Pergamon PII: S0005-1098(96)00255-5 *Automatica*, Vol. 33, No. 5, pp. 763–781, 1997

Printed in Great Britain
0005-1098/97 $17.00 + 0.00

Worst-case Formulations of Model Predictive Control for Systems with Bounded Parameters*

J. H. LEE† and ZHENGHONG YU

Different min-max formulations of MPC for state-space systems with bounded parameters are examined from a closed-loop robustness viewpoint, and computationally attractive algorithms are proposed.

Key Words—Dynamic programming; min-max technique; predictive control; robust control.

Abstract—Two different predictive control formulations are developed based on minimization of the worst-case quadratic cost for systems with bounded parameters. The two formulations differ on the assumptions made about the future inputs in optimizing the current input: one assumes open-loop control, while the other considers closed-loop control. Their closed-loop properties such as asymptotic stability are examined. We then focus on a moving average model with an integrator, and derive computationally simpler suboptimal algorithms. © 1997 Elsevier Science Ltd.

1. INTRODUCTION

Model predictive control (MPC), also known as receding horizon control (RHC) and generalized predictive control (GPC), has recently received much attention from both theoreticians and practitioners in diverse fields. Although different fields have adopted different names and favorite model forms, the underlying concept is the same: Solve a finite or infinite horizon open-loop optimal control problem at every sample time and set the control input according to the optimal profile until the next sample time. Updating states using measurements before each optimization results in feedback control. Motivations for using this technique also differ from field to field. For example, in the process industries, the main attractive feature of MPC is its ability to handle time-domain constraints and model nonlinearities in multivariable control problems (Garcia *et al.*, 1989).

The main problem with the technique has been the lack of stability and robustness guarantees. The stability problem has been solved to a satisfactory degree in several recent papers (Keerthi and Gilbert, 1988; Mayne and Michalska, 1990; Rawlings and Muske, 1993). The robustness of model predictive controllers has also been analyzed using the structured singular value theory, to derive tuning rules for observer and regulator parameters (Lee and Yu, 1994). However, this type of analysis is limited mostly to unconstrained, linear systems.

One promising way to utilize parametric uncertainty information within the MPC framework is to base the optimization on a whole set of future predictions defined by possible parameter values. This notion has been explored by several researchers in various contexts. For example, one can model the uncertain system parameters as stochastic variables to formulate a predictive control law that minimizes the expected cost for some future horizon (Lee and Cooley, 1995). Another possibility is to model the uncertainty using deterministic bounds in the parameter space and minimize the 'worst-case' cost. The latter approach has been referred to as 'min-max MPC' in the literature, and is the focus of this paper.

One of the early works on min-max MPC was by Campo and Morari (1987), who applied the idea of worst-case error minimization to finite impulse response (FIR) models with ∞-norm bounded affine perturbations. For this particular model type, they showed that the problem of finding an input sequence minimizing the worst-case tracking error can be solved via linear programming (LP). More recently, Lau *et al.* (1991) showed that quadratic minimization of the worst-case error for an ellipsoidal set of FIR coefficients is convex. Their motivation for using ellipsoidal parameter bounds was that such

* Received 26 September 1994; revised 26 July 1995; received in final form 5 October 1996. This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor J. C. Kantor under the direction of Editor Yaman Arkun.
Corresponding author Professor Jay H. Lee. Tel. +1 334 844 2060; Fax +1 334 844 2063; E-mail jhl@eng.auburn.edu.
† Department of Chemical Engineering, Auburn University, Auburn, AL 36849-5127, U.S.A.

bounds arise naturally when noise and parameters are assumed to be Gaussian variables during identification (Goodwin *et al.*, 1990; Kosut *et al.*, 1990). More recently, Veres and Norton (1993) presented an adaptive method in which parameter bounds are updated recursively and open-loop worst-case error is minimized at each time step. They also proposed a way to relax the tracking error minimization requirement and to inject test input signals in order to tighten the parameter bounds. Although simplistic examples presented in these papers indicate the potentials of the min-max approach, none of the algorithms have been tested extensively, especially as means of feedback control.

A noteworthy point about the min-max MPC techniques that have appeared in the literature thus far is that, following the convention of regular MPC techniques, they all assume *open-loop* control in the optimization. In other words, the formulations do not account for the fact that only the first element of the computed input sequence is implemented and the future inputs are determined from new optimizations performed after feedback updates. This inconsistency is insignificant for the single-model-based formulation (either because the future is entirely deterministic or because separation holds), but it can lead to poor performance of min-max MPC. Realization of this has led to efforts to modify the min-max formulation such that it is robust in the *closed-loop* sense. For example, Zheng and Morari (1993) have presented an ∞-norm based min-max formulation for a special class of linear, time-varying systems and have proved robust stability under certain choices of tuning parameters.

The focus of the present paper is the development of min-max MPC formulations from the viewpoint of closed-loop control. First, we present open-loop and closed-loop formulations of min-max MPC for general state-space systems with time-invariant or time-varying parametric uncertainties. We then examine the closed-loop properties of these formulations. We demonstrate that min-max MPC based on the open-loop control assumption can give poor closed-loop performance, especially when uncertainties are assumed to be time-invariant in the formulation. This is true even when the underlying system is time-invariant. When the uncertainty is allowed to vary from one time step to next in the prediction, the open-loop formulation gives robust, but cautious, control. The closed-loop formulation, presented only for the time-varying case, duly considers the recurrent nature of the optimization. The resulting problem, however, is a dynamic program, which must be solved numerically. We examine the robust stability of different formulations under infinite horizon.

We then narrow the scope to FIR systems with integrating-type disturbances. As mentioned, the closed-loop formulation leads to a dynamic program, which must be solved numerically. Numerical solution requires discretization of states, and therefore suffers from the 'curse of dimensionality'. The technique is impractical for all but small-size problems. Hence, we focus on the open-loop formulation (with time-varying uncertianty) and show that the resulting optimization is convex. Next, we restrict the parameter set to ellipsoids and make further approximations to interpret the 'worst-case' objective as a squared sum of nominal error plus an input penalty term. The computational requirement of this suboptimal algorithm is comparable to that of the conventional MPC. We present several examples to compare the performance of the suboptimal algorithms with numerical solutions of the dynamic programs.

2. MIN-MAX FORMULATIONS OF PREDICTIVE CONTROL

2.1. *Problem definition*

Throughout this section, we will base our discussion on the following state–space system:

$$z_{k+1} = A(\vartheta_k)z_k + B(\vartheta_k)v_k, \tag{1}$$

where z is the state vector and v is the manipulated input vector. ϑ is an *uncertain* vector that parametrizes the system matrices and can be constant or time-varying. For simplicity, we assume that z is measured perfectly at each time. Bounded disturbance and noise signals are not considered in this paper for the sake of simplicity, but can be easily incorporated into the results we present subsequently.

We are interested in minimizing the cost

$$\sum_{k=0}^{\infty} z_k^{\mathrm{T}}Qz_k + v_k^{\mathrm{T}}Rv_k \tag{2}$$

by using feedback control of the form $v_k = f(z_k)$, where $f(\cdot)$ may be defined either explicitly or implicitly through an optimization problem. In the above, it is assumed that $Q \geq 0$ and $R > 0$. The following constraints are imposed in achieving the above objective:

$$v_k \in \mathscr{V}, \tag{3}$$

$$z_k \in \mathscr{X}, \tag{4}$$

where $\mathscr{V}$ and $\mathscr{X}$ are compact, convex sets that include the origin. The input constraints represent physical limits of actuators, while the state constraints can arise from performance and safety considerations.

2.2. *Characterization of parameter vector*

The parameter vector sequence $\{\vartheta_k\}$ is unknown, but is assumed to belong to some compact set. We consider both the time-invariant and time-varying cases.

Time-invariant case. In the time-invariant case, $\vartheta_k = \vartheta$, $\forall k$ and

$$\vartheta \in \Theta, \tag{5}$$

where Θ is a compact set. The choice of parameter sets studied in the literature has been driven mainly by how convenient they are for developing efficient identification and control algorithms. The following two choices have been most popular.

(i) *Ellipsoidal set*: The compact set can be chosen to be an ellipsoid, which may be expressed as

$$\Theta \triangleq \{\theta : \|W(\theta - \tilde{\vartheta})\|_2 \leq 1, \theta \in \mathscr{R}^n\} \tag{6}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. The nominal parameter vector $\tilde{\vartheta}$ represents the center of the ellipsoid. $W \in \mathscr{R}^{m \times n}$ specifies the orientation and size of the ellipsoid. Because parameter estimation under the Gaussian noise assumption naturally yields ellipsoidal bounds, this uncertainty description has been used by many researchers (Goodwin *et al.*, 1990; Kosut *et al.*, 1990; Lau *et al.*, 1991).

(ii) *Axis-aligned polyhedron*: In this case, the Euclidean norm is replaced by the ∞-norm. The axis-aligned polyhedron description has also been adopted by many researchers, including Campo and Morari (1987), Zheng and Morari (1993), and Genceli and Nikolaou (1993).

Figure 1 shows both types of uncertainty description for a two-dimensional case. Methods for obtaining and updating such bounds from input–output data will not be discussed here, but can be found in the literature (Milanese and Belforte, 1982; Fogel and Huang, 1982; Belforte *et al.*, 1990).

Time-varying case. In the time-varying case, we may assume

$$\vartheta_k \in \Theta \quad \forall k, \tag{7}$$

or, more generally,

$$\vartheta_k \in \Theta_k. \tag{8}$$

In the above, each element of $\{\vartheta_k\}$ is assumed to be independent of the other elements, and, therefore, it can be a conservative description. More generally, one can use

$$\vartheta_{k+1} = \Phi\vartheta_k + \eta_k, \tag{9}$$

where η_k is a sequence the elements of which belong to a compact set. We can express the above as $\vartheta_k \in \Theta_k$ where Θ_k evolves from Θ_0 according to (9). However, this representation is conservative, as elements of the sequence ϑ_k are not completely independent, i.e. there are sequences that satisfy $\vartheta_k \in \Theta_k$ but are not possible parameter sequences.

2.3. *Open-loop formulation*

Conventional MPC (which requires ϑ_k to be completely deterministic) is formulated as the computation of an open-loop input trajectory minimizing the quadratic cost defined for a fixed-size, moving horizon at each sample time. At $t = k$, MPC solves

$$\min_{\mathscr{U}_k} \left\{ x_{k+p}^{\mathrm{T}} Q_p x_{k+p} + \sum_{l=1}^{p-1} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j} \right\}, \tag{10}$$

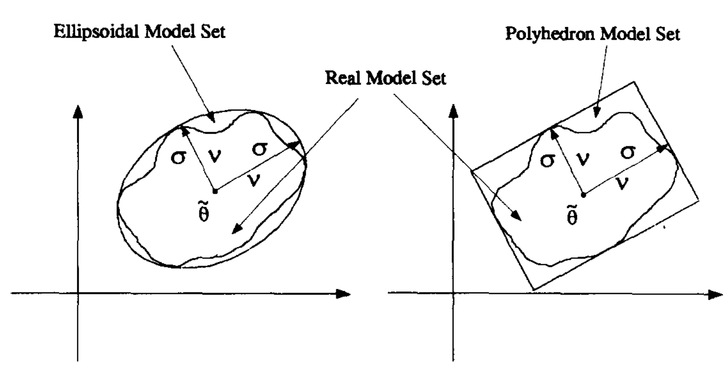


Fig. 1. Geometric interpretation of ellipsoidal and polyhedron model sets for the two-dimensional case.

with the constraints

$$x_{k+l} \in \mathscr{X}, \quad l = 1, \ldots, p, \tag{11}$$

$$u_{k+j} \in \mathscr{V}, \quad j = 0, \ldots, q-1, \tag{12}$$

$$u_{k+i} = 0, \quad i = q, \ldots, p-1, \tag{13}$$

and

$$x_{k+l} = \left(\prod_{i=0}^{l-1} A(\vartheta_{k+i})\right) z_k + \sum_{j=0}^{l-1} \left(\prod_{i=j+1}^{l-1} A(\vartheta_{k+i})\right) B(\vartheta_{k+j}) u_{k+j}, \tag{14}$$

$$\mathscr{U}_k = [u_k^{\mathrm{T}}, u_{k+1}^{\mathrm{T}}, \ldots, u_{k+q-1}^{\mathrm{T}}]^{\mathrm{T}} \tag{15}$$

In the above, $q \le p$; hence, the control-input horizon can be set smaller than the prediction horizon. The actual control input is determined by

$$v_k = u_k^* \triangleq [I \quad 0 \quad \ldots \quad 0] \times \arg\left\{\min_{\mathscr{U}_k} \left(x_{k+p}^{\mathrm{T}} Q_p x_{k+p} + \sum_{l=1}^{p-1} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j}\right) \text{ subject to constraints (11)–(15)}\right\} \tag{16}$$

As u_k^* depends only on z_k, the above is a state feedback control law, defined implicitly through the optimization problem. MPC belongs to a class of optimal-control techniques called open-loop optimal-feedback control (OLOFC), as the open-loop optimal control computation is repeated with feedback update. It is natural to apply the same train of logic in developing a min-max predictive control algorithm.

Time-invariant parameter case. In this case, $A(\vartheta_{k+i}) = A(\vartheta)$ and $B(\vartheta_{k+i}) = B(\vartheta)$ for all i and k. We assume that $\vartheta \in \Theta$. The open-loop input sequence that minimizes the worst-case cost can be computed through the following optimization:

$$\min_{\mathscr{U}_k} \max_{\theta \in \Theta} \left\{x_{k+p}^{\mathrm{T}} Q_p x_{k+p} + \sum_{l=1}^{p-1} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j}\right\}, \tag{17}$$

with the constraints

$$x_{k+l} \in \mathscr{X}, \quad l = 1, \ldots, p \quad \forall \theta \in \Theta, \tag{18}$$

$$u_{k+j} \in \mathscr{V}, \quad j = 0, \ldots, q-1, \tag{19}$$

$$u_{k+i} = 0, \quad i = 1, \ldots, p-1, \tag{20}$$

and the prediction equation

$$x_{k+l} = A^l(\theta) z_k + \sum_{j=0}^{l-1} A^{l-j-1}(\theta) B(\theta) u_{k+j} \tag{21}$$

As before, the actual input v_k is set equal to the optimal value of u_k at each sample time. The resulting feedback control will be referred to as open-loop worst-case optimal-feedback control (OLWOFC). Open-loop minimization of the worst-case error, the idea used in formulating the above, represents the essence of most min-max MPC techniques that have appeared in the literature (Campo and Morari, 1987; Lau *et al.*, 1991; Veres and Norton, 1993).

There are two very important ramifications of the open-loop control assumption in the above formulation. The first is that update of the parameter bound Θ is not considered. For time-invariant systems, state measurements provide information about the constant parameter vector ϑ that can be used to reduce the size of the set Θ. This is true even when bounded disturbance and noise are present. There are algorithms in the literature that enable the updating of the feasible parameter set in the presence of bounded disturbances (see Milanese and Belforte, 1982; Fogel and Huang, 1982; Belforte *et al.*, 1990). However, the problem is far more complex than simple updating of the feasible parameter set at each time step. A more subtle aspect is that the choice of control input affects the feasible parameter set for *future* time steps. In the open-loop control computation, however, this relationship cannot be accounted for in the optimization (or it is not meaningful to do so), because the future inputs are assumed to be computed independently of future measurements. The second aspect is that the formulation fails to address the fact that only the first element of the optimal input trajectory is implemented and the whole min-max optimization is repeated at the next time step with a feedback update. In the subsequent optimization, the worst-case parameter values may change because of the feedback update. This can have some undesirable consequences. We will demonstrate this point through a simple example later in this section.

Time-varying parameter case. In the case where the parameter sequence is modeled as an independently varying bounded sequence as in (7), the open-loop minimization of the worst-case cost becomes

$$\min_{\mathscr{U}_k} \max_{[\theta_k, \ldots, \theta_{k+p-1}] \in \Theta \times \ldots \times \Theta} \left\{x_{k+p}^{\mathrm{T}} Q_p x_{k+p} + \sum_{l=1}^{p-1} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j}\right\}, \tag{22}$$

with the constraints

$$x_{k+l} \in \mathscr{X}, \quad l = 1, \ldots, p$$
$$\forall[\theta_k, \ldots, \theta_{k+p-1}] \in \Theta \times \ldots \times \Theta, \quad (23)$$
$$u_{k+j} \in \mathscr{V}, \quad j = 0, \ldots, q-1, \quad (24)$$
$$u_{k+i} = 0, \quad i = 1, \ldots, p-1, \quad (25)$$

and the prediction equation

$$x_{k+l} = \left(\prod_{i=0}^{l-1} A(\theta_{k+i})\right) z_k + \sum_{j=0}^{l-1} \left(\prod_{i=j+1}^{l-1} A(\theta_{k+i})\right) B(\theta_{k+j}) u_{k+j}. \quad (26)$$

To distinguish the above from the previous formulation, which was based on the constant parameter assumption, we refer to the above as OLWOFC-II. This distinction is necessary because there is an incentive to apply the above even to time-invariant systems. One of the incentives is that OLWOFC-II tends to be more robust than OLWOFC, as the parameters are allowed to take on differnet values at each time step. For instance, we can prove robust stability of OLWOFC-II under certain conditions (see Section 2.6). However, this formulation suffers from the same drawback as the first, in that it does not consider the benefit of future measurements in the prediction. Even when the parameters do not have time correlation, future measurements still contain information about the past parameter values that can be useful for control computation. A consequence is that the resulting closed-loop response can be more sluggish than necessary.

For the more general case of (9), a conservative formulation is

$$\min_{\mathscr{U}_k} \max_{[\theta_k, \ldots, \theta_{k+p-1}] \in \Theta_k \times \ldots \times \Theta_{k+p-1}} \left\{ x_{k+p}^{\mathrm{T}} Q_p x_{k+p} + \sum_{l=1}^{p-1} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j} \right\}, \quad (27)$$

with (23)–(26). In addition to being an open-loop formulation, it is conservative because time correlations in the parameters are ignored. On the other hand, entering the parameter model of (9) directly into the optimization can lead to the same problem as in the time-invariant case.

2.4. *Closed-loop formulation*

In the foregoing discussion, we have stated the need to consider the recurrent nature of the optimization in the formulation. Hence, the focus of this section is on formulating the worst-case optimal control from a closed-loop standpoint. We will start with the time-varying parameter case.

Time-varying parameter case. Let us consider the case where $\vartheta_k \in \Theta \forall k$. Hence, we do not consider any correlation among the parameter deviations at different times. We consider the same finite-horizon-cost-minimization problem as before. This time, however, we consider closed-loop control in the prediction to derive the worst-case optimal state feedback law. For this, we apply Bellman's principle of optimality (Bellman, 1961) and use dynamic programming (as in the derivation of the LQ regulator).

For the finite-horizon problem which ends at $t = k + p$, the cost for the min-max optimization at $t = k + p - 1$ for given x_{k+p-1} is

$$V_{k+p-1}^{k+p}(x_{k+p-1}) = \min_{u_{k+p-1} \in \mathscr{V}} \max_{\theta_{k+p-1} \in \Theta} \{x_{k+p}^{\mathrm{T}} Q_p x_{k+p} + u_{k+p-1}^{\mathrm{T}} R u_{k+p-1}\}, \quad (28)$$

with

$$x_{k+p} = A(\theta_{k+p-1}) x_{k+p-1} + B(\theta_{k+p-1}) u_{k+p-1} \quad (29)$$

and the constraint

$$x_{k+p} \in \mathscr{X} \quad \forall \theta_{k+p-1} \in \Theta. \quad (30)$$

Similarly, at $t = k + p - 2$, the cost can be stated as

$$V_{k+p-2}^{k+p}(x_{k+p-2}) = \min_{u_{k+p-2} \in \mathscr{V}} \max_{\theta_{k+p-2} \in \Theta} \{x_{k+p-1}^{\mathrm{T}} Q x_{k+p-1} + u_{k+p-2}^{\mathrm{T}} R u_{k+p-2} + V_{k+p-1}^{k+p}(x_{k+p-1})\}, \quad (31)$$

with

$$x_{k+p-1} = A(\theta_{k+p-2}) x_{k+p-2} + B(\theta_{k+p-2}) u_{k+p-2} \quad (32)$$

and the constraint

$$x_{k+p-1} \in \mathscr{X} \quad \forall \theta_{k+p-2} \in \Theta. \quad (33)$$

V_{k+p-1}^{k+p} depends on x_{k+p-1}, which in turn depends on x_{k+p-2}, the minimizer u_{k+p-2}, and the maximizer θ_{k+p-2} through (32). Applying this idea successively, we arrive at the following expression for the min-max cost $t = k$:

$$V_k^{k+p}(z_k) = \min_{u_k \in \mathscr{V}} \max_{\theta_k \in \Theta} \{x_{k+1}^{\mathrm{T}} Q x_{k+1} + u_k^{\mathrm{T}} R u_k + V_{k+1}^{k+p}(x_{k+1})\}, \quad (34)$$

$$x_{k+1} = A(\theta_k) z_k + B(\theta_k) u_k, \quad (35)$$

$$x_{k+1} \in \mathscr{X} \quad \forall \theta_k \in \Theta, \quad (36)$$

where $V^{k+p}_{k+1}(x_{k+1})$ is computed recursively from $i=p$ to $i=2$ through

$$V^{k+p}_{k+i-1}(x_{k+i-1}) = \min_{u_{k+i-1}\in\mathcal{V}} \max_{\theta_{k+i-1}\in\Theta} \{x^{\mathrm{T}}_{k+i}Q_i x_{k+i} + u^{\mathrm{T}}_{k+i-1}Ru_{k+i-1} + V^{k+p}_{k+i}(x_{k+i})\}, \quad (37)$$

$$x_{k+i} = A(\theta_{k+i-1})x_{k+i-1} + B(\theta_{k+i-1})u_{k+i-1}, \quad (38)$$

$$x_{k+i} \in \mathcal{X} \quad \forall\theta_{k+i-1} \in \Theta, \quad (39)$$

with the starting condition $V^{k+p}_{k+p}(x_{k+p})=0$ and $Q_1 = \ldots = Q_{p-1} = Q$. When v_k is set to the optimal value of u_k from (34) at every sample time, we name the resulting feedback control closed-loop worst-case optimal receding horizon control (CLWORHC). It is a noteworthy point that the computation of the worst-case optimal feedback control policy for a linear system with a convex parameter set is nonlinear and nonconvex.

Extension to time-invariant/time-correlated parameter case. In the time-invariant case (5), the closed-loop formulation is further complicated by the fact that input decisions affect the feasible parameter sets for future time steps in a complex, nonlinear manner. Accounting for this in the control computation offers certain advantages (such as the active probing trait found in dual control (see Feldbaum, 1965)), but doing so rigorously makes the control formulation computation unwieldy. For instance, the state vector must be extended to include information necessary to define the feasible parameter set (this is referred to as the 'hyperstate' in the adaptive control literature) and the effect of the input on those extended states must be quantified.

If we assume in the prediction that the parameter set will not be updated in the future time instances, we can go through the same cost-to-go calculation as before to arrive at the same exact optimization of (34)–(39) (where Θ is the feasible parameter set computed at $t=k$). However, the fact that we obtain the same algorithm as when the uncertainty is assumed to vary from one time step to next in an arbitrary manner clearly indicates the conservativeness. The conservativeness arises from the fact that we did not consider the parameter bound update in the control computation. The same observation holds for the more general time-correlated case (9).

Numerical solution procedure. CLWORHC, given by (34)–(39), involves a sequence of min-max optimizations formulated as a dynamic program. Since the min-max optimization at each stage does not yield an analytical solution, it must be solved numerically. The numerical procedure involves discretizing the states at each stage and computing and storing the min-max costs for all combinations of the discretized states. The procedure for doing this with constraints $u_{k+q} = \ldots = u_{k+p-1} = 0$ is given below.

Step 1. Discretize the state vector x_{k+q-1} and compute the following cost-to-go for each combination of the discrete states:

$$V^{k+p}_{k+q}(x_{k+q}) = \max_{\theta_{k+q},\ldots,\theta_{k+p-1}\in\Theta} \left\{ \sum_{i=q+1}^{p} x^{\mathrm{T}}_{k+i}Q_i x_{k+i} \right\}, \quad (40)$$

with

$$x_{k+i} = \left(\prod_{j=q}^{i-1} A(\theta_{k+j})\right)x_{k+q}, \quad i=q+1,\ldots,p. \quad (41)$$

If $q=p$, skip this step and set $V^{k+q}_{k+q}=0$.

Step 2. Set $l=q$.

Step 3. Discretize the state vector x_{k+l-1}.

Step 4. For each combination of discretized states, compute and store the cost for

$$\min_{u_{k+l-1}\in\mathcal{V}} \max_{\theta_{k+l-1}\in\Theta} \{x^{\mathrm{T}}_{k+l}Q_l x_{k+l} + u^{\mathrm{T}}_{k+l-1}Ru_{k+l-1} + V^{k+p}_{k+l}(x_{k+l})\}, \quad (42)$$

with

$$x_{k+l} = A(\theta_{k+l-1})x_{k+l-1} + B(\theta_{k+l-1})u_{k+l-1}, \quad (43)$$

$$x_{k+l} \in \mathcal{X}, \quad \forall\theta_{k+l-1} \in \Theta. \quad (44)$$

If $l=1$, also store the optimal value for u_{k+l-1}.

Step 5. Set $l=l-1$. If $l=0$, stop. If not, go back to Step 3.

Remarks.

(i) In Steps 1 and 3, the range of discretization should be chosen to reflect the constraints as well as the region of feasible operation. One can also include performance considerations. For instance, the allowed region in the state space can be made gradually smaller to reflect required speed of convergence.

(ii) In Step 4, $V^{k+p}_{k+l}(x_{k+l})$ is the min-max cost from the previous iteration, which should be in storage for discrete points of x_{k+l}. Since x_{k+l} is determined uniquely by x_{k+l-1}, u_{k+l-1}, and θ_{k+l-1}, V^{k+p}_{k+l} can be computed for given values of x_{k+l-1}, u_{k+l-1}, and θ_{k+l-1}, by interpolating the stored values.

(iii) In Step 4, it is required to find an input $u_{k+l-1} \in \mathcal{V}$ that keeps x_{k+l} within $\mathcal{X}$ for all $\theta_{k+l-1} \in \Theta$. If this is not possible, it implies that the state constraint is not feasible for the particular value of x_{k+l-1}. In this case, a 'very large' cost should be assigned to the value. This way, if the min-max cost at the end is declared 'very large', we know that constraint violation is inevitable for the particular value of z_k.

(iv) What results from the above is a feedback law $v_k = f(z_k)$, which is given in a table look-up form. One can use interpolation functions to construct an explicit function, if desired.

Undoubtedly, the above procedure is numerically demanding and suffers from the curse of dimensionality. For example, if there are d states and each state is discretized with v points, one needs to perform the min-max optimization and store the costs for v^d points. This can be a formidable task for large v and d. In addition, each min-max optimization is nonconvex in general. The complexity of optimization increases only linearly with the prediction horizon, which is a feature of dynamic programming. Even though the calculation can be performed off-line, the procedure is applicable only to a system of small dimension. Nevertheless, it is of interest for us to solve simple problems numerically in order to gain insights into properties of CLWORHC and assess the performances of other suboptimal algorithms.

2.5. *Example*

Example 1: *SISO integrating system with unknown delay.* We will demonstrate the closed-loop properties of different formulations through a simple example. For this, we consider the following integrating system with an unknown delay:

$$y_k = y_{k-1} + v_{k-\delta}, \quad \delta \in \{1, 2, 3\}. \tag{45}$$

The above can be converted into the following state–space system:

$$z_{k+1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ \vartheta_2 & \vartheta_3 & 1 \end{bmatrix} z_k + \begin{bmatrix} 1 \\ 0 \\ \vartheta_1 \end{bmatrix} v_k, \tag{46}$$

$$y_k = [0 \;\; 0 \;\; 1] z_k, \tag{47}$$

$$\vartheta \triangleq \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \\ \vartheta_3 \end{bmatrix} \in \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}. \tag{48}$$

Note that the above is in the form of (1).

We simulated the closed-loop responses of different min-max predictive control algorithms applied to the above system. In the simulation, we assumed that $z_0 = [0 \;\; 0 \;\; 1]^{\mathrm{T}}$ (which corresponds to a step disturbance occuring in the output at $t = 0$) and δ for the real plant was 2 ($\vartheta = [0 \;\; 1 \;\; 0]$). The parameters used for all control computations were $p = 3$, $q = 3$, $Q = Q_p = \mathrm{diag}\,\{0, 0, 1\}$ and $R = 0.0001$.

First, Figure 2 shows the response obtained by applying the OLWOFC (formulated with *time-invariant* uncertainty as in (17)–(21)). The simulation shows an oscillatory output response that is inferior to what can be achieved with a well-tuned PID controller. This poor performance can be attributed to the fact that the worst-case parameter value keeps changing from one time step to next due to the feedback. Figure 2(b) shows the simulated response of the output when we apply the CLWORHC. Here, we solved the dynamic program (34)–(39) to construct a feedback law and applied it to the plant. The simulation shows clear improvement in the output response when compared to that obtained with OLWOFC. Figure 2(b) also shows the simulation result obtained with OLWOFC-II (which is formulated with time-varying uncertainty of (22)–(26)). As we predicted, the response is farily good (better than OLWOFC, but slightly worst than CLWORHC).

2.6. *Robust stability of infinite-horizon min-max MPCs*

In this section, we prove robust stability of the min-max algorithms for the infinite-horizon case. The proof is based on the idea that the optimal infinite-horizon cost decreases monotonically with time. The same idea has also been used by others (e.g. Mayne and Michaska, 1990; Rawlings and Muske, 1993) to prove the attractivity of various linear and nonlinear MPC algorithms. The value of the result is twofold: first, it provides further justification for the algorithms; and, secondly, there are certain types of system (e.g. FIR models) for which an infinite-horizon problem can be directly converted into a finite-horizon problem.

Theorem 1. *Robust stability of infinite-horizon CLWORHC.* Let $f_\infty(z_k)$ denote the solution of the optimization (34)–(39) with $p = \infty$ for given z_k with $Q > 0$ and $R > 0$. Under the feedback law $v_k = f_\infty(z_k)$, the plant given by (1) and (7) is attractive for all possible parameter sequences over the set for which the infinite-horizon cost function for the optimization is bounded. In

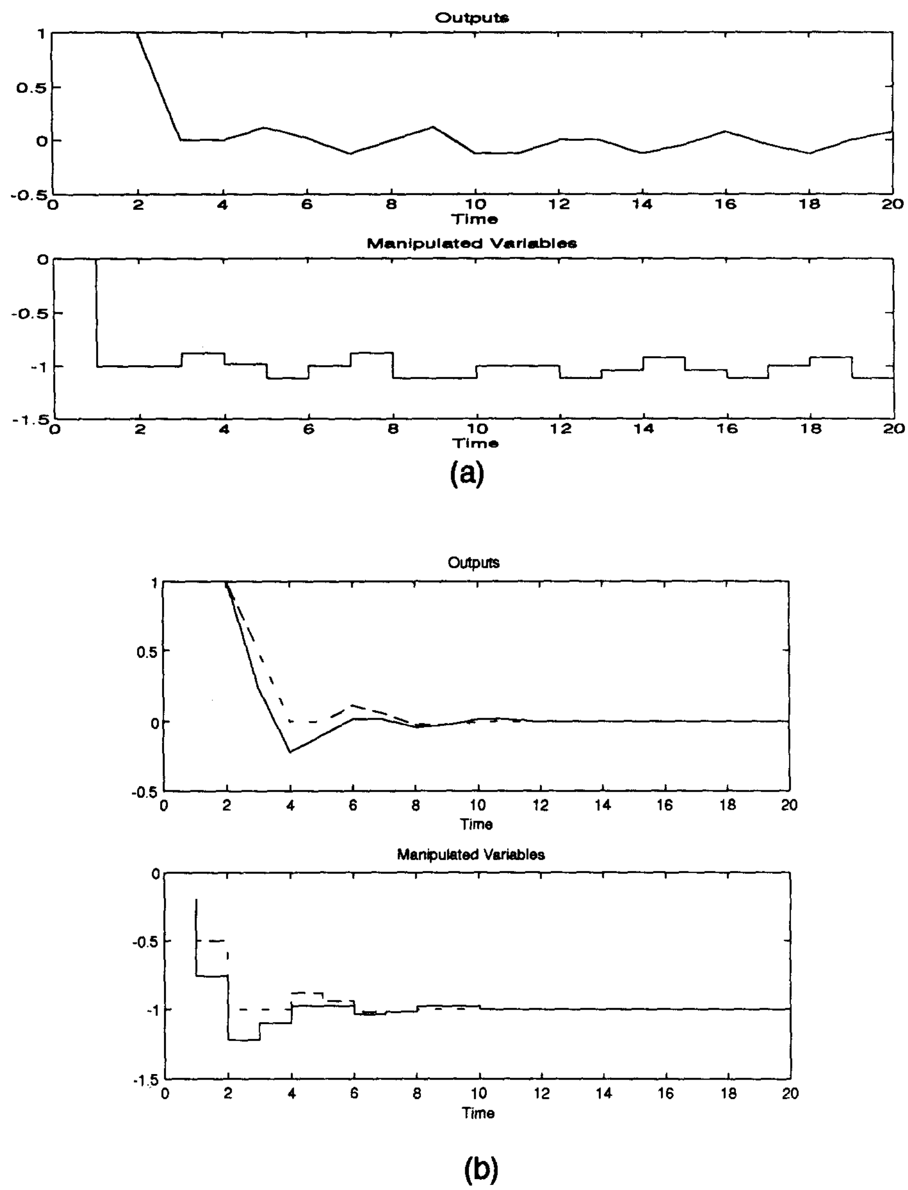


Fig. 2. Simulation of different min-max MPCs applied to the delay system of Example 1: (a) OLWOFC; (b) CLWORHC (----), OLWOFC-II (——).

other words, $z_k \to 0$ and $v_k \to 0$ as $k \to \infty$, for z_0 for which $V_0^\infty(z_0) < \infty$.

Proof. We will leave out the state constraints for simplicity. The proof with state constraints is exactly the same (as long as they are feasible). Note that, with constraints that $u_{k+q} = \ldots = u_{k+p-1} = 0$, we can rewrite (34) as the following nested optimization:

$$V_k^{k+p} = \min_{u_k \in \mathcal{U}} \max_{\theta_k \in \Theta} \min_{u_{k+1} \in \mathcal{U}} \max_{\theta_{k+1} \in \Theta} \ldots \min_{u_{k+q-1} \in \mathcal{U}} \max_{\theta_{k+q} \in \Theta, \ldots, \theta_{k+p-1} \in \Theta} \left\{ x_{k+p}^{\mathrm{T}} Q_p x_{k+p} + \sum_{l=1}^{p-1} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j} \right\}, \quad (49)$$

where

$$x_{k+l} = \left(\prod_{i=0}^{l-1} A(\theta_{k+i}) \right) z_k + \sum_{j=0}^{l-1} \left(\prod_{i=j+1}^{l-1} A(\theta_{k+i}) \right) B(\theta_{k+j}) u_{k+j}, \quad (50)$$

$$l = 1, \ldots, p.$$

Now, let us define V_k^∞ as:

$$V_k^\infty = \min_{u_k \in \mathcal{U}} \max_{\theta_k \in \Theta} \min_{u_{k+1} \in \mathcal{U}} \max_{\theta_{k+1} \in \Theta} \ldots \min_{u_{k+q-1} \in \mathcal{U}} \max_{\theta_{k+q-1} \in \Theta, \ldots} \left\{ \sum_{l=1}^{\infty} x_{x+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j} \right\}. \quad (51)$$

We claim that

$$V_{k+1}^\infty - V_k^\infty \le -(z_k^{\mathrm{T}} Q z_k + v_k^{\mathrm{T}} R v_k). \quad (52)$$

This is easy to show. Let u_k^* and θ_k^* be the arguments of the outermost optimization in the above. Then,

$$V_k^\infty = \min_{u_k\in\mathcal{V}} \max_{\theta_k\in\Theta} \min_{u_{k+1}\in\mathcal{V}} \max_{\theta_{k+1}\in\Theta} \ldots \min_{u_{k+q-1}\in\mathcal{V}} \max_{\theta_{k+q-1}\in\Theta,\ldots} \left\{\sum_{l=1}^{\infty} x_{k+l}^{\mathrm{T}} Q s_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j}\right\} \quad (53)$$

$$= \max_{\theta_k\in\Theta} \min_{u_{k+1}\in\mathcal{V}} \max_{\theta_{k+1}\in\Theta} \ldots \min_{u_{k+q-1}\in\mathcal{V}} \max_{\theta_{k+q-1}\in\Theta,\ldots} \left\{\sum_{l=1}^{\infty} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q-1} u_{k+j}^{\mathrm{T}} R u_{k+j}\right\}, \quad (54)$$

with $u_k = u_k^*$,

$$\geq \min_{u_{k+1}\in\mathcal{V}} \max_{\theta_{k+1}\in\Theta} \ldots \min_{u_{k+q-1}\in\Theta} \max_{\theta_{k+q-1}\in\Theta,\ldots} \left\{\sum_{l=1}^{\infty} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=0}^{q} u_{k+j}^{\mathrm{T}} R u_{k+j}\right\}, \quad (55)$$

with $u_k = u_k^*$ and $\theta_k = \vartheta_k$ since $\vartheta_k \in \Theta$,

$$= \min_{u_{k+1}\in\mathcal{V}} \max_{\theta_{k+1}\in\Theta} \ldots \min_{u_{k+q-1}\in\mathcal{V}} \max_{\theta_{k+q-1}\in\Theta,\ldots} \left\{\sum_{l=2}^{\infty} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=1}^{q} u_{k+j}^{\mathrm{T}} R u_{k+j}\right\} + z_{k+1}^{\mathrm{T}} Q z_{k+1} + v_k^{\mathrm{T}} R v_k,$$

with $x_{k+l} = \left(\prod_{i=1}^{l-1} A(\theta_{k+i})\right) z_{k+1} + \sum_{j=1}^{l-1} \left(\prod_{i=j+1}^{l-1} A(\theta_{k+i})\right) B(\theta_{k+j}) u_{k+j}$,

$l = 2, \ldots, p$,

since $v_k = u_k^*$ and $z_{k+1} = A(\vartheta_k) z_k + B(\vartheta_k) v_k$,

$$\geq \min_{u_{k+1}\in\mathcal{V}} \max_{\theta_{k+1}\in\Theta} \ldots \min_{u_{k+q-1}\in\mathcal{V}} \max_{\theta_{k+q-1}\in\Theta} \min_{u_{k+q}\in\mathcal{V}} \max_{\theta_{k+q}\in\Theta,\ldots} \left\{\sum_{l=2}^{\infty} x_{k+l}^{\mathrm{T}} Q x_{k+l} + \sum_{j=1}^{q} u_{k+j}^{\mathrm{T}} R u_{k+j}\right\} + z_{k+1}^{\mathrm{T}} Q z_{k+1} + v_k^{\mathrm{T}} R v_k,$$

since $u_{k+q} = 0 \in \mathcal{V}$,

$$= V_{k+1}^\infty + z_{k+1}^{\mathrm{T}} Q z_{k+1} + v_k^{\mathrm{T}} R v_k. \quad (56)$$

This means

$$V_k^\infty \geq V_{k+1}^\infty + z_{k+1}^{\mathrm{T}} Q z_{k+1} + v_k^{\mathrm{T}} R v_k, \quad (57)$$

which can be rearranged into (52). The inequality (52) implies that V_k^∞ is a converging sequence, and z_k and v_k must vanish as $k \to \infty$, implying the attractivity.

Remarks.

(i) The attractivity is proven only over the set of z_0 for which $V_0^\infty < \infty$. For open-loop stable systems, global attractivity can be established.

(ii) As mentioned above, the same proof holds without modification when the output constraints are present, as long as they are feasible. If this is not the case, the constraints must be relaxed in some way. For example, one can remove the constraints one by one, starting from the initial time, until they become feasible (as suggested by Rawlings and Muske, 1993). Another way is to include a quadratic term that penalizes the extent of constraint violation in the objective function (see Zheng and Morari, 1994). The stability proof works, with some modifications, in both cases.

Corollary 1. *Robust stability of infinite horizon OLWOFC-II.* Let $\hat{f}_\infty(z_k)$ represent the solution of the optimization (22)–(26) for given z_k with $Q > 0$ and $R > 0$ with $p = \infty$. With the feedback law $v_k = \hat{f}_\infty(z_k)$, the plant given by (1) and (7) is attractive for all possible parameter sequences over the set for which the infinite horizon cost function for the optimization is bounded.

Proof. Same as the proof of Theorem 1.

Remarks.

(i) The corollary holds only when the infinite horizon cost for the *open-loop* optimization is bounded. This means that there cannot be any uncertainty associated with unstable modes in the system.

(ii) The infinite horizon algorithm can be implemented, in certain cases, like the FIR system. See Section 3.2.

(iii) Note that the proof does not work if we require the parameter vector to be time-invariant in the formulation (as in OLWOFC). This is because the worst-case parameter for the optimization at the next time step may change because of the feedback. We can no longer show that the cost is monotonically decreasing.

2.7. *Summary*

For systems with time-varying parametric uncertainty (as defined by (7)), both CLWORHC and OLWOFC-II can be used. CLWORHC gives the worst-case optimal feedback control (for the finite-horizon problem defined). OLWOFC-II, on the other hand, neglects the benefit of the future measurements and, therefore, is suboptimal. However, the open-loop formulation still provides good robustness as demonstrated through the example and Corollary 1.

For systems with time-invariant or time-correlated parametric uncertainties (as in (5) or (9)), the open-loop formulation based on the given uncertainty description can yield poor results. This is mainly because the future feedback can change the worst-case parameter values. In addition, the benefits of reduced uncertainty drawn from future parameter updates cannot be reflected in the open-loop formulation. Rigorous closed-loop formulation, on the other hand, is very difficult in this case, owing to the complex relationship between the states, inputs, parameters, and feasible parameter set for future times. The best option is to neglect the future updates of the feasible parameter set in the prediction. Then we obtain the same closed-loop formulation as when the uncertainty is allowed to vary with time. This approximation generally results in robust, but conservative, control. The conservatism is due to the fact that future updates of the parameter set are not considered in the prediction. One can also apply OLWOFC-II (the open-loop formulation with time-varying uncertainty), but at the expense of a further increase in conservatism. Finally, although ignored in the prediction, updating of the parameter set can still be performed at each time step.

3. SUBOPTIMAL ALGORITHMS

As the dynamic programming solution is computationally prohibitive for large systems, it is of practical interest to develop suboptimal, but more computationally amenable, algorithms. In this section, we restrict our discussion to moving-average (MA) systems with an integrator. First, we show that, when the parameter set is convex, the optimization problem for OLWOFC-II is convex and, therefore, can potentially be solved on-line. Next, we restrict the parameter set to ellipsoids and approximate the objective function to arrive at a function similar to the one used in the conventional MPC (i.e. nominal error plus quadratic input penalty terms). This suboptimal algorithm has a nice interpretation of penalizing various orthogonal linear combinations of the input sequence according to the magnitudes of the respective error bounds in the parameter space.

3.1. *Model description*

In this section, we will mainly consider the following SISO MA system with an integrator:

$$y_k = y_{k-1} + \vartheta_{k-1}^{\mathrm{T}} V_{k-1}, \tag{58}$$

with

$$V_{k-1} = [v_{k-1}, \ldots, v_{k-n}]^{\mathrm{T}}, \tag{59}$$

$$\vartheta_{k-1}^{\mathrm{T}} = [h_1, \ldots, h_n]. \tag{60}$$

In the above, y_k is the output at time k, V_{k-1} is the vector containing the n past input sequence, and ϑ is the parameter vector containing n MA coefficients. We assume that $\{\vartheta_k\}$ is, in general, an independent sequence the elements of which belong to a *convex* set Θ for all k. Hence, it includes the time-invariant parameter case as a special case. The model form of (58) is used in many popular predictive control algorithms like dynamic matrix control (Cutler and Ramaker, 1980) to model open-loop stable systems with a finite impulse response. The integrator is added to account for 'persistent-type' errors in the output. Hence, for stable systems, v_k represents the *incremental change* in the actuator position at $t = k$. Note that we can put the input–output system (58) into the standard state-space form as follows:

$$\begin{bmatrix} y_{k+1} \\ v_k \\ \vdots \\ \vdots \\ v_{k-n+1} \end{bmatrix} = \begin{bmatrix} 1 & h_{2,k} & \cdots & h_{n,k} & 0 \\ 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \times \begin{bmatrix} y_k \\ v_{k-1} \\ \vdots \\ \vdots \\ v_{k-n} \end{bmatrix} + \begin{bmatrix} h_{1,k} \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} v_k. \tag{61}$$

We are interested in minimizing

$$\sum_{k=0}^{\infty} y_k^2 + \lambda v_k^2. \tag{62}$$

The minimization is subject to the following constraint:

$$v_{\min} \le v_k \le v_{\max}, \tag{63}$$

where $v_{\min} < 0$ and $v_{\max} > 0$. In addition to the above, one may have the constraint

$$v_{\min}^{\mathrm{int}} \le \sum_{i=0}^{k} v_i + b \le v_{\max}^{\mathrm{int}}. \tag{64}$$

The constraint expresses the actuator limit, as v represents the incremental change in the actuator position. The above constraints will be assumed in the subsequently presented formulations, even though we do not write them out explicitly every time.

Although we work with the above specific model form, the results presented subsequently can be extended straightforwardly to more general cases of (i) autoregressive moving average (ARMA) systems with unknown, but bounded, moving average coefficients, and (ii) state–space systems with an input gain matrix parameterized through an unknown, bounded vector.

3.2. *OLWOFC as convex optimization*

We start from the optimization for CLWORHC applied to the system (58):

$$\min_{u_k}\max_{\theta_k\in\Theta}\min_{u_{k+1}}\max_{\theta_{k+1}\in\Theta}\dots\min_{u_{k+q-1}}\max_{[\theta_{k+q-1},\dots,\theta_{k+p-1}]\in\Theta\times\dots\times\Theta}\left\{\sum_{l=1}^{p}\tilde{y}^2_{k+l|k}+\sum_{j=0}^{q-1}\lambda u^2_{k+j}\right\},\quad(65)$$

where

$$\tilde{y}_{k+l|k}=y_k+\sum_{j=1}^{l}\theta^{\mathrm{T}}_{k+j-1}U_{k+j-1},\quad(66)$$

$$U_{k+j-1}=[\tilde{u}_{k+j-1},\dots,\tilde{u}_{k+j-n}]^{\mathrm{T}},\quad(67)$$

with

$$\tilde{u}_{k+i}=\begin{cases}v_{k+i}, & \text{for } i<0,\\ u_{k+i}, & \text{for } 0\le i<q,\\ 0, & \text{for } i\ge q,\end{cases}\quad(68)$$

and constraints (63) and (64). The above optimization is nonconvex and, therefore, is very difficult to solve.

As an alternative, we may consider solving the following optimizaion in OLWOFC-II:

$$\min_{\mathcal{U}_k\in\hat{U}_k}\max_{[\theta_k,\dots,\theta_{k+p-1}]\in\Theta\times\dots\times\Theta}\left\{\sum_{l=1}^{p}\tilde{y}^2_{k+l|k}+\sum_{j=0}^{q-1}\lambda u^2_{k+j}\right\},\quad(69)$$

where $\hat{U}_k$ denotes the feasible set for $\mathcal{U}_k$ (defined by constraints (63) and (64)). Recall the result from the previous section that replacing (65) with (69) increases conservatism, but the robustness is retained. A major benefit is that the optimization given in (69) is convex. This is easy to see. First, note that $\{\sum_{l=1}^{p}\tilde{y}^2_{k+l|k}+\sum_{j=0}^{q-1}\lambda u^2_{k+j}\}$ is a convex functional of $[\theta^{\mathrm{T}}_k,\dots,\theta^{\mathrm{T}}_{k+p-1}]^{\mathrm{T}}$, because it contains only quadratic and linear terms. It is also convex in $\mathcal{U}_k$ for the same reason. This has two consequences. First, the maximum always lies on the boundary of $\Theta\times\dots\times\Theta$. If Θ is a polytope, the maximum occurs at one of the vertices of $\Theta\times\dots\times\Theta$. Secondly, $\max_{\theta_k\in\Theta,\dots,\theta_{h+p-1}\in\Theta}\{\sum_{l=1}^{p}\tilde{y}^2_{k+l|k}+\sum_{j=1}^{q-1}\lambda^2_{k+j}\}$ is a convex function of $\mathcal{U}_k$ as the maximum of the convex functionals is also convex. Although convex, the objective function (69) is not necessarily differentiable, and standard gradient-based algorithms cannot be applied in finding optimal $\mathcal{U}_k$. See Boyd and Barratt (1991) for the types of algorithm that can be applied to minimization of non-differentiable, but convex, functionals. In addition, note that the constraints on $\mathcal{U}_k$ (from (63) and (64)) are simply linear constraints. This means that a local minimum is also the global minimum. The convexity provides some flexibility in terms of implementation. For instance, instead of solving the optimization off-line for all possible discretized values of states (which leads to the curse of dimensionality and the need for interpolation), one can solve it on-line at each sample time for particular value of z_k, as is done in conventional MPC implementation.

Corollary 1 states that robust stability can be guaranteed for infinite-horizon OLWOFC-II. However, the infinite-horizon algorithm cannot be implemented directly as the cost is unbounded due to the presence of the integrator. For the FIR system, the output settles to a constant value at $k+q+n-1$ with the restriction of $u_{k+q+i}=0$ for $i\ge 0$. Hence, one could approximate the infinite-horizon MPC with

$$\min_{\mathcal{U}_k\in\hat{U}_k}\max_{[\theta_k,\dots,\theta_{k+q+n-2}]\in\Theta\times\dots\times\Theta}\left\{\gamma\tilde{y}^2_{k+q+n-1|k}+\sum_{l=1}^{k+q+n-2}\tilde{y}^2_{k+l|k}+\sum_{j=0}^{q-1}\lambda u^2_{k+j}\right\},\quad(70)$$

for which $\gamma\gg\max\{1,\lambda\}$. A better way to implement the infinite horizon MPC is to solve the following multiobjective optimization:

Primary objective. Steady-state error minimization:

$$\min_{\mathcal{U}_k\in\hat{U}_k}\max_{[\theta_k,\dots,\theta_{k+q+n-2}]\in\Theta\times\dots\times\Theta}\{\tilde{y}^2_{k+q+n-1|k}\},\quad(71)$$

$$\tilde{y}_{k+q+n-1|k}=y_k+\sum_{j=1}^{q+n-1}\theta^{\mathrm{T}}_{k+j-1}U_{k+j-1}.\quad(72)$$

Secondary objective. Dynamic error minimization:

$$\min_{\mathcal{U}_k\in\hat{U}_k}\max_{[\theta_k,\dots,\theta_{k+q+n-3}]\in\Theta\times\dots\times\Theta}\left\{\sum_{l=1}^{k+q+n-2}\tilde{y}^2_{k+l|k}+\sum_{j=0}^{q-1}\lambda u^2_{k+j}\right\}.\quad(73)$$

In the above, the degrees of freedom left after the steady-state error minimization are to be adjusted to minimize the dynamic error. Whether or not such degrees of freedom remain depends on the uncertainty parameterization and the number of input moves calculated.

Under the feedback control defined by the above, the input and output can be shown to converge to zero when (i) the initial condition is such that the constraint (64) on the integrated input remains inactive, and (ii) the sign of the steady-state gain (which is defined as $g^{ss}_k=\sum_{i=1}^{n}h_{i,k+i-1}$, not $g^{ss}_k=\sum_{i=1}^{n}h_{i,k}$) is the same for all possible parameter values at all time steps. The proof for $q=1$ is given in the Appendix. We also discuss in the Appendix how the proof extends to the general choice of q. (See also Lee and Cooley (1997) for a similar proof for state-space systems.) The formal proof for the general case is quite technical, and is not presented due to the space limitation.

Finally, we note that use of other norms are possible. For instance, replacing 2-norm with 1-norm in the above results in a linear-programming problem.

3.3. *Derivation of conventional MPC from min-max MPC*

Let us further assume that Θ is an ellipsoid given by:

$$\Theta \triangleq \{\theta : \|W(\theta - \tilde{\vartheta})\|_2 \leq 1,\ \theta \in \mathscr{R}^n\}. \quad (74)$$

In the above, $\tilde{\vartheta}$ represents the center of the ellipsoid and corresponds to the nominal value of the parameter vector. $W \in \mathscr{R}^{m \times n}$ is in the form of

$$W \triangleq \mathrm{diag}\left[\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_m}\right] V^{\mathrm{T}},$$

where V is an $n \times m$ orthogonal matrix, i.e. its m columns are made up of n-dimensional orthogonal vectors of unit length. $v_l \in \mathscr{R}^n$, the lth column of V, defines the lth principal axis of the ellipsoid, the corresponding radius of which is represented by σ_l. Note that m, the number of orthogonal vectors, can be smaller than n in the case when the parameter set belongs to a lower dimensional subspace.

With this parameter set, we can derive the following upper bound for the worst-case error:

$$\max_{[\theta_k, \ldots, \theta_{k+l-1}] \in \Theta \times \ldots \times \Theta} \tilde{y}^2_{k+l|k}, \quad (75)$$

$$\leq 2\left\{\epsilon^2_{k+l} + \max_{[\theta_k, \ldots, \theta_{k+l-1}] \in \Theta \times \ldots \times \Theta} \left[\sum_{j=1}^{l} (\theta_{k+j-1} - \tilde{\vartheta})^{\mathrm{T}} U_{k+j-1}\right]^2\right\}, \quad (76)$$

$$\leq 2\left\{\epsilon^2_{k+l} + \max_{[\theta_k, \ldots, \theta_{k+l-1}] \in \Theta \times \ldots \times \Theta} l \sum_{j=1}^{l} [(\theta_{k+j-1} - \tilde{\vartheta})^{\mathrm{T}} W^{\mathrm{T}} (W^{\mathrm{T}})^{-1} U_{k+j-1}]^2\right\}, \quad (77)$$

$$\leq 2\left\{\epsilon^2_{k+l} + l\left[\max_{[\theta_k, \ldots, \theta_{k+l-1}] \in \Theta \times \ldots \times \Theta} \sum_{j=1}^{l} \|W(\theta_{k+j-1} - \tilde{\vartheta})\|_2^2 \, \|(W^{\mathrm{T}})^{-1} U_{k+j-1}\|_2^2\right]\right\}, \quad (78)$$

$$= 2\left\{\epsilon^2_{k+l} + l \sum_{j=1}^{l} \|(W^{\mathrm{T}})^{-1} U_{k+j-1}\|_2^2\right\}, \quad (79)$$

where

$$\epsilon_{k+l} = y_k + \sum_{j=1}^{l} \tilde{\vartheta}^{\mathrm{T}} U_{k+j-1} \quad (80)$$

and

$$(W^{\mathrm{T}})^{-1} = \mathrm{diag}[\sigma_1, \ldots, \sigma_m] V^{\mathrm{T}}. \quad (81)$$

The upper bound (76) can be interpreted as splitting the worst-case error (75) into two terms: the first term $\epsilon^2(k+l)$ represents the error when $\theta_k, \ldots, \theta_{k+l-1}$ all take on their nominal value of $\tilde{\vartheta}$ and the second term is the worst-case error due to the deviation of θ from $\tilde{\vartheta}$. In going from (75) to (76) and from (76) to (77) we used the property

$$\left(\sum_{i=1}^{l} a_i\right)^2 \leq l \sum_{i=1}^{l} a_i^2 \quad (82)$$

In addition, in going from (77) to (78), we used the Schwarz inequality (i.e. $(x^{\mathrm{T}}y)^2 \leq \|x\|_2^2 \, \|y\|_2^2$).

Similarly, we can derive a lower bound for (75):

$$\max_{[\theta_k, \ldots, \theta_{k+l-1}] \in \Theta \times \ldots \times \Theta} \tilde{y}^2_{k+l|k}, \quad (83)$$

$$\geq \left\{\epsilon^2_{k+l} + \max_{[\theta_k, \ldots, \theta_{k+l-1}] \in \Theta \times \ldots \times \Theta} \left[\sum_{j=1}^{l} (\theta_{k+j-1} - \tilde{\vartheta})^{\mathrm{T}} U_{k+j-1}\right]^2\right\}, \quad (84)$$

$$\geq \left\{\epsilon^2_{k+l} + \max_{[\theta_k, \ldots, \theta_{k+l-1}] \in \Theta \times \ldots \times \Theta} \sum_{j=1}^{l} [(\theta_{k+j-1} - \tilde{\vartheta})^{\mathrm{T}} W^{\mathrm{T}} (W^{\mathrm{T}})^{-1} U_{k+j-1}]^2\right\}, \quad (85)$$

$$= \left\{\epsilon^2_{k+l} + \sum_{j=1}^{l} \|(W^{\mathrm{T}})^{-1} U_{k+j-1}\|_2^2\right\}. \quad (86)$$

In going from (83) to (84) and from (84) to (85), we used

$$\left(\sum_{i=1}^{l} a_i\right)^2 \geq \sum_{i=1}^{l} a_i^2, \quad (87)$$

when all a_i terms have same sign. This inequality can be used, since ϵ_{k+l} and $(\theta_{k+j-1} - \tilde{\vartheta})^{\mathrm{T}} U_{k+j-1}$, $1 \leq j \leq l$ all have the same sign for the worst-case choice of $\theta_k, \ldots, \theta_{k+l-1} \in \Theta$, because Θ is a norm-bounded set centered on $\tilde{\vartheta}$. In addition, (86), which can be proven to be an upper bound of (85) using the Schwarz inequality, can be proven to be equal to (85) by showing that the upper bound is always achievable for a certain choice of $\theta_k, \ldots, \theta_{k+l-1} \in \Theta$.

Minimization of the open-loop worst-case error based on the upper bound (79) is:

$$\min_{\mathscr{U}_k \in \hat{U}_k} \sum_{l=1}^{p} \left\{\left(y_k + \sum_{j=1}^{l} \tilde{\vartheta}^{\mathrm{T}} U_{k+j-1}\right)^2 + \lambda u^2_{k+l-1} + l \sum_{j=1}^{l} \|(W^{\mathrm{T}})^{-1} U_{k+j-1}\|_2^2\right\}. \quad (88)$$

In the above, the first term in (88) can be interpreted as the squared sum of nominal error (i.e. the error when assuming $\theta = \tilde{\vartheta}$), while the second and third terms can be viewed as input penalty terms. The third term, which arises from the model error, has the nice physical interpretation that various orthogonal linear combinations of the input sequence U_{k+l-1} (i.e. $v_i^T U_{k+l-1}$, $1 \le i \le n$) are penalized in proportion to σ_i, representing the largest possible deviation of θ from the nominal parameter in the respective direction. The coefficient l multiplying the penalty term for U_{k+l-1} arises because of the upper bound approximation (77) and the integrating nature of the model.

In the same manner, minimization based on the lower bound (86) becomes

$$\min_{\mathcal{U}_k \in \hat{U}_k} \sum_{l=1}^{p} \left\{ \left(y_k + \sum_{j=1}^{l} \tilde{\vartheta}^T U_{k+j-1} \right)^2 + \lambda u_{k+l-1}^2 + \sum_{j=1}^{l} \|(W^T)^{-1} U_{k+j-1}\|_2^2 \right\}. \quad (89)$$

It is noteworthy that the lower bound has the same structure as the upper bound. The only difference is that the coefficient multiplying the penalty term for U_{k+j-1} is smaller. This implies that minimization of the upper bound will generally lead to more cautious control than that of the lower bound. Another noteworthy point is that the lower bound turns out to be exactly the sum of the variances of the future output errors (plus the input weighting term $\sum_{l=1}^{p} \lambda u_{k+l-1}$) when the parameter vector is modeled as a *Gaussian* variable with covariance $(W^T W)^{-1}$, and the future decision variables are treated as deterministic variables (see Lee and Cooley, 1995).

Given the fact that minimizations of the upper and lower bounds are both interpreted as minimization of nominal error plus appropriate penalty terms for finite input sequences U_{k+j-1} and differ only by a scalar multiplying the weighting matrix, it is reasonable to minimize a function obtained by linearly interpolating the two functions. The interpolation parameter can be left as an adjustable parameter. This is admittedly *ad hoc*, because it assumes that the true objective function is well approximated by linearly interpolating the upper and lower bounds, but it adds to the tunability of the algorithm. The interpolation leads to

$$\min_{\mathcal{U}_k \in \hat{U}_k} \sum_{l=1}^{p} \left\{ \left(y_k + \sum_{j=1}^{l} \tilde{\vartheta}^T U_{k+j-1} \right)^2 + \lambda u_{k+l-1}^2 + [\zeta l + (1-\zeta)] \sum_{j=1}^{l} \|(W^T)^{-1} U_{k+j-1}\|_2^2 \right\}. \quad (90)$$

In the above, ζ is the interpolation parameter the value of which ranges from 0 to 1. Choosing $\zeta = 0$ is equivalent to the lower-bound minimization, while choosing $\zeta = 1$ results in the upper-bound minimization. It should be intuitively clear that, in general, increasing ζ increases the robustness, but slows down the closed-loop speed. Hence, one could start out with the most conservative setting of $\zeta = 1$ and decrease it slowly to reach a desirable response. For convenience of exposition, we will refer to (90) as suboptimal min-max predictive control (SMMPC) in the examples. The minimization is a linear least-squares problem in the unconstrained case and is a QP in the constrained case. The expressions for the hessian and gradient of the QP can be found in the Appendix. Compared to conventional MPC, the main difference is that the penalty term for current/future input decision variables is affected by past inputs through the nondiagonal weighting matrix.

These ideas generalize straightforwardly to multivariable systems. For example, for $n_y \times n_u$ systems, (90) becomes

$$\min_{\mathcal{U}_k \in \hat{U}_k} \left(\sum_{l=1}^{p} \sum_{i=1}^{n_y} \left\{ \left((y_i)_k + \sum_{j=1}^{l} \tilde{\vartheta}_i^T U_{k+j-1} \right)^2 + [\zeta l + (1-\zeta)] \sum_{j=1}^{l} \|(W_i^T)^{-1} U_{k+j-1}\|_2^2 \right\} + \sum_{j=0}^{q-1} u_{k+j}^T R u_{k+j} \right). \quad (91)$$

In the above, $U_{k+l-1} = [u_{k+l-1}^T \ldots u_{k+l-n}^T]^T$, the $\tilde{\vartheta}_i$ terms are vectors of dimension $n \cdot n_u$, and the $(W_i^T)^{-1}$ terms are also matrices of $m \times (n \cdot n_u)$. Having multiple inputs only increases the dimension of the FIR input space and the parameter space. The error for each output is then calculated independently and added together to make up the objective function in (91).

3.4. *More examples*

Example 2. SISO case with ∞-norm-bounded model set. We study the following example given by Zheng and Morari (1993):

$$y_k = y_{k-1} + a_1 u_{k-1} + a_2 u_{k-2},$$

$$\text{where} \quad \left\| \begin{bmatrix} \frac{1}{0.25} & 0 \\ 0 & \frac{1}{0.1} \end{bmatrix} \begin{bmatrix} a_1 - 0.75 \\ a_2 - 0.5 \end{bmatrix} \right\|_\infty \le 1. \quad (92)$$

Figure 3 compares the performance of OLWOFC, CLWORHC, and OLWOFC-II, when $y_0 = 1$ (step disturbance in the output at $t = 0$) and $a_1 = 0.75$ and $a_2 = 0.5$. We see that all algorithms work well for the nominal plant,

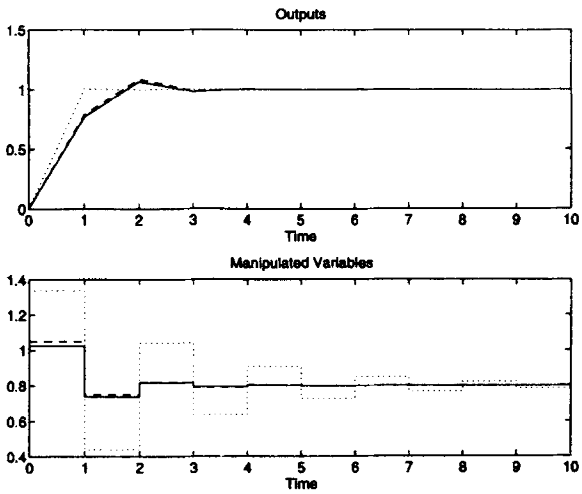


Fig. 3. Comparisons of open-loop and closed-loop algorithms with plant parameters $a_1 = 0.75$ and $a_2 = 0.5$ for Example 2: (——) OLWOFC-II; (- - - -) CLWORHC; (· · · ·) OLWOFC.

particularly OLWOFC. Figure 4 gives the same comparison for the case when $a_1 = 1$ and $a_2 = 0.4$. In the latter case, the OLWOFC algorithm becomes unstable, while the other algorithms remain stable.

Example 3. *SISO case with* 2*-norm-bounded model set.* The purpose of this example is to compare the performance of different suboptimal algorithms. For this purpose, we consider the following model with 2-norm bounded parameter vector:

$$y_k = y_{k-1} + a_1 u_{k-1} + a_2 u_{k-2},$$

$$\text{where} \quad \left\| \begin{bmatrix} \frac{1}{0.3} & 0 \\ 0 & \frac{1}{0.4} \end{bmatrix} \begin{bmatrix} a_1 - 1.0 \\ a_2 - 0.5 \end{bmatrix} \right\|_2 \leq 1. \quad (93)$$

Again, we set $y_0 = 1$, meaning we have a step disturbance starting at $t = 0$. Both the prediction horizon p and the control move horizon q are chosen to be 2, and the input penalty factor λ is set to 0. We compare the simulated responses obtained with OLWOFC-II, and with SMMPC for three different values of the interpolation parameter ($\zeta = 0, 0.65, 1$). $\zeta = 1$ corresponds to the upper-bound minimization of (88), while $\zeta = 0$ corresponds to the lower-bound minimization of (89). As it is difficult to find the worst-case scenario analytically, we gridded the boundary of the ellipsoid with 100 points and tried simulations for all these points. The worst case turned out to be $a_1 = 0.9812$ and $a_2 = 0.8992$. Figure 5 presents the simulation results for the worst case. The level of performance for SMMPC is close to that of computationally more expensive OLWOFC-II algorithm. We also see that input moves become less aggressive as ζ is increased. This is in accordance with our expectation.

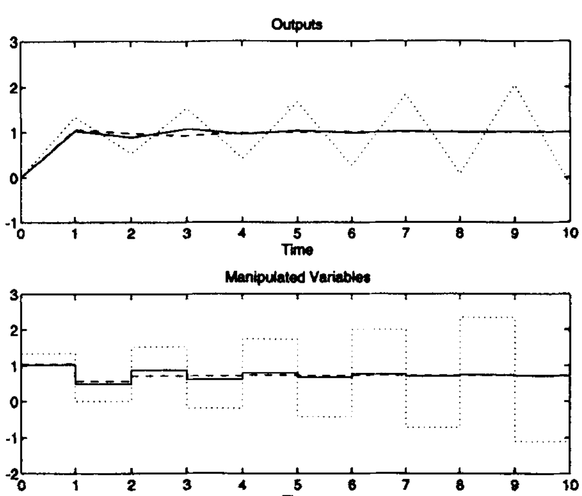


Fig. 4. Comparisons of open-loop and closed-loop algorithms with plant parameters $a_1 = 1$ and $a_2 = 0.4$ for Example 2: (——) OLWOFC-II; (- - - -) CLWORHC; (· · · ·) OLWOFC.

Example 4. *MISO high-purity distillation control problem.* The purpose of this example is to test the SMMPC algorithm further. We consider a control problem derived from an ideal distillation column studied by Skogestad *et al.* (1988). The control configuration we use consists of two manipulated variables (the reflux flow and vapor boilup) and one controlled variable (the overhead composition). We assume the real system is represented by

$$y(s) = \frac{1}{10s+1}[0.878(1+\delta_1) \quad 0.864(1+\delta_2)] \times \begin{bmatrix} L(s) \\ V(s) \end{bmatrix}, \quad (94)$$

where y represents the controlled output, L and V represent the two manipulated inputs, and δ_1 and δ_2 represent relative errors in the input channels. With

$$\left\| \begin{bmatrix} \frac{\delta_1}{\gamma_1} \\ \frac{\delta_2}{\gamma_2} \end{bmatrix} \right\|_2 \leq 1,$$

we allow maximum relative errors of γ_1 and γ_2 in the effect of manipulated inputs v_1 and v_2, respectively. We can approximate the above model as an integrating MA system in the form of (58) (with sample time of 1 minute, 30 coefficients were judged to be sufficient):

$$y_k = y_{k-1} + \vartheta^{\mathrm{T}} \begin{bmatrix} v_{k-1} \\ v_{k-2} \\ \vdots \\ v_{k-30} \end{bmatrix}, \quad (95)$$

$$\vartheta \in \Theta \triangleq \{\theta : \|W(\theta - \tilde{\vartheta})\|_2 \leq 1\},$$

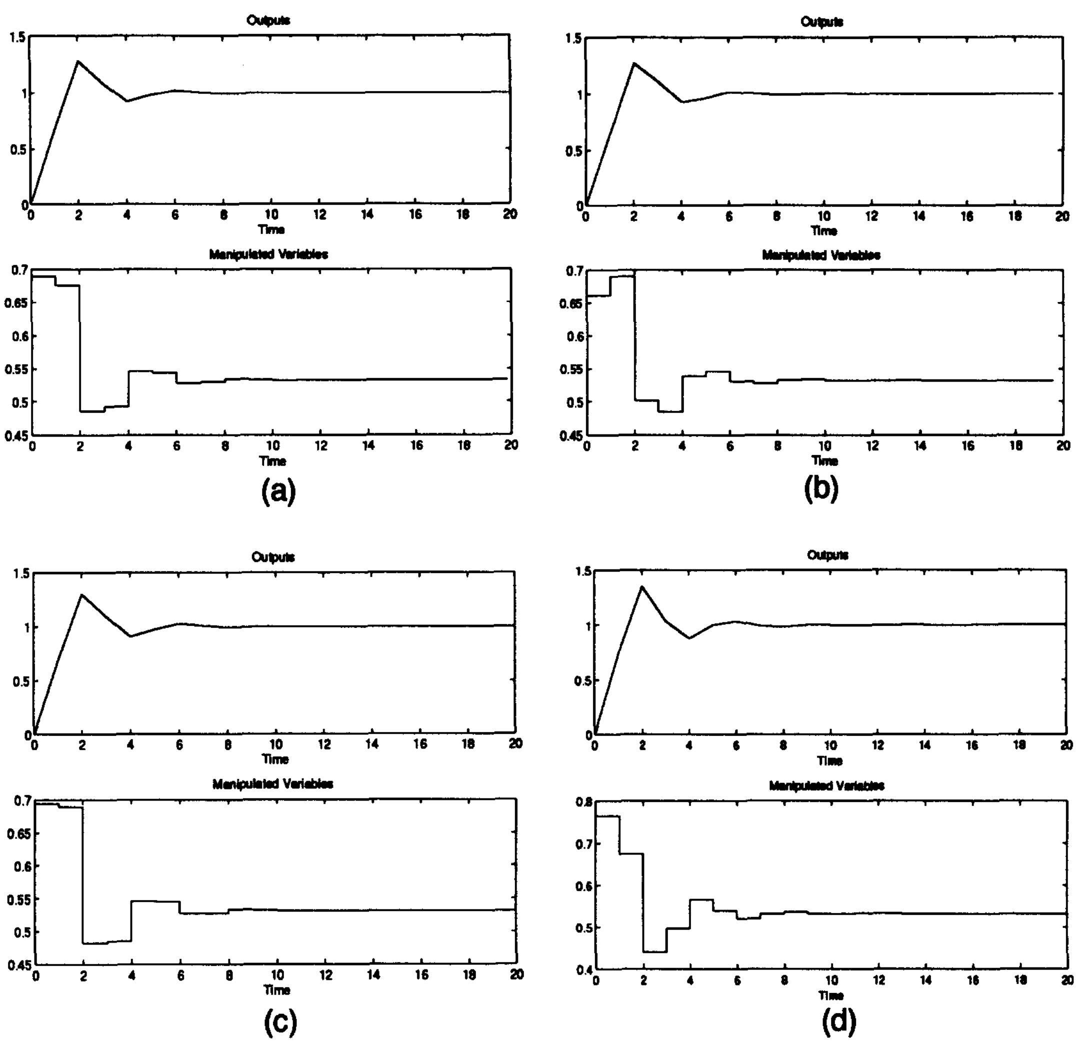


Fig. 5. Simulation results of various suboptimal algorithms with plant parameters $a_1 = 0.9812$ and $a_2 = 0.8992$ for Example 3: (a) OLWOFC-II; (b) SMMPC with $\zeta = 1$; (c) SMMPC with $\zeta = 0.65$; (d) SMMPC with $\zeta = 0$.

where

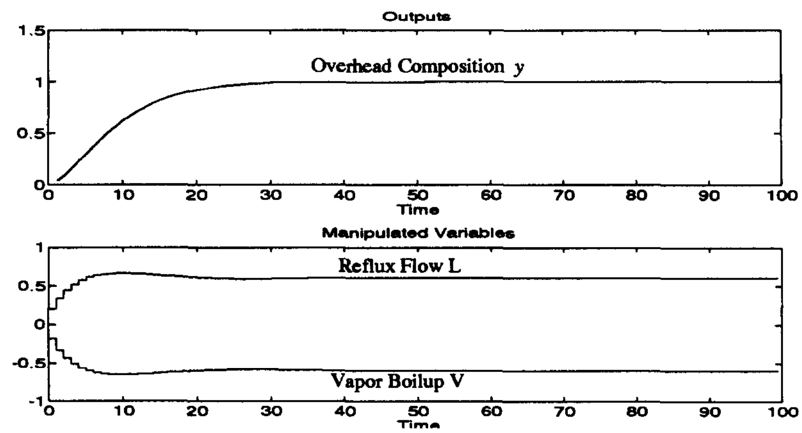
$$\tilde{\vartheta} = [\tilde{h}_{1,1} \;\; \tilde{h}_{1,2} \;\; \tilde{h}_{2,1} \;\; \tilde{h}_{2,2} \;\; \cdots \;\; \cdots \;\; \tilde{h}_{30,1} \;\; \tilde{h}_{30,2}]^{\mathrm{T}},$$

$$W = \begin{bmatrix} \dfrac{1}{\gamma_1 |\tilde{h}_1|} & 0 \\ 0 & \dfrac{1}{\gamma_1 |\tilde{h}_2|} \end{bmatrix} \times \begin{bmatrix} \dfrac{\tilde{h}_{1,1}}{|\tilde{h}_1|} & 0 & \cdots & \dfrac{\tilde{h}_{30,1}}{|\tilde{h}_1|} & 0 \\ 0 & \dfrac{\tilde{h}_{1,2}}{|\tilde{h}_2|} & \cdots & 0 & \dfrac{\tilde{h}_{30,2}}{|\tilde{h}_2|} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{\gamma_1 |\tilde{h}_1|} & 0 \\ 0 & \dfrac{1}{\gamma_2 |\tilde{h}_2|} \end{bmatrix} V^{\mathrm{T}}, \quad (96)$$

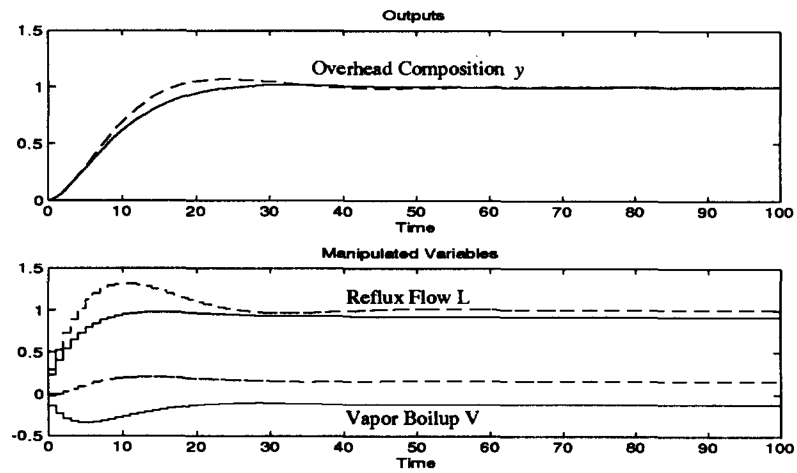
where $\tilde{h}_l = [\tilde{h}_{1,l} \; \tilde{h}_{2,l} \; \ldots \; \tilde{h}_{30,l}]^{\mathrm{T}}$ is a vector containing the impulse response coefficients for the lth input channel. $|h_l|$ is the 2-norm of h_l, which divides all the elements of h_l for normalization. Although the parameter vector θ is 60-dimensional, the ellipsoid Θ lies within a two-dimensional subspace, defined by two orthonormal vectors v_1 and v_2 (i.e. the first and second columns of V). The radius of the ellipsoid in the v_l direction is $1/(\gamma_l |h_l|)$.

We applied the SMMPC algorithm to the above system. Here we chose $\zeta = 1$. Other tuning parameters were chosen as follows: the prediction horizon $p = 30$, control move horizon $q = 10$ and input penalty $\lambda = 10$ for both inputs. Figure 6 shows the results obtained with various values of γ_1 and γ_2. Figure 6(a) gives the response of the inputs and the output when $\gamma_1 = \gamma_2 = 0.2828$. The actual plant error used for simulation was $(\delta_1, \delta_2) = (0.2, -0.2)$ which lies on the boundary of the ellipsoid. We observe that the output follows the setpoint change smoothly. Both inputs are utilized equally for control, because both have the same level of uncertainty. Figure 6(b) shows the response when $(\gamma_1, \gamma_2) = (0.2828, 0.4243)$, with the actual

(a)



(b)



(c)

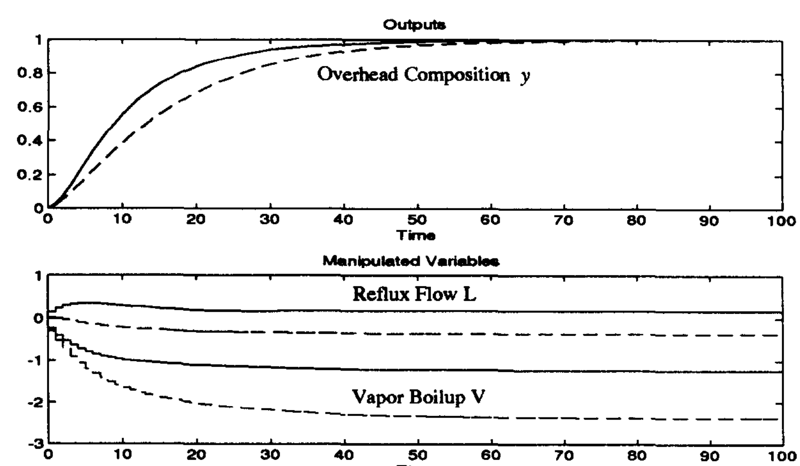


Fig. 6. Simulation results obtained with SMMPC algorithm for various sizes of the ellipsoid and correspondingly chosen input errors for Example 4. (a) $(\gamma_1, \gamma_2) = (0.2828, 0.2828)$, $(\delta_1, \delta_2) = (0.2, -0.2)$. (b) $(\gamma_1, \gamma_2) = (0.2828, 0.4243)$, $(\delta_1, \delta_2) = (0.2, -0.3)$; (----) $(\gamma_1, \gamma_2) = (0.2830, 1.2718)$, $(\delta_1, \delta_2) = (0.2, -0.9)$. (c) $(\gamma_1, \gamma_2) = (0.4243, 0.2828)$, $(\delta_1, \delta_2) = (0.3, -0.2)$; (----) $(\gamma_1, \gamma_2) = (1.2718, 0.2830)$, $(\delta_1, \delta_2) = (0.9, -0.2)$.

plant error given by $(\delta_1, \delta_2) = (0.2, -0.3)$. This time the first input is utilized much more because the controller knows that it has a lower level of uncertainty. This point becomes more dramatic when we choose $(\gamma_1, \gamma_2) = (0.2830, -1.2718)$ with the actual plant error given by $(\delta_1, \delta_2) = (0.2, -0.9)$, as shown in the same figure. Figure 6(c) shows an opposite trend when the uncertainty level for u_1 is increased instead.

4. CONCLUSION

In this paper we have presented different formulations of the 'worst-case' predictive control algorithms under a set membership uncertainty description and quadratic performance criterion. We have shown that, when the uncertainty is assumed to be time-invariant, the open-loop control assumption can lead to poor closed-loop performance and instability. The main reason for this is that the worst-case parameter can change from one optimization to another due to the feedback update (which is not accounted for in the open-loop formulation). We presented two alternatives. The first is a closed-loop min-max control formulation given as a dynamic program. The second is the open-loop formulation, but with uncertainty that is allowed to vary from one time step to next. We proved that both algorithms deliver robust stability under infinite prediction horizon. We also derived a computationally attractive suboptimal algorithm for a particular class of systems (i.e. FIR systems).

The conclusion drawn here for the min-max approach applies similarly to stochastic systems in which model parameters are random variables rather than bounded deterministic variables. The practice of treating parameters as random variables has been popular in the adaptive control field, as it tends to simplify the mathematics involved in comparison to the set-based approach. In this spirit, one could envisage the two approaches being merged. For example, the stochastic parameter model can be used to define a performance objective and additional constraints can be imposed, which guarantee closed-loop stability for all plants of probability higher than a certain level.

Acknowledgement—Support from the National Science Foundation (CST No. 9209808) is gratefully acknowledged.

REFERENCES

Belforte, G., Bona, B. and Cerone, V. (1990) Parameter estimation algorithms for a set-membership description of uncertainty. *Automatica* **26**, 887–898.

Bellman, R. (1961) *Adaptive Control Processes—A Guided Tour.* Princeton University Press.

Boyd, S. P. and Barratt, C. H. (1991) *Linear Control Design—Limits of Performance.* Prentice Hall, Englewood Cliffs, NJ.

Campo, P. J. and Morari, M. (1987) Robust model predictive control. In *Proc. Automatic Control Conf.*, Atlanta, GA, pp. 1021–1026.

Cutler, C. R. and Ramaker, B. L. (1980) Dynamic matrix control—a computer control algorithm. In *Proc. Automatic Control Conf.*, San Francisco, CA, Paper WP5-B.

Feldbaum, A. A. (1965) *Optimal Control Theory.* Academic Press, New York.

Fogel, E. and Huang, Y. F. (1982) On the value of information in system identification—bounded noise case. *Automatica* **18**, 229–238.

Garcia, C. E., Prett, D. M. and Morari, M. (1989) Model predictive control: theory and practice—a survey. *Automatica* **25**, 335–348.

Genceli, H. and Nikolaou, M. (1993) Robust stability analysis of constrained l_1 norm MPC. *AIChE J.* **39**, 1954–1965.

Goodwin, G., Ninness, C. B. and Salgado, M. E. (1990) Quantification of uncertainty in estimation. In *Proc. American Control Conf.*, San Diego, CA, pp. 2400–2405.

Keerthi, S. and Gilbert, E. (1988) Optimal infinite feedback laws for a general class of constrained discrete-time systems: stability and moving-horizon approximation. *J. Optim. Theory Applics*, **57**, 265–293.

Kosut, R. L., Lau, M. and Boyd, S. (1990) Identification of systems with parametric and nonparametric uncertainty. In *Proc. American Control Conf.*, San Diego, CA, pp. 2412–2417.

Lau, M. K., Boyd, S., Kosut, R. L. and Franklin, G. F. (1991) A robust control design for FIR plants with parameter set uncertainty. In *Proc. American Control Conf.*, Boston, MA, pp. 83–88.

Lee, J. H. and Cooley, B. (1995) Optimal feedback control strategies for state-space systems with stochastic parameters. *IEEE Trans. Autom. Control*, submitted.

Lee, J. H. and Cooley, B. (1997) Stable min-max control for state-space systems with bounded input matrix. In *Proc. American Control Conf.*, Alberquerque, NM.

Lee, J. H. and Yu, Z. H. (1994) Robust tuning parameters for model predictive control. *Comput. Chem. Engng*, **18**, 15–37.

Mayne, D. Q. and Michalska, H. (1990) Receding horizon control of nonlinear systems. *IEEE Trans. Autom. Control* **AC-35**, 714–724.

Milanese, M. and Belforte, G. (1982) Estimation theory and uncertainty intervals evaluation in presence of unknown but bounded errors. Linear families of models and estimators. *IEEE Trans. Autom. Control* **AC-30**, 730–738.

Rawlings, J. B. and Muske, K. R. (1993) The stability of constrained receding horizon control. *IEEE Trans. Autom. Control* **AC-38**, 1512–1516.

Skogestad, S., Morari, M. and Doyle, J. C. (1988) Robust control of ill-conditioned plants: high-purity distillation. *IEEE Trans. Autom. Control* **AC-33**, 1092–1105.

Veres, S. M. and Norton, J. P. (1993) Predictive self-tuning control by parameter bounding and worst-case design. *Automatica* **29**, 911–928.

Zheng, Z. Q. and Morari, M. (1993) Robust stability of constrained model predictive control. In *Proc. American Control Conf.*, San Francisco, CA, pp. 379–383.

Zheng, Z. Q. and Morari, M. (1994) Stability of model predictive control with soft constraints. In *Proc. 33rd IEEE Conf. on Decision and Control*, Lake Buena Vista, FL.

APPENDIX

Attractivity of FIR systems with an integrator under OLWOFC-II

Here we prove that, under the receding horizon control based on (71)–(73), input and output converge to zero for all possible parameter values under certain assumptions. We present the proof for the case of $q = 1$ only. For this case, only the primary optimization (71) is relevant.

First, we prove that the optimal cost for the primary optimization (71) converges.

Theorem A.1. For the system (58)–(60) with the constraints (63) and (64), the optimal cost for (71) with $q = 1$ converges as $k \to \infty$ under feedback control $v_k = u_k^*$, where u_k^* is the optimal value of u_k for (71)–(73).

Proof. Let

$$V_k = \min_{u_k} \left\{ J_k \triangleq \max_{[\theta_k,\ldots,\theta_{k+n-1}]\in\Theta\times\ldots\times\Theta} \tilde{y}^2_{k+n|k} \right\}, \qquad (A.1)$$

where

$$\tilde{y}_{k+n|k} = y_k + \sum_{j=0}^{n-1} \theta^{T}_{k+j} U_{k+j}. \qquad (A.2)$$

Let u_k^* be the minimizing value for u_k in the above optimization. Then,

$$V_k = \min_{u_k} \max_{[\theta_k,\ldots,\theta_{k+n-1}]\in\Theta\times\ldots\times\Theta} \{\tilde{y}^2_{k+n|k}\}, \tag{A.3}$$

$$\geq \max_{[\theta_{k+1},\ldots,\theta_{k+n-1}]\in\Theta\times\ldots\times\Theta} \{\tilde{y}^2_{k+n|k}\}, \tag{A.4}$$

$$\text{with } \theta_k = \vartheta_k \text{ and } u_k = u_k^*, \tag{A.5}$$

$$= \max_{[\theta_{k+1},\ldots,\theta_{k+n-1}]\in\Theta\times\ldots\times\Theta} \{\tilde{y}^2_{k+n|k+1}\} \tag{A.6}$$

$$= \max_{[\theta_{k+1},\ldots,\theta_{k+n}]\in\Theta\times\ldots\times\Theta} \{\tilde{y}^2_{k+n+1|k+1}\}, \text{ with } u_{k+1} = 0, \tag{A.7}$$

$$\geq \min_{u_{k+1}} \max_{[\theta_{k+1},\ldots,\theta_{k+n}]\in\Theta\times\ldots\times\Theta} \{\tilde{y}^2_{k+n+1|k+1}\}, \tag{A.8}$$

since $u_{k+1} = 0$ is feasible,

$$= V_{k+1} \tag{A.9}$$

Since $V_{k+1} - V_k \leq 0$ and V_i is bounded below by zero, it is a converging sequence. □

Next we show that the cost actually converges to zero under some mild assumptions. For this, we define

$$g_k^{ss} = \sum_{i=1}^{n} h_{i,k+i-1}. \tag{A.10}$$

Note that the feasible sequence g_k^{ss} is determined by $\vartheta_k \in \Theta, \ldots, \vartheta_{k+n-1} \in \Theta$. The assumptions we make are:

Assumption A.1. For all k, g_k^{ss} is nonzero and has the same sign for all feasible values of $h_{1,k}, \ldots, h_{n,k+n-1}$, as defined by $[\vartheta_k, \ldots, \vartheta_{k+n-1}] \in \Theta \times \ldots \times \Theta$. Hence, the feasible set is a closed interval that does not include the origin.

Assumption A.2. The initial condition is such that the constraint on integrated v_k of (64) remains inactive.

Theorem A.3. For system (58)–(60), under Assumptions A.1 and A.2, the optimal cost for (71) with $q = 1$ converges to zero and $y_k \to 0$ and $v_k \to 0$ as $k \to \infty$ under feedback control $v_k = u_k^*$.

Proof. A formal proof requires several lengthy and technical arguments. Instead, we present an informal proof, which captures the essence of the idea.

In the proof of Theorem A.1, we showed that V_k is a converging sequence and $V_{k+1} \leq J_{k+1}|_{u_{k+1}=0} \leq V_k$. Based on this, we first show that the optimal cost cannot converge to a value other than zero. For this, assume that the optimal cost remained at a constant value for at least n time steps (i.e. $V_{k-n+1} = \ldots = V_k$). Given the uniqueness of the optimal solution (which can be proven for the particular optimization), this implies that the optimal input remained zero for at least n time steps ($v_{k-n+1} = \ldots, v_k = 0$). Then,

$$\tilde{y}_{k+n+1|k+1} = y_{k+1} + \hat{g}^{ss}_{k+1} u_{k+1}, \tag{A.11}$$

where $\hat{g}^{ss}_{k+1}$ is a parameter that occupies the feasible set for g^{ss}_{k+1}. With $u_{k+1} = 0$, $J_{k+1} = y^2_{k+1}$. Suppose $y_{k+1} \neq 0$. Then,

$$\left.\frac{\mathrm{d}(\tilde{y}^2_{k+n+1|k+1})}{\mathrm{d}u_{k+1}}\right|_{u_{k+1}=0} = (2y_{k+1}\hat{g}^{ss}_{k+1} + 2(\hat{g}^{ss}_{k+1})^2 u_{k+1})_{u_{k+1}=0}$$
$$= 2y_{k+1}\hat{g}^{ss}_{k+1}. \tag{A.12}$$

The above derivative is nonzero and has the same sign for all feasible values of $\hat{g}^{ss}_{k+1}$ (Assumption A.1). In addition, the feasible set for u_{k+1} is a closed set that includes the origin as an interior point (Assumption A.2). This means that $u_{k+1} = 0$ is not an optimal solution and the cost can be lowered from y^2_{k+1}. In fact, there exists $0 < \epsilon \leq 1$ such that $V_{k+1} \leq (1-\epsilon)y^2_{k+1}$. This proves that the cost must converge to zero.

Next, we show that, once the optimal cost converges to zero, the input and output sequences must converge to zero. Suppose that $V_k = 0$. Then, $u_{k+1} = 0$ gives $J_{k+1} = V_k = 0$. Given the uniqueness of the optimal solution, $u_{k+1} = 0$ is the optimal solution for the optimization at $t = k+1$. By induction, the optimal input remains zero for all future time steps. In addition, the output must converge to zero at most in n steps. □

The proof for the general choice of q is somewhat lengthy and technical. The proof is complicated by the fact that the secondary optimization is performed whenever the optimal solution for primary optimization is not unique. In the case that only the steady-state error is minimized (either there exists no extra degree of freedom or only the inputs are minimized in the secondary optimization), the above proof extends straightforwardly.

Derivation of expressions of hessian and gradient for SMMPC algorithm

Equation (90) is more conveniently written as

$$\min_{\mathcal{U}_k} \sum_{l=1}^{p} \left\{ \left(y_k + \sum_{j=1}^{l} \tilde{\vartheta}^{\mathrm{T}} U_{k+j-1} - r_{k+l} \right)^2 + \lambda u^2_{k+l-1} \right.$$
$$+ \left((1-\zeta)(p-l+1) + \sum_{j=1}^{p-l+1} (j+l-1) \right)$$
$$\left. \times \|(W^{\mathrm{T}})^{-1} U_{k+l-1}\|_2^2 \right\}. \tag{A.13}$$

Note that $(W^{\mathrm{T}})^{-1} = [V \cdot \mathrm{diag}(1/\sigma_1, \ldots, 1/\sigma_n)]^{-1} = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)V^{\mathrm{T}}$. To facilitate the further exposition, we rewrite (A.13) using some new notation:

$$\min_{\mathcal{U}_k} \sum_{l=1}^{p} \left\{ \left[y_k - r_{k+l} + \tilde{\vartheta}^{\mathrm{T}} \left(\sum_{j=1}^{l} B_j^{\mathrm{past}} \right) V_{k-1} + \tilde{\vartheta}^{\mathrm{T}} \left(\sum_{j=1}^{l} B_j \right) \mathcal{U}_k \right]^2 \right.$$
$$+ \lambda u^2_{k+l-1} + \left[(1-\zeta)(p-l+1) + \sum_{j=1}^{p-l+1} (j+l-1) \right]$$
$$\left. \times \sum_{i=1}^{N} [\sigma_i v_i^{\mathrm{T}} (B_l^{\mathrm{past}} V_{k-1} + B_l \mathcal{U}_k)]^2 \right\}, \tag{A.14}$$

where $V_{k-1} = [v_{k-1}, \ldots, v_{k-N+1}]^{\mathrm{T}}$, and B_l^{past} and B_l are matrices chosen such that

$$U_{k+l-1} = B_l^{\mathrm{past}} V_{k-1} + B_l \mathcal{U}_k. \tag{A.15}$$

Let us also define

$$\mathcal{E}_k^y = \mathcal{Y}_k + \mathcal{H}^{\mathrm{past}} V_{k-1} + \mathcal{H}\mathcal{U}_k, \tag{A.16}$$

$$\mathcal{E}_k^u = \mathcal{M}^{\mathrm{past}} V_{k-1} + \mathcal{M}\mathcal{U}_k, \tag{A.17}$$

where

$$\mathcal{Y}_k = \begin{bmatrix} y_k \\ \vdots \\ y_k \end{bmatrix}, \quad \mathcal{H}^{\mathrm{past}} = \begin{bmatrix} \tilde{\vartheta}^{\mathrm{T}} B_1^{\mathrm{past}} \\ \tilde{\vartheta}^{\mathrm{T}} (B_1^{\mathrm{past}} + B_2^{\mathrm{past}}) \\ \vdots \\ \tilde{\vartheta}^{\mathrm{T}} (\sum_{l=1}^{p} B_l^{\mathrm{past}}) \end{bmatrix}$$

$$\mathcal{H} = \begin{bmatrix} \tilde{\vartheta}^{\mathrm{T}} B_1 \\ \tilde{\vartheta}^{\mathrm{T}} (B_1 + B_2) \\ \vdots \\ \tilde{\vartheta}^{\mathrm{T}} (\sum_{l=1}^{p} B_l) \end{bmatrix}_{p\times q},$$

$$\mathcal{M}^{\mathrm{past}}$$
$$= \begin{bmatrix} \sqrt{p(1-\zeta) + \zeta \sum_{j=1}^{p} j}\, (\sum_{i=1}^{n} \sigma_i v_i^{\mathrm{T}} B_1^{\mathrm{past}}) \\ \sqrt{(p-1)(1-\zeta) + \zeta \sum_{j=2}^{p} j}\, (\sum_{i=1}^{n} \sigma_i v_i^{\mathrm{T}} B_2^{\mathrm{past}}) \\ \vdots \\ \sqrt{(1-\zeta) + \zeta \sum_{j=p}^{p} j}\, (\sum_{i=1}^{n} \sigma_i v_i^{\mathrm{T}} B_p^{\mathrm{past}}) \end{bmatrix}_{p\times(n-1)},$$

$$\mathcal{M} = \begin{bmatrix} \sqrt{p(1-\zeta) + \zeta \sum_{j=1}^{p} j}\, (\sum_{i=1}^{n} \sigma_i v_i^{\mathrm{T}} B_1) \\ \sqrt{(p-1)(1-\zeta) + \zeta \sum_{j=2}^{p} j}\, (\sum_{i=1}^{n} \sigma_i v_i^{\mathrm{T}} B_2) \\ \vdots \\ (\sqrt{(1-\zeta) + \zeta \sum_{j=p}^{p} j}\, \sum_{i=1}^{n} \sigma_i v_i^{\mathrm{T}} B_p) \end{bmatrix}_{p\times q}. \tag{A.18}$$

Then, (A.14) can be simplified as

$$\min_{\mathcal{U}_k} \{(\mathcal{E}_k^y)^{\mathrm{T}} \mathcal{E}_k^y + (\mathcal{E}_k^u)^{\mathrm{T}} \mathcal{E}_k^u + \mathcal{U}_k^{\mathrm{T}} \Lambda \mathcal{U}_k\}, \tag{A.19}$$

where $\Lambda = \mathrm{diag}\{\lambda, \ldots, \lambda\}$. Multiplying out the above (after substituting (A.16) and (A.17)) and dropping the constant term, one obtains

$$\min_{\mathcal{U}_k} \mathcal{U}_k^{\mathrm{T}}\Big(\underbrace{\mathcal{H}^{\mathrm{T}}\mathcal{H} + \mathcal{M}^{\mathrm{T}}\mathcal{M} + \Lambda}_{\text{hessian}}\Big)\mathcal{U}_k + \Big(\underbrace{\mathcal{H}^{\mathrm{T}}\mathcal{Y}_k + (\mathcal{H}^{\mathrm{T}}\mathcal{H}^{\text{past}} + \mathcal{M}^{\mathrm{T}}\mathcal{M}^{\text{past}})V_{k-1}}_{\text{gradient}}\Big)^{\mathrm{T}}\mathcal{U}_k. \quad (A.20)$$

In the absence of any constraint on $\mathcal{U}_k$, the solution to the above minimization can be written analytically as

$$\mathcal{U}_k = -(\mathcal{H}^{\mathrm{T}}\mathcal{H} + \mathcal{M}^{\mathrm{T}}\mathcal{M} + \Lambda)^{-1} \times \{\mathcal{H}^{\mathrm{T}}\mathcal{Y}_k + (\mathcal{H}^{\mathrm{T}}\mathcal{H}^{\text{past}} + \mathcal{M}^{\mathrm{T}}\mathcal{M}^{\text{past}})V_{k-1}\}. \quad (A.21)$$

In the presence of linear inequality constraints on $\mathcal{U}_k$ (arising from input rate/magnitude limits), quadratic programming can be used to find the optimal solution.