

1

Linear Model Predictive Control via Multiparametric Programming

Vassilis Sakizlis, Konstantinos I. Kouramas, and Efstratios N. Pistikopoulos

1.1 Introduction

Linear systems with input, output, or state constraints are probably the most important class of systems in practice and the most studied as well. Although, a variety of control design methods have been developed for linear systems, it is widely accepted that stability and good performance for these systems, especially in the presence of constraints, is guaranteed with a nonlinear control law. The most popular nonlinear control approach for linear systems with constraints is model predictive control or simply MPC which has become the standard for constrained multivariable control problems in the process industries [24, 25, 35].

MPC is an online constrained optimization method, based on the so-called receding horizon philosophy [24, 25]. At each sampling instant, the measurements of the current output and/or state of the system are retrieved and an open-loop optimal control problem is solved over a finite time horizon to obtain the sequence of future control values. The first value of the sequence is then obtained and the procedure is repeated in the next sampling instant and over a shifted horizon, leading to a moving horizon policy. Since the objective function and the constraints of the open-loop optimal control problem can be set to express true performance objectives, the benefits of MPC are tremendous; optimality and constraints satisfaction being obviously its main advantage.

The application of MPC is, nevertheless, rather restricted, considering its profit potential, due to its online computational requirements which involve the repetitive solution of an optimization problem at regular time intervals. This limitation is in spite of the significant advances in the computational power of modern computers and in the area of online optimization over the past few years. Thus, it is fair to state that an efficient implementation of online optimization tools relies on a quick and repetitive online computation of optimal control actions. A way to avoid these repetitive online computations is by using *multiparametric programming techniques* to solve the optimization problem. With this approach, the control variables are obtained as an explicit function of the state variables and therefore the online

optimization breaks down to simple function evaluations, at regular time intervals, to compute the corresponding control actions for the given state of the plant. This is known as the *online optimization via off-line parametric optimization* concept.

1.1.1

Multiparametric Programming

Overall, multiparametric programming is a technique for solving any optimization problem, where the objective is to minimize or maximize a performance criterion subject to a given set of constraints and where some of the parameters vary between specified lower and upper bounds. The main characteristic of multiparametric programming is its ability to obtain [1, 4, 15, 19, 20, 26, 28]

- (i) the objective and optimization variable as functions of the varying parameters, and
- (ii) the regions in the space of the parameters where these functions are valid.

Multiparametric programming has been applied to a number of applications listed here

- (i) hybrid parametric/stochastic programming [2, 22],
- (ii) process planning under uncertainty [29],
- (iii) material design under uncertainty [13],
- (iv) multiobjective optimization [26, 27, 30],
- (v) exibility analysis [5],
- (vi) computation of singular multivariate normal probabilities [6], and
- (vii) model predictive control [9, 10, 31].

The advantage of using multiparametric programming to address these problems is that for problems pertaining to plant operations, such as for process planning, scheduling, and control, one can obtain a complete map of all the optimal solutions. Hence, as the operating conditions vary, one does not have to reoptimize for the new set of conditions, since the optimal solution is already available as a function of the operating conditions. Mathematically, all the above applications can be generally posed as a multiparametric mixed-integer nonlinear programming (mp-MINLP) problem

$$\begin{aligned}
 z(\gamma) &= \min_{\gamma, x} d^T \gamma + f(x) \\
 \text{s.t. } & E\gamma + g(x) \leq b + F, \\
 & \min \leq \leq \max, \\
 & x \in X \subset \mathbb{R}^n, \\
 & \gamma \in Y = \{0, 1\}^m, \\
 & \gamma \in \mathbb{R}^s,
 \end{aligned} \tag{1.1}$$

where γ is a vector of $\{0, 1\}$ binary variables, x is a vector of continuous variables, f a scalar, continuously differentiable and convex function of x , g a vector of con-

tinuous differentiable and convex functions of x , b and d are constant vectors, E and F are constant matrices, θ is a vector of parameters, u_{\min} and u_{\max} are the lower and upper bounds of u , and X and Y are compact and convex polyhedral sets of dimensions n and s , respectively.

The detailed theory and algorithms of solving the mp-MINLP problem (1.1) are exploited in [1, 14, 15]. Furthermore, a number of variations of the problem (1.1) have been developed and examined in [2, 14]. A special formulation of (1.1) which, as we will later see in this chapter, is of great importance for linear model predictive control problems is the multiparametric quadratic programming problem in which $d = 0$, $E = 0$, and f and g are quadratic and linear functions of x , respectively. This case will be given our full attention and will be treated carefully in the following sections of this chapter.

1.1.2

Model Predictive Control

Consider the general mathematical description of discrete-time, linear time-invariant systems

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t \end{aligned} \quad (1.2)$$

subject to the following constraints:

$$\begin{aligned} y_{\min} &\leq y_t \leq y_{\max}, \\ u_{\min} &\leq u_t \leq u_{\max}, \end{aligned} \quad (1.3)$$

where $x_t \in \mathbb{R}^n$, $u_t \in \mathbb{R}^m$, and $y_t \in \mathbb{R}^p$ are the state, input, and output vectors, respectively, subscripts min and max denote lower and upper bounds, respectively, and the matrix pair (A, B) is stabilizable.

The linear MPC problem for regulating (1.2) to the origin is posed as the following quadratic programming problem [24, 25, 35]

$$\begin{aligned} \min_U J(U, x_t) &= x'_{t+N_y|t} P x_{t+N_y|t} + \sum_{k=0}^{N_y-1} x'_{t+k|t} Q x_{t+k|t} + u'_{t+k} R u_{t+k} \\ \text{s.t. } y_{\min} &\leq y_{t+k|t} \leq y_{\max}, \quad k = 1, \dots, N_c, \\ u_{\min} &\leq u_{t+k} \leq u_{\max}, \quad k = 0, 1, \dots, N_c, \\ x_{t|t} &= x_t, \\ x_{t+k+1|t} &= Ax_{t+k|t} + Bu_{t+k}, \quad k \geq 0, \\ y_{t+k|t} &= Cx_{t+k|t}, \quad k \geq 0, \\ u_{t+k} &= Kx_{t+k|t}, \quad N_u \leq k \leq N_y, \end{aligned} \quad (1.4)$$

where $U \triangleq \{u_t, \dots, u_{t+N_u-1}\}$, $Q = Q' \geq 0$, $R = R' \geq 0$, $P \geq 0$, $(Q^{1/2}, A)$ is detectable, N_u, N_y, N_c are the input, output, and constraint horizon, respectively, such that $N_y \geq N_u$ and $N_c \leq N_y - 1$, and K is a stabilizing state feedback gain. Problem (1.4) is solved repetitively at each time t for the current measurement x_t and the vector of predicted state variables, $x_{t+1|t}, \dots, x_{t+k|t}$ at time $t+1, \dots, t+k$, respectively, and

corresponding optimal control actions $U^* = \{u_t^*, \dots, u_{t+k-1}^*\}$ is obtained. The input that is applied to the system is the first control action

$$u_t = u_t^*$$

and the procedure is repeated at time $t + 1$, based on the new state x_{t+1} .

The state feedback gain K and the terminal cost function matrix P usually are used to guarantee stability for the MPC (1.4). The stability problem of the MPC has been treated extensively (see also [12, 24, 32]). Since it is not in the scope of this chapter to expand on this issue, we will briefly present two methods to obtain K and P . One possible choice is to set $K = 0$ and P to be the solution of the discrete Lyapunov equation

$$P = A'PA + Q.$$

However, this solution is restricted only to open-loop stable systems, since the control action is stopped after N_u steps. Alternatively, one can choose K and P as the solutions of the unconstrained, infinite-horizon linear quadratic regulation (LQR) problem, i.e., when $N_c = N_u = N_y = \infty$,

$$\begin{aligned} K &= -(R + B'PB)^{-1}B'PA, \\ P &= (A + BK)'P(A + BK) + K'RK + Q. \end{aligned} \quad (1.5)$$

This is possibly the most popular method for obtaining the K and P matrices (see also [10, 32]).

Introducing the following relation, derived from (1.2),

$$x_{t+k|t} = A^k x_t + \sum_{j=0}^{k-1} A^j B u_{t+k-1-j} \quad (1.6)$$

in (1.4) results in the following quadratic programming or QP problem

$$\begin{aligned} J^*(x_t) &= \min_U \frac{1}{2} U' H U + x_t' F U + \frac{1}{2} x_t' Y x(t) \\ \text{s.t.} \quad & G U \leq W + E x_t, \end{aligned} \quad (1.7)$$

where $U \triangleq [u_t', \dots, u_{t+N_u-1}']' \in \mathbb{R}^s$, $s \triangleq mN_u$, is the vector of optimization variables, $H = H' \geq 0$, and H, F, Y, G, W, E are obtained from Q, R and (1.4)–(1.6). Thus, the MPC is applied by repetitively solving the QP problem (1.7) at each time $t \geq 0$ for the current value of the state x_t . Due to this formulation, the solution U^* of the QP is a function $U^*(x_t)$ of the state x_t , implicitly defined by (1.7) and the control action u_t is given by

$$u_t = [I \ 0 \ \dots \ 0] U^*(x_t). \quad (1.8)$$

The problem in (1.4) obviously describes the constrained linear quadratic regulation problem [11, 12, 32, 33], while (1.7) is the formulation of the MPC as a QP optimization problem. Despite the fact that efficient QP solvers are available to solve (1.7), computing the input u_t online may require significant computational effort. The solution of (1.4) via multiparametric programming means, which avoids the repetitive optimization, was first treated in [9, 10, 31] and will be discussed in the following sections of this chapter.

1.2

Multiparametric Quadratic Programming

Transforming the QP problem (1.7) into a multiparametric programming problem is easy, once the following linear transformation is considered:

$$z \triangleq U + H^{-1}F'x_t \quad (1.9)$$

$z \in \mathbb{R}^s$. The QP (1.7) is then formulated to the following multiparametric quadratic programming (mp-QP) problem:

$$\begin{aligned} V_z(x_t) = \min_z \quad & \frac{1}{2}z'Hz \\ \text{s.t.} \quad & Gz \leq W + Sx_t, \end{aligned} \quad (1.10)$$

where $z \in \mathbb{R}^s$ is the vector of optimization variable, x_t is the vector of parameters, and

$$S = E + GH^{-1}F'. \quad (1.11)$$

Note that in (1.7) the state vector x_t is present both in the objective function and the right-hand side (rhs) of the constraints, whereas in (1.10) it only appears on the rhs of the constraints. The main advantage of writing (1.4) in the form given in (1.10) is that z (and therefore U) can be obtained as an affine function of x for the complete feasible space of x .

In order to proceed with a method to solve (1.10), a number of important results have to be established. An important theorem is first recalled from [18] before we proceed.

Theorem 1 [18]. *Let $x_0 \in \mathbb{R}^n$ be a vector of parameters and (z_0, λ_0) be a KKT pair for problem (1.10), where $\lambda_0 = \lambda_0(x_0)$ is a vector of nonnegative Lagrange multipliers, λ , and $z_0 = z(x_0)$ is feasible in (1.10). Also assume that the (i) linear independence constraint satisfaction and (ii) strict complementary slackness conditions hold. Then, there exists in the neighborhood of x_0 a unique, once continuously differentiable function $[z(x), \lambda(x)]$ where $z(x)$ is a unique isolated minimizer for (1.10), and*

$$\frac{\frac{dz(x_0)}{dx}}{\frac{d\lambda(x_0)}{dx}} = -(M_0)^{-1}N_0, \quad (1.12)$$

where

$$M_0 = \begin{pmatrix} H & G_1^T & \cdots & G_q^T \\ -\lambda_1 G_1 & -V_1 & & \\ \vdots & & \ddots & \\ -\lambda_p G_p & & & -V_p \end{pmatrix} \quad (1.13)$$

$$N_0 = (Y, \lambda_1 S_1, \dots, \lambda_p S_p)^T, \quad (1.14)$$

where G_i denotes the i th row of G , S_i denotes the i th row of S , $V_i = G_i z_0 - W_i - S_i x_0$, W_i denotes the i th row of W , and Y is a null matrix of dimension $(s \times n)$.

The optimization variable $z(x)$ can then be obtained as an affine function of the state x_i by exploiting the first-order Karush–Kuhn Tucker (KKT) conditions for (1.10). More specifically

Theorem 2. *Let x be a vector of parameters and assume that assumptions (i) and (ii) of Theorem 1 hold. Then, the optimal z and the associated Lagrange multipliers λ are affine functions of x .*

PROOF. The first-order KKT conditions for the mp-QP (1.10) are given by

$$Hz + G'\lambda = 0, \quad (1.15)$$

$$\lambda_i(G_i z - W_i - S_i x) = 0, \quad i = 1, \dots, q, \quad (1.16)$$

$$\lambda \geq 0. \quad (1.17)$$

Recalling that H is invertible (1.15) is written as

$$z = -H^{-1}G'\lambda. \quad (1.18)$$

Let λ and λ denote the Lagrange multipliers corresponding to inactive and active constraints, respectively. For inactive constraints, $\lambda = 0$. For active constraints,

$$Gz - W - Sx = 0, \quad (1.19)$$

where G, W, S correspond to the set of active constraints. From (1.18) and (1.19),

$$\lambda = -GH^{-1}G'^{-1}(W + Sx). \quad (1.20)$$

Note that $(GH^{-1}G')^{-1}$ exists because of the LICQ assumption. Thus λ is an affine function of x . We can substitute λ from (1.20) into (1.18) to obtain

$$z = H^{-1}G'GH^{-1}G'^{-1}(W + Sx) \quad (1.21)$$

and note that z is also an affine function of x .

An interesting observation, resulting from Theorems 1 and 2, is given in the next corollary.

Corollary 1. *From Theorems 1 and 2,*

$$\begin{bmatrix} z(x) \\ \lambda(x) \end{bmatrix} = -(M_0)^{-1}N_0(x - x_0) + \begin{bmatrix} z_0 \\ \lambda_0 \end{bmatrix}. \quad (1.22)$$

The results in Theorems 1 and 2 and Corollary 1 are summarized in the following theorem (see also in [35]).

Theorem 3. *For the problem in (1.10) let x_0 be a vector of parameter values and (z_0, λ_0) a KKT pair, where $\lambda_0 = \lambda(x_0)$ is a vector of nonnegative Lagrange multipliers, λ , and $z_0 = z(x_0)$ is feasible in (1.10). Also assume that (i) linear independence constraint qualification and (ii) strict complementary slackness conditions hold. Then,*

$$\begin{bmatrix} z(x) \\ \lambda(x) \end{bmatrix} = -(M_0)^{-1}N_0(x - x_0) + \begin{bmatrix} z_0 \\ \lambda_0 \end{bmatrix}, \quad (1.23)$$

where

$$M_0 = \begin{pmatrix} H & G_1^T & \cdots & G_q^T \\ -\lambda_1 G_1 & -V_1 & & \\ \vdots & & \ddots & \\ -\lambda_p G_p & & & -V_p \end{pmatrix} \quad (1.24)$$

$$N_0 = (Y, \lambda_1 S_1, \dots, \lambda_p S_p)^T, \quad (1.25)$$

where G_i denotes the i th row of G , S_i denotes the i th row of S , $V_i = G_i z_0 - W_i - S_i x_0$, W_i denotes the i th row of W , and Y is a null matrix of dimension $(s \times n)$.

Theorems 2 and 3 show that given the solution z_0, λ_0 for a specific vector of parameters x_0 , one can obtain the solution $z(x), \lambda(x)$ for any parameter vector x from (1.23). Thus, the optimization variable z and eventually the control sequence U are linear, affine functions of the state x , $z(x)$ and $U(x)$. In that way the sequence of control actions is obtained as an *explicit function* of the state. It remains now to establish for which values of the parameter (state) x , this solution (1.23) remains optimal.

The set of x where solution (1.23) remains optimal is defined as the critical region (CR^0) and can be obtained as follows. Let CR^R represent the set of inequalities obtained (i) by substituting $z(x)$ into the inactive constraints in (1.10), and (ii) from the positivity of the Lagrange multipliers corresponding to the active constraints, as follows:

$$CR^R = \{Gz(x) \leq W + Sx(x), \lambda(x) \geq 0\}, \quad (1.26)$$

then CR^0 is obtained by removing the redundant constraints from CR^R as follows:

$$CR^0 = \{CR^R\}, \quad (1.27)$$

where $\{ \}$ is an operator which removes the redundant constraints for a procedure to identify the redundant constraints, see [20]. Since for a given space of state variables, X , so far we have characterized only a subset of X , i.e., $CR^0 \subset X$, in the next step the rest of the region CR^{rest} , is obtained as follows:

$$CR^{\text{rest}} = X - CR^0, \quad (1.28)$$

by using a procedure which is going to be described in what follows (see also [14]). The above steps, (1.23)–(1.28), are repeated and a set of $z(x), \lambda(x)$ and corresponding CR^0 's is obtained. The solution procedure terminates when no more regions can be obtained, i.e., when $CR^{\text{rest}} = \emptyset$. For the regions which have the same solution and can be unified to give a convex region, such a unification is performed and a compact representation is obtained.

The continuity and convexity properties of the optimal solution are summarized in the next theorem.

Theorem 4. For the mp-QP problem (1.10), the set of feasible parameters $X_f \subset X$ is convex, the optimal solution, $z(x): X_f \rightarrow \mathbb{R}^s$ is continuous and piecewise affine, and

the optimal objective function $V_z(x): X_f \rightarrow \mathbb{R}$ is continuous, convex, and piecewise quadratic.

PROOF. We first prove convexity of X_f and $V_z(x)$. Take generic $x_1, x_2 \in X_f$, and let $V_z(x_1), V_z(x_2)$, and z_1, z_2 be the corresponding optimal values and minimizers. Let $\alpha \in [0, 1]$, and define $z \triangleq \alpha z_1 + (1 - \alpha)z_2$, $x \triangleq \alpha x_1 + (1 - \alpha)x_2$. By feasibility, z_1, z_2 satisfy the constraints $Gz_1 \leq W + Sx_1$, $Gz_2 \leq W + Sx_2$. These inequalities can be linearly combined to obtain $Gz \leq W + Sx$, and therefore z is feasible for the optimization problem (1.10) where $x_t = x$. Since a feasible solution $z(x)$ exists at x , an optimal solution exists at x and hence X_f is convex. The optimal solution at x will be less than or equal to the feasible solution, i.e.,

$$V_z(x) \leq \frac{1}{2} z' H z$$

and hence

$$\begin{aligned} V_z(x) &= \frac{1}{2} [\alpha z_1' H z_1 + (1 - \alpha) z_2' H z_2] \\ &\leq \frac{1}{2} z' H z - \frac{1}{2} [\alpha z_1' H z_1 + (1 - \alpha) z_2' H z_2] \\ &= \frac{1}{2} [\alpha^2 z_1' H z_1 + (1 - \alpha)^2 z_2' H z_2 + 2 \alpha (1 - \alpha) z_2' H z_1 - \alpha z_1' H z_1 - (1 - \alpha) z_2' H z_2] \\ &= -\frac{1}{2} \alpha (1 - \alpha) (z_1 - z_2)' H (z_1 - z_2) \leq 0, \end{aligned} \quad (1.29)$$

i.e.,

$$V_z(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha V_z(x_1) + (1 - \alpha)V_z(x_2)$$

$x_1, x_2 \in X$, $\alpha \in [0, 1]$, which proves the convexity of $V_z(x)$ on X_f . Within the closed polyhedral regions CR^0 in X_f the solution $z(x)$ is affine (1.21). The boundary between two regions belongs to both closed regions. Because the optimum is unique the solution must be continuous across the boundary. The fact that $V_z(x)$ is continuous and piecewise quadratic follows trivially.

An algorithm for the solution of an mp-QP of the form given in (1.10) to calculate U as an affine function of x and characterize X by a set of polyhedral regions, CRs, is summarized in Table 1.1. The optimal control sequence $U^*(x)$, once $z(x)$ is obtained by (1.23), is obtained from (1.9)

$$U^*(x) = z(x) - H^{-1} F' x. \quad (1.30)$$

Finally, the feedback control law

$$u_t = [I \quad 0 \quad \cdots \quad 0] U^*(x_t) \quad (1.31)$$

is applied to the system. The algorithm in Table 1.1 obtains the regions CR^0 s in the subset X of the state space, where the optimal solution (1.23) and hence (1.31) exist. Hence, the MPC can be implemented by performing online the following heuristic rule:

Table 1.1 mp-QP algorithm.

Step 1	For a given space of x solve (1.10) by treating x as a free variable and obtain $[x_0]$.
Step 2	In (1.10) fix $x = x_0$ and solve (1.10) to obtain $[z_0, \lambda_0]$.
Step 3	Obtain $[z(x), \lambda(x)]$ from (1.23).
Step 4	Define CR^R as given in (1.26).
Step 5	From CR^R remove redundant inequalities and define the region of optimality CR^0 as given in (1.27).
Step 6	Define the rest of the region, CR^{rest} , as given in (1.28).
Step 7	If no more regions to explore, go to the next step, otherwise go to Step 1.
Step 8	Collect all the solutions and unify a convex combination of the regions having the same solution to obtain a compact representation.

1. obtain the measurements of the state x at the current time;
2. obtain the region CR^0 in which x belongs, i.e., $x \in CR^0$;
3. obtain the control action u_t from (1.31) and apply it to the system;
4. repeat in the next sampling instant.

Thus, the online optimization of the MPC problem (1.4) is reduced to the following simple function evaluation scheme:

$$\text{if } x \in CR^0 \text{ then } u_t = [I \quad 0 \quad \cdots \quad 0] U^*(x_t).$$

Remark 1. Note that the union of all regions CR^0 forms the space of feasible parameters for the mp-QP problem (1.10) which is also the space of feasible initial conditions for which a control exists that solves the MPC problem (1.4).

This approach provides a significant advancement in the solution and online implementation of MPC problems, since its application results in a complete set of control actions as a function of state variables (from (1.23)) and the corresponding regions of validity (from (1.27)), which are computed off-line, i.e., the explicit control law. Therefore during online optimization, no optimizer call is required and instead the region CR^0 for the current state of the plant where the value of the state variables is valid, can be identified by substituting the value of these state variables into the inequalities which define the regions. Then, the corresponding control actions can be computed by using a function evaluation of the corresponding affine function.

1.2.1

Definition of CR^{rest}

This section describes a procedure for calculating the rest CR^{rest} of a subset X , when a region of optimality $CR^0 \subset X$ is given, i.e., $CR^{\text{rest}} = X - CR^0$. For the sake of simplifying the explanation of the procedure, consider the case when only

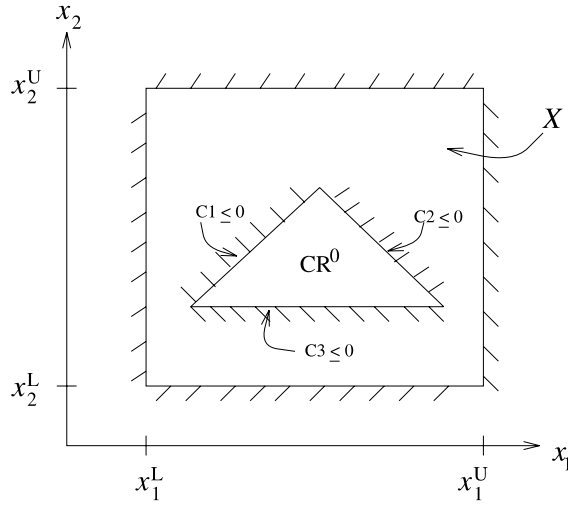


Fig. 1.1 Critical regions, X and CR^0 .

two state-variables x_1 and x_2 , are present (see Fig. 1.1), and X is defined by the inequalities:

$$X \triangleq \{x \in \mathbb{R}^n \mid x_1^L \leq x_1 \leq x_1^U, x_2^L \leq x_2 \leq x_2^U\}$$

and CR^0 is defined by the inequalities:

$$CR^0 \triangleq \{x \in \mathbb{R}^n \mid C1 \leq 0, C2 \leq 0, C3 \leq 0\},$$

where $C1$, $C2$, and $C3$ represent linear functions of x . The procedure consists of considering one by one the inequalities which define CR^0 . Considering, for example, the inequality $C1 \leq 0$, the rest of the region is given by

$$CR_1^{\text{rest}} = \{x \in \mathbb{R}^n \mid C1 \geq 0, x_1^L \leq x_1, x_2 \leq x_2^U\}$$

which is obtained by reversing the sign of inequality $C1 \leq 0$ and removing redundant constraints in X (see Fig. 1.2). Thus, by considering the rest of the inequalities, the complete rest of the region is given by

$$CR^{\text{rest}} = \{CR_1^{\text{rest}} \cup CR_2^{\text{rest}} \cup CR_3^{\text{rest}}\},$$

where CR_1^{rest} , CR_2^{rest} , and CR_3^{rest} are given in Table 1.2 and are graphically depicted in Fig. 1.3. Note that for the case when X is unbounded, simply suppress the inequalities involving X in Table 1.2.

Table 1.2 Definition of rest of the regions.

Region	Inequalities				
CR_1^{rest}	$C1 \geq 0,$	$x_1^L \leq x_1,$	$x_2 \leq x_2^U$		
CR_2^{rest}	$C1 \leq 0,$	$C2 \geq 0,$	$x_1 \leq x_1^U,$	$x_2 \leq x_2^U$	
CR_3^{rest}	$C1 \leq 0,$	$C2 \leq 0,$	$C3 \geq 0,$	$x_1^L \leq x_1 \leq x_1^U,$	$x_2^L \leq x_2$

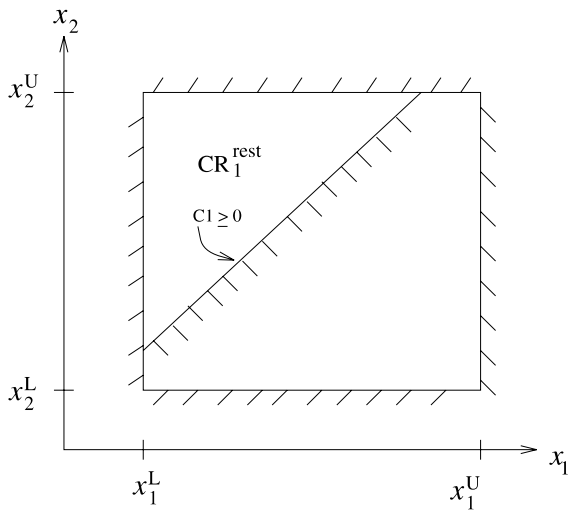


Fig. 1.2 Reversing the sign of C1 inequality to obtain CR_1^{rest} .

1.3 Numerical Example

Consider the following state-space representation:

$$\begin{cases} x_{t+1} = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x_t + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u_t, \\ y_t = \begin{bmatrix} 0 & 1.4142 \end{bmatrix} x_t. \end{cases} \quad (1.32)$$

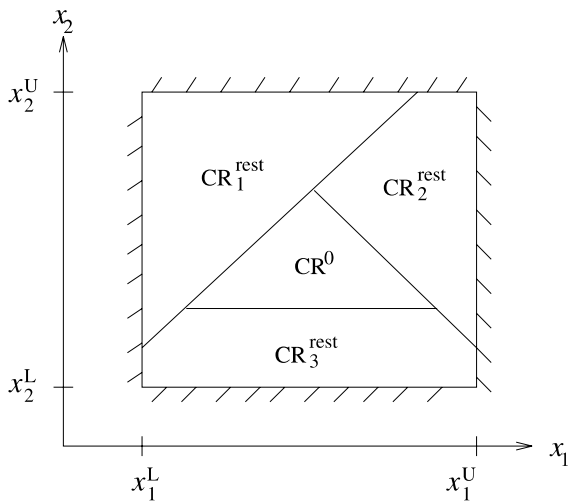


Fig. 1.3 Rest of the regions.

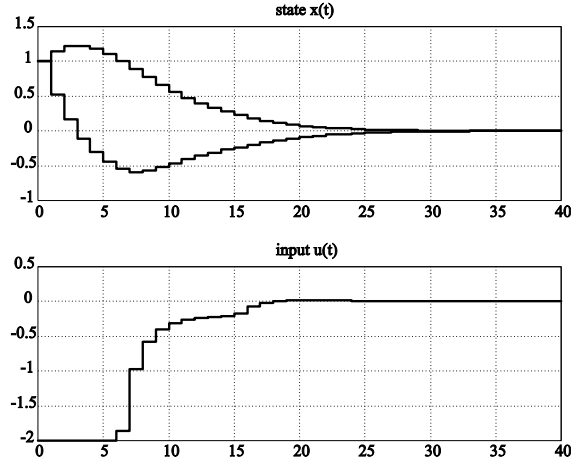


Fig. 1.4 Closed-loop response.

The constraints on input are as follows:

$$-2 \leq u_t \leq 2. \quad (1.33)$$

The corresponding optimization problem of the form (1.4) for regulating to the origin is given as follows:

$$\begin{aligned} \min_{u_t, u_{t+1}} & x'_{t+2|t} P x_{t+2|t} + \sum_{k=0}^1 [x'_{t+k|t} x_{t+k|t} + 0.01 u_{t+k}^2] \\ \text{s.t.} & -2 \leq u_{t+k} \leq 2, \quad k = 0, 1 \\ & x_{i|t} = x_t, \end{aligned} \quad (1.34)$$

where P solves the Lyapunov equation $P = A'PA + Q$,

$$Q = \begin{bmatrix} 10 & \\ 0 & 1 \end{bmatrix}, \quad R = 0.01, \quad N_u = N_y = N_c = 2.$$

The closed-loop response from the initial condition $x_0 = [1 \ 1]'$ is shown in Fig. 1.4. The same problem is now solved by using the parametric programming approach. The corresponding mp-QP problem of the form (1.10) has the following constant vectors and matrices:

$$\begin{aligned} H &= \begin{bmatrix} 0.0196 & 0.0063 \\ 0.0063 & 0.0199 \end{bmatrix}, & F &= \begin{bmatrix} 0.1259 & 0.0679 \\ 0.0922 & -0.0924 \end{bmatrix} \\ G &= \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix}, & W &= \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}, & E &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ y_0 & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

The solution of the mp-QP problem, as computed by using the algorithm given in Table 1.1, is given in Table 1.3 and is depicted in Fig. 1.5. Note that the CRs 2, 4 and 7, 8 in Table 1.3 are combined together and a compact convex representation is obtained. To illustrate how online optimization reduces to a function evaluation

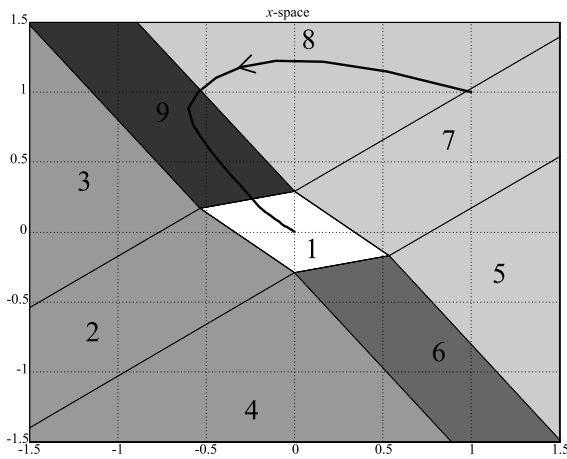


Fig. 1.5 Polyhedral partition of the state space.

Table 1.3 Parametric solution of the numerical example.

Region no.	Region	u
1	-5 9220 -6 8883 2 0000	$[-5.9220 \quad -6.8883]x$
	5 9220 6 8883 2 0000	
	-1 5379 6 8291 2 0000	
	1 5379 -6 8291 2 0000	
2,4	-3 4155 4 6452 2 6341	2.0000
	0 1044 0 1215 $x \leq -0 \quad 0353$	
	0 1259 0 0922 -0 0267	
3	$\begin{bmatrix} 0 & 0679 & -0 & 0924 \\ 0 & 1259 & 0 & 0922 \end{bmatrix} x \leq \begin{bmatrix} -0 & 0524 \\ -0 & 0519 \end{bmatrix}$	2.0000
5	$\begin{bmatrix} -0 & 1259 & -0 & 0922 \\ -0 & 0679 & 0 & 0924 \end{bmatrix} x \leq \begin{bmatrix} -0 & 0524 \\ -0 & 0524 \end{bmatrix}$	-2.0000
6	-6 4159 -4 6953 1 3577	$[-6.4159 \quad -4.6953]x + 0.6423$
	-0 0275 0 1220 $x \leq -0 \quad 0357$	
	6 4159 4 6953 2 6423	
7,8	3 4155 -4 6452 2 6341	-2.0000
	-0 1044 -0 1215 $x \leq -0 \quad 0353$	
	-0 1259 -0 0922 -0 0267	
9	6 4159 4 6953 1 3577	$[-6.4159 \quad -4.6953]x - 0.6423$
	0 0275 -0 1220 $x \leq -0 \quad 0357$	
	-6 4159 -4 6953 2 6423	

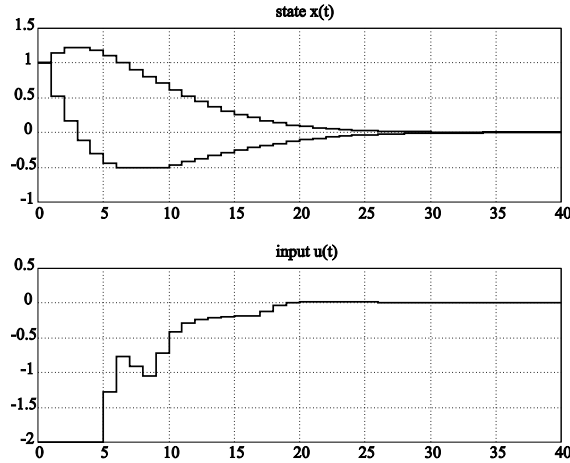


Fig. 1.6 Closed-loop response with additional constraint $x_{t+k|t} \geq -0.5$.

problem, consider the starting point $x_0 = [1 \ 1]'$. This point is substituted into the constraints defining the CRs in Table 1.3 and it satisfies only the constraints of CR^{7,8} (see also Fig. 1.5). The control action corresponding to CR^{7,8} from Table 1.3 is $u^{7,8} = -2$, which is obtained without any further optimization calculations and it is same as the one obtained from the closed-loop response depicted in Fig. 1.4.

The same example is repeated with the additional constraint on the state

$$x_{t+k|t} \geq x_{\min}, \quad x_{\min} \triangleq \begin{bmatrix} -0.5 \\ -0.5 \end{bmatrix}, \quad k = 1.$$

The closed-loop behavior from the initial condition $x_0 = [1 \ 1]'$ is presented in Fig. 1.6. The MPC controller is given in Table 1.4. The polyhedral partition of the state-space corresponding to the modified MPC controller is shown in Fig. 1.7. The partition consists now of 11 regions. Note that there are feasible states smaller than x_{\min} , and vice versa, infeasible states $x \geq x_{\min}$. This is not surprising. For instance, the initial state $x_0 = [-0.6 \ 0]'$ is feasible for the MPC controller (which checks state constraints at time $t + k$, $k = 1$), because there exists a feasible input such that x_1 is within the limits. In contrast, for $x_0 = [-0.47 \ -0.47]'$ no feasible input is able to produce a feasible x_1 . Moreover, the union of the regions depicted in Fig. 1.7 should not be confused with the region of attraction of the MPC closed loop. For instance, by starting at $x_0 = [46.0829 \ -7.0175]'$ (for which a feasible solution exists), the MPC controller runs into infeasibility after $t = 9$ time steps.

1.4 Computational Complexity

The algorithm given in Table 1.1 solves an mp-QP by partitioning X in N_r convex polyhedral regions. This number N_r depends on the dimension n of the state, the

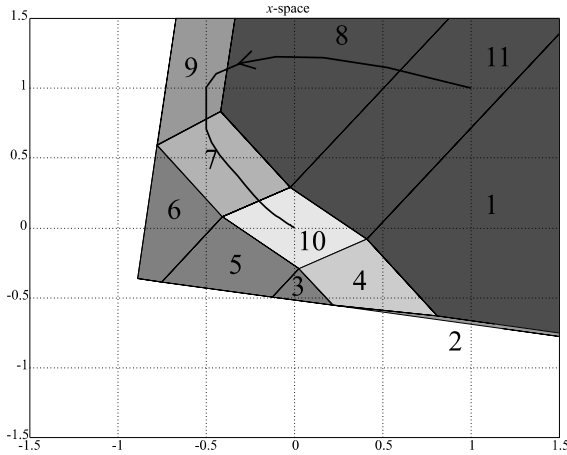


Fig. 1.7 Polyhedral partition of the state space with additional constraint $x_{t+k|t} \geq -0.5$.

product $s = mN_u$ of the number N_u of control moves and the dimension m of the input vector, and the number of constraints q in the optimization problem (1.10).

In an LP the optimum is reached at a vertex, and therefore s constraints must be active. In a QP the optimizer can lie everywhere in the admissible set. As the number of combinations of ℓ constraints out of a set of q is

$$\binom{q}{\ell} = \frac{q!}{(q-\ell)!\ell!} \quad (1.35)$$

the number of possible combinations of active constraints at the solution of a QP is at most

$$\sum_{\ell=0}^q \binom{q}{\ell} = 2^q. \quad (1.36)$$

This number represents an upper bound on the number of different linear feedback gains which describe the controller. In practice, far fewer combinations are usually generated as x spans X . Furthermore, the gains for the future input moves $u_{t+1}, \dots, u_{t+N_u-1}$ are not relevant for the control law. Thus several different combinations of active constraints may lead to the same first m components $u_t^*(x)$ of the solution. On the other hand, the number N_r of regions of the piecewise affine solution is in general larger than the number of feedback gains, because the regions have to be convex sets.

A worst case estimate of N_r can be computed from the way the algorithm in Table 1.1 generates critical regions CR to explore the set of parameters X . The following analysis does not take into account (i) the reduction of redundant constraints, and (ii) possible empty sets are not further partitioned. The first critical region CR_0 is defined by the constraints $\lambda(x) \geq 0$ (q constraints) and $Gz(x) \leq W + Sx$ (q constraints). If the strict complementary slackness condition holds, only q constraints can be active, and hence CR is defined by q constraints. From Section 1.2.1, CR^{rest} consists of q convex polyhedra CR_i , defined by at most q inequalities. For each CR_i ,

Table 1.4 Parametric solution of the numerical example for $x_{t+k|t} \geq -0.5$.

Region no.	Region			u
1, 2	3.4155	-4.6452	2.6341	-2.0000
	-0.1044	-0.1215	-0.0353	
	-0.7326	0.0861	0.3782	
	-0.1259	-0.0922	-0.0267	
3, 9	-3.4155	4.6452	2.6341	2.0000
	-0.7326	0.0861	0.6218	
	0.1044	0.1215	-0.0353	
	0.1259	0.0922	-0.0267	
	-0.1722	-0.9909	0.5128	
4	-0.7326	0.0861	0.6218	2.0000
	0.1259	0.0922	-0.0519	
	0.0679	-0.0924	-0.0524	
5	-12.0326	1.4142	10.2120	[-12.0326 1.4142]x -8.2120
	12.0326	-1.4142	-6.2120	
	1.8109	-1.9698	-2.4406	
6	-6.4159	-4.6953	2.6423	[-6.4159 -4.6953]x -0.6423
	-0.3420	0.3720	0.4609	
	6.4159	4.6953	1.3577	
	0.0275	-0.1220	-0.0357	
7	-1.5379	6.8291	2.0000	[-5.9220 -6.8883]x
	-5.9220	-6.8883	2.0000	
	1.5379	-6.8291	2.0000	
	5.9220	6.8883	2.0000	
8	-0.1722	-0.9909	0.4872	-2.0000
	-0.1259	-0.0922	-0.0519	
	-0.0679	0.0924	-0.0524	
10	-0.1311	-0.9609	0.5041	[-6.4159 -4.6953]x +0.6423
	6.4159	4.6953	2.6423	
	-0.0275	0.1220	-0.0357	
	-6.4159	-4.6953	1.3577	
11	-26.8936	-154.7504	80.0823	[-26.8936 -154.7504]x -78.0823
	26.8936	154.7504	-76.0823	
	62.7762	460.0064	-241.3367	

a new CR is determined which consists of $2q$ inequalities (the additional q inequalities come from the condition $\text{CR} \subseteq \text{CR}_i$), and therefore the corresponding CR^{rest} partition includes $2q$ sets defined by $2q$ inequalities. As mentioned above, this way of generating regions can be associated with a search tree. By induction, it is easy to prove that at the tree level $k+1$ there are $k!m^k$ regions defined by $(k+1)q$ constraints. As observed earlier, each CR is the largest set corresponding to a certain combina-

Table 1.5 Computational time to solve the mp-QP problem (seconds).

Free moves	States			
	2	3	4	5
2	3.02	4.12	5.05	5.33
3	10.44	26.75	31.7	70.19
4	25.27	60.20	53.93	58.61

tion of active constraints. Therefore, the search tree has a maximum depth of 2^q , as at each level there is one admissible combination less. In conclusion, the number of regions is

$$N_r \leq \sum_{k=0}^{2^q-1} k!q^k$$

each one defined by at most $q2^q$ linear inequalities.

1.4.1

Computational Time

In Tables 1.5 and 1.6 we report the computational time and the number of regions obtained by solving a few test MPC problems. In the comparison, we vary the number of free moves N_u and the number of poles of the open-loop system (and consequently the number of states x). Computational times have been evaluated by running the algorithm in Table 1.1 in Matlab 5.3 on a Pentium II-300 MHz machine. No attempts were made to optimize the efficiency of the algorithm and its implementation.

The theory and algorithm presented in this work are quite general and seem to have great potential for large scale, industrial applications. While the framework presented in this work may still require significant computational effort, most computations are executed off-line, while online implementation basically reduces to simple function evaluations. The suitability and applicability of the proposed parametric optimization based approach to large scale applications is a topic of current investigation.

Table 1.6 Number of regions N_r in the MPC solution.

Free moves	States			
	2	3	4	5
2	7	7	7	7
3	17	47	29	43
4	29	99	121	127

1.5

Extensions to the Basic MPC Problem

The basic MPC problem (1.4) and the corresponding mp-QP formulation (1.10), despite the fact that they treat the constrained regulation problem, can be extended to treat other important control problems. To present the capability of multiparametric programming to deal with a number of control problems, some of these extensions are presented here.

1.5.1

Reference Tracking

In reference tracking problems the objective is that the output, instead of being regulated to the origin, is required to either asymptotically converge to a constant reference value or follow a reference signal that may vary in time. In either case, the objective is to minimize the error between the system output y_t and the reference signal $r_t \in \mathbb{R}^p$, which is given by the problem specifications and is a function of time.

The general MPC formulation to treat the reference tracking problem is

$$\begin{aligned} \min_U J(U, x_t, \mathbf{r}_t) &= \sum_{k=0}^{N_y-1} y_{t+k|t} - r_t \quad ' Q \quad y_{t+k|t} - r_{t+k} + u'_{t+k} R u_{t+k} \\ \text{s.t. } y_{\min} &\leq y_{t+k|t} \leq y_{\max}, \quad k = 1, \dots, N_c \\ u_{\min} &\leq u_{t+k} \leq u_{\max}, \quad k = 0, 1, \dots, N_c \\ u_{\min} &\leq u_{t+k} \leq u_{\max}, \quad k = 0, 1, \dots, N_u - 1 \\ x_{t+k+1|t} &= Ax_{t+k|t} + Bu_{t+k}, \quad k \geq 0 \\ y_{t+k|t} &= Cx_{t+k|t}, \quad k \geq 0 \\ u_{t+k} &= u_{t+k-1} + u_{t+k}, \quad k \geq 0 \\ u_{t+k} &= 0, \quad k \geq N_u, \end{aligned} \quad (1.37)$$

where $U \triangleq \{u_t, \dots, u_{t+N_u-1}\}$, $\mathbf{r}_t \triangleq \{r_t, \dots, r_{t+N_y-1}\}$, and $u \in \mathbb{R}^m$ represent the control increments that act as correcting terms in the input to force the output to track the reference signal. The equation

$$u_{t+k} = u_{t+k-1} + u_{t+k}$$

corresponds to adding an integrator in the control loop. Due to this formulation, the past input u_{t-1} is introduced in the above problem as a new vector of m parameters.

The reference tracking MPC (1.37) can be formulated into an mp-QP problem, just like the regulation problem (1.4), by using the same procedure described in (1.1) and (1.2). One should note, though, that the number of parameters in this case has increased since, except for the state x , the past input u_{t-1} and the reference inputs \mathbf{r}_t were also introduced. Note, that if the reference is constant then $r_t = \dots = r_{t+N_y-1} = r$ and only one parameter, namely r , is introduced. By taking these extra parameters into account and repeating the procedure in Sections 1.1 and 1.2, we can transform the tracking problem (1.37) into

$$\begin{aligned}
J^*(x_t, u_{t-1}, \mathbf{r}_t) &= \min_U \frac{1}{2} U' H U + [x_t' u_{t-1}' \mathbf{r}_t'] F U \\
\text{s.t. } GU &\leq W + E \begin{matrix} x_t \\ u_{t-1} \\ \mathbf{r}_t \end{matrix}
\end{aligned} \tag{1.38}$$

and finally into the mp-QP problem

$$\begin{aligned}
V_z(x_t) &= \min_z \frac{1}{2} z' H z \\
\text{s.t. } Gz &\leq W + S \begin{matrix} x_t \\ u_{t-1} \\ \mathbf{r}_t \end{matrix} ,
\end{aligned} \tag{1.39}$$

where

$$z = U + H^{-1} F' \begin{matrix} x_t \\ u_{t-1} \\ \mathbf{r}_t \end{matrix} \quad \text{and} \quad S = E + G H^{-1} F'.$$

The mp-QP algorithm (1.1) can then be used to solve (1.39). The solution of (1.39) U is a linear, piecewise affine function $U(x_t, u_{t-1}, \mathbf{r}_t)$ of $x_t, u_{t-1}, \mathbf{r}_t$ defined over a number of regions CR^0 where this solution is valid. The reference tracking MPC is implemented by applying the following control:

$$u_t = u_{t-1} + u_t(x_t, u_{t-1}, \mathbf{r}_t),$$

where $u_t(x_t, u_{t-1}, \mathbf{r}_t)$ is the first component of the vector $U(x_t, u_{t-1}, \mathbf{r}_t)$.

1.5.2

Relaxation of Constraints

It is rather inevitable in some applications to consider possible violation of the output and input constraints, which can lead to infeasible solution of the problem (1.4) [36]. It is a common practice then to relax some of the output constraints in order to guarantee that a feasible solution is obtained for (1.4) and that the input constraints are satisfied, since usually these constraints are related to safety and performance issues. Therefore, the output constraints have to be posed again as

$$\gamma_{\min} - \leq \gamma_t \leq \gamma_{\max} - ,$$

where the scalar variable corresponds to the magnitude of the relaxation and the vector $\in \mathbb{R}^p$ is constant and is used to determine the extent to which each of the constraints is relaxed. In order to penalize the constraints violation the quadratic term 2 is added to the objective function and the relaxation variable is treated as an optimization variable. Therefore, solving the mp-QP problem (1.10), with the extra optimization variable, is obtained as a piecewise affine function of the state which allows one to know the exact violation of the constraints for each value of the system states.

1.5.3

The Constrained Linear Quadratic Regulator Problem

It is easy to observe that the MPC problem (1.4) describes the constrained linear quadratic (CLQR) problem [12, 32], when $N_c = N_u = N_y = N$ and the state feedback gain K and P are obtained from the solution of the unconstrained, infinite-horizon LQR problem (1.5). It is easy to note then that the solution to the CLQR, by following the mp-QP method described in Section 1.2, is obtained as a piecewise affine function of the states.

1.6

Conclusions

The solution of the linear MPC optimization problem, with a quadratic objective and linear output and input constraints, by using multiparametric programming techniques and specifically multiparametric quadratic programming, provides a complete map of the optimal control as a function of the states and the characteristic partitions of the state space where this solution is feasible. In that way the solution of the MPC problem is obtained as piecewise affine feedback control law. The online computational effort is small since the online optimization problem is solved off-line and no optimizer is ever called online. In contrast, the online optimization problem is reduced to a mere function evaluation problem; when the measurements of the state are obtained and the corresponding region and control action are obtained by evaluation of a number of linear inequalities and a linear affine function, respectively. This is known as the online optimization via off-line parametric optimization concept.

References

- 1 ACEVEDO, J., PISTIKOPOULOS, E. N., *Ind. Eng. Chem. Res.* 35 (1996), p. 147
- 2 ACEVEDO, J., PISTIKOPOULOS, E. N., *Ind. Eng. Chem. Res.* 36 (1997), p. 2262
- 3 ACEVEDO, J., PISTIKOPOULOS, E. N., *Ind. Eng. Chem. Res.* 36 (1997), p. 717
- 4 ACEVEDO, J., PISTIKOPOULOS, E. N., *Oper. Res. Lett.* 24 (1999), p. 139
- 5 BANSAL, V., PERKINS, J. D., PISTIKOPOULOS, E. N., *AIChE J.* 46 (2000a), p. 335
- 6 BANSAL, V., PERKINS, J. D., PISTIKOPOULOS, E. N., *J. Stat. Comput. Simul.* 67 (2000b), pp. 219–253
- 7 BEMPORAD, A., MORARI, M., *Automatica* 35 (1999), p. 407
- 8 BEMPORAD, A., MORARI, M., in: *Robustness in Identification and Control*, Lecture Notes in Control and Information Sciences, vol. 245, Springer, Berlin, 1999
- 9 BEMPORAD, A., MORARI, M., DUA, V., PISTIKOPOULOS, E. N., Tech. Rep. AUT99-16, Automatic Control Lab, ETH Zürich, Switzerland, 1999
- 10 BEMPORAD, A., MORARI, M., DUA, V., PISTIKOPOULOS, E. N., *Automatica* 38 (2002), p. 3
- 11 CHISCI, L., ZAPPA, G., *Int. J. Control* 72 (1999), p. 1020

- 12 CHMIELEWSKI, D., MANOUSIOUTHAKIS, V., *Syst. Control Lett.* 29 (1996), p. 121
- 13 DUA, V., PISTIKOPOULOS, E. N., *Trans. IChemE* 76 (1998), p. 408
- 14 DUA, V., PISTIKOPOULOS, E. N., *Ann. Oper. Res.* 99 (1999), p. 123
- 15 DUA, V., PISTIKOPOULOS, E. N., *Ind. Eng. Chem. Res.* 38 (1999), p. 3976
- 16 EDGAR, T. F., HIMMELBLAU, D. M., *Optimization at Chemical*, McGraw-Hill, Singapore, 1989
- 17 ENGELL, S., KOWALEWSKI, S., KROGH, B. H., in: *5th Int. Conf. Chem. Proc. Control*, AIChE Symposium Series, vol. 93, 1997
- 18 FIACCO, A. V., *Math. Prog.* 10 (1976), p. 287
- 19 FIACCO, A. V., *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, USA, 1983
- 20 GAL, T., *Postoptimal Analyses, Parametric Programming, and Related Topics*, de Gruyter, New York, 1995
- 21 GAL, T., NEDOMA, J., *Math. Prog. Stud.* 18 (1972), p. 406
- 22 HENÉ, T. S., DUA, V., PISTIKOPOULOS, E. N., *Ind. Eng. Chem. Res.* 41 (2002), p. 67
- 23 MARLIN, T. E., HRYMAK, A. N., in: *5th Int. Conf. Chem. Proc. Control*, AIChE Symposium Series, vol. 93, 1997
- 24 MAYNE, D. Q., RAWLINGS, J. B., RAO, C. V., SCOKAERT, P. O. M., *Automatica* 36 (2000), p. 789
- 25 MORARI, M., LEE, J., *Comput. Chem. Eng.* 23 (1999), p. 667
- 26 PAPALEXANDRI, K., DIMKOU, T. I., *Ind. Eng. Chem. Res.* 37 (1998), p. 1866
- 27 PERTSINIDIS, A., On the parametric optimization of mathematical programs with binary variables and its application in the chemical engineering process synthesis, *Ph.D. Thesis*, Carnegie Mellon University, Pittsburg, PA, USA, 1992
- 28 PERTSINIDIS, A., GROSSMANN, I. E., McRAE, G. J., *Comput. Chem. Eng.* 22 (1998), p. S205.
- 29 PISTIKOPOULOS, E. N., DUA, V., in: *Proc. 3rd Int. Conf. on FOCAPO*, 1998
- 30 PISTIKOPOULOS, E. N., GROSSMANN, I. E., *Comput. Chem. Eng.* 12 (1988), p. 719
- 31 PISTIKOPOULOS, E. N., DUA, V., BOZINIS, N. A., BEMPORAD, A., MORARI, M., *Comput. Chem. Eng.* 26 (2002), p. 175
- 32 SCOKAERT, P., RAWLINGS, J., *IEEE Trans. Autom. Control* 43 (1998), p. 1163
- 33 SZNAIER, M., DAMBORG, M., in: *Proc. 26th IEEE Conf. on Decision and Control*, 1987
- 34 WRIGHT, S. J., in: *5th Int. Conf. Chem. Proc. Control*, AIChE Symposium Series, vol. 93, 1997
- 35 ZAFIRIOU, E., *Comput. Chem. Eng.* 14 (1990), p. 359
- 36 ZHENG, A., MORARI, M., *IