l_1 and quadratic-stage cost penalties in DMC with a terminal stability constraint (Genceli and Nikolaou, 1993; Vuthandam et al., 1995).

Because the soft-constraint MPC objective we use effectively penalizes the 1-norm of constraint violations when $s > 0$, it can be viewed as an l_1 penalty function for the hard-constraint problem. The essential property of l_1 penalty functions, which allows them to be made exact, is that they have a discontinuity in slope at the zero value of the slack variables. The quadratic penalty functions, on the other hand, do not have this property, which is why we introduce the s term in the quadratic penalty used in the objective function. The implementation of exact soft constraints requires the calculation of a weight s for which the objective defines an exact penalty function. Although it is not difficult to compute a lower bound for s, in practice, one typically just makes the penalty large.

We illustrate the method with the following example. We use the plant of Eq. 1, but the initial state we consider is $x_0 = [0.95\ \ 0.95\ \ 0.95]'$, for which the output constraints can be enforced at all times, with $N = 20$. At time 0, the vector of Lagrange multipliers at the solution of the problem with hard constraints has infinity norm 11.07. The results of a simulation performed with the combined linear/quadratic soft-constraint MPC law described previously with $S = 20$ and $s = 11.08$ are presented in Figure 13, where the open-loop and closed-loop results are too close to be distinguished. As expected, no constraint violations are observed, and simulations show that violations are experienced with $s < 11.07$, confirming the results discussed previously. Incidentally, for any s greater than the infinity norm of the vector of Lagrange multipliers, the constraints are satisfied exactly if they are feasible; therefore, for this initial state, no change in performance is obtained by increasing s further.

Performance. As mentioned previously, when we cannot satisfy all of the state constraints, performance is increasingly Pareto optimal for large S, if Pareto optimality is defined in terms of the same measure of size of violations as we use in the soft-constraint objective. Tuning requires the adjustment of two parameters, S and s. However, the effects of changing S and s are intuitive and, more importantly, nominal performance remains satisfactory for all designs.

Furthermore, if the use of exact soft constraints is not critical, so that small violations are allowed at times when the hard constraints could be enforced, s may be set to zero, simplifying the tuning procedure. When, on the other hand, it is desired to enforce exact soft constraints, s can be made large to achieve this result easily.

In general, s should either be set to zero or to the least value that gives exact soft constraints. The entries of the diagonal weighting matrix, S, can be made as large as possible,

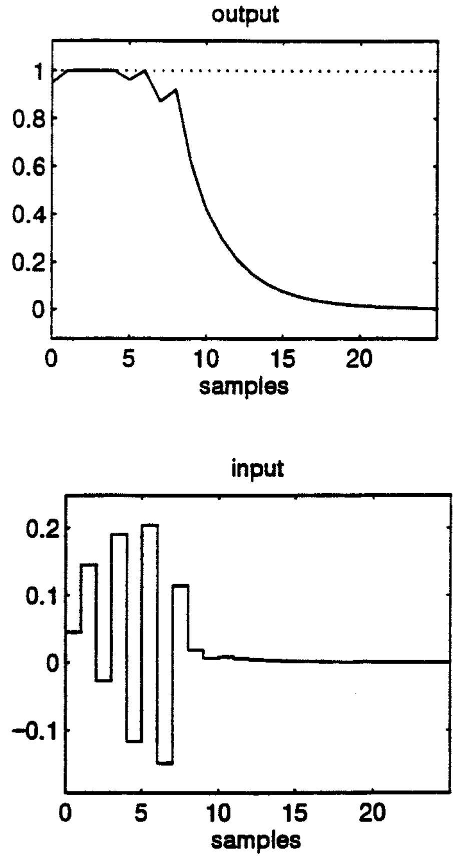

Figure 13. Exact soft constraints.

with due regard to numerical difficulties that can arise in the solution of Problem 2. When more than one output is constrained, the relative values of the diagonal weights in S should be adjusted to place emphasis on the more important constraints.

Open-/Closed-Loop Comparison. A desirable property of the soft-constraint control law described in this section is that, for a fixed set of parameters Q, R, S, and s, the mismatch between open-loop predictions and nominal closed-loop behavior approaches zero as N increases. This infeasibility solution is therefore the only one discussed in this article for which a principle of optimality can be stated. Control laws that penalize the peak constraint violation in the horizon are not defined in terms of the multistage open-loop optimization, and therefore do not admit a principle of optimality. This is the fundamental reason for the open-/closed-loop mismatch and tuning difficulties illustrated in the introduction.

In receding-horizon control, it is relatively easy to define a control optimization that gives good open-loop performance. The difficulties arise when the properties of the closed-loop are considered. The principle of optimality implies that for large N, performance is similar in open and closed loop. Hence it leads to more transparent control laws and to an easier understanding of nominal closed-loop behavior.

Stability. Soft-constraint MPC formulations are nominally exponentially stabilizing and asymptotically stabilizing under decaying perturbations.

Conclusions

In this article, we have discussed the feasibility problems that can arise due to the use of hard state constraints in MPC. Current solution techniques were reviewed and their limitations were highlighted. Two alternative control laws that resolve infeasibility were described: the optimized minimal-time solution and a new soft-constraint approach. Both methods were shown to lead to exponentially stabilizing control, and perturbed stability was established for soft-constraint MPC.

Central to our discussion was the concept of Pareto optimality of the constraint violations that are experienced at times of infeasibility and the mismatch that can arise between open-loop predictions and closed-loop behavior.

With the previous soft-constraint approaches, open-loop behavior was found to be good for a wide range of tuning choices and close to Pareto optimal for large S; however, closed-loop behavior was shown to be poor in some cases. The weakness of the previous soft-constraint approaches therefore is the inability to track open-loop predictions, nominally, in closed-loop. The minimal-time approach discussed earlier was shown to satisfy constraints as soon as possible, but the transient violations were larger than necessary.

Ideally, performance should be Pareto optimal. With the optimized minimal-time approach discussed in the subsection titled "Optimized Minimal-Time Approach," the control law forces operation to the Pareto optimal point that corresponds to the minimal duration of violations. Although operation at this point is desirable in some applications, the method is not always appropriate because it cannot be tuned to give performance at other optimal points.

Except for the optimized minimal-time approach, however, it appears difficult, in general, to force operation at Pareto optimality by receding-horizon implementation of the solution of open-loop optimizations that are no more complex than quadratic programs.

The use of least-squares soft constraints nevertheless leads to a control scheme for which open- and closed-loop become increasingly similar for large N. For large S, the control law also leads to performance that is close to Pareto optimal, when the measure of the size of violations considered is the sum of squares. Moreover, tuning of the soft constraints allows for a variety of duration/size trade-offs, and closed-loop performance remains good for all designs. The price for these desirable features is that the control calculation requires the solution of a larger quadratic program than previous soft-constraint formulations. However, by properly structuring the quadratic program, the addition of the soft constraints does not increase the complexity of the quadratic program (Rao et al., 1998).

Acknowledgments

The financial support of the National Science Foundation, through Grant CTS-9311420, and the industrial members of the Texas-Wisconsin Modeling and Control Consortium, is gratefully acknowledged. We acknowledge K. R. Muske and C. V. Rao for helpful discussions and feedback about this article.

Literature Cited

de Oliveira, N. M. C., and L. T. Biegler, "Constraint Handling and Stability Properties of Model-Predictive Control," *AIChE J.*, **40**, 1138 (1994).

Fletcher, R., *Practical Methods of Optimization*, Wiley, New York (1987).
García, C. E., and A. M. Morshedi, ''Quadratic Programming Solution of Dynamic Matrix Control (QDMC),'' *Chem. Eng. Commun.*, **46**, 73 (1986).
Genceli, H., and M. Nikolaou, ''Robust Stability Analysis of Constrained l_1-Norm Model Predictive Control,'' *AIChE J.*, **39**, 1954 (1993).
Meadows, E. S., K. R. Muske, and J. B. Rawlings, ''Implementable Model Predictive Control in the State Space,'' *Proc. 1995 Amer. Control Conf.*, IEEE, p. 3699 (1995).
Rao, C. V., S. J. Wright, and J. B. Rawlings, ''On the Application of Interior Point Methods to Model Predictive Control,'' *J. Optimiz. Theory Appl.*, **99**, 723 (1998).
Rawlings, J. B., and K. R. Muske, ''Stability of Constrained Receding Horizon Control,'' *IEEE Trans. Automat. Contr.*, **AC-38**, 1512 (1993).
Ricker, N. L., T. Subrahmanian, and T. Sim, ''Case Studies of Model-Predictive Control in Pulp and Paper Production,'' *Proc. 1988 IFAC Workshop on Model Based Process Control*, T. J. McAvoy, Y. Arkun, and E. Zafiriou, eds., Pergamon Press, Oxford, p. 13 (1988).
Scokaert, P. O., and J. B. Rawlings, ''On Infeasibilities in Model Predictive Control,'' *Chemical Process Control—V*, J. C. Kantor, C. E. García, and B. Carnahan, eds., CACHE, AIChE, New York, p. 331 (1997).
Scokaert, P. O., J. B. Rawlings, and E. S. Meadows, ''Discrete-Time Stability with Perturbations: Application to Model Predictive Control,'' *Automatica*, **33**, 463 (1997).
Vuthandam, P., H. Genceli, and M. Nikolaou, ''Performance Bounds for Robust Quadratic Dynamic Matrix Control with End Condition,'' *AIChE J.*, **41**, 2083 (1995).
Zafiriou, E., and A. L. Marchal, ''Stability of SISO Quadratic Dynamic Matrix Control with Hard Output Constraints,'' *AIChE J.*, **37**, 1550 (1991).
Zheng, A., and M. Morari, ''Stability of Model Predictive Control with Mixed Constraints,'' *IEEE Trans. Automat. Contr.*, **AC-40**, 1818 (1995).

Manuscript received Aug. 28, 1998, and revision received May 12, 1999.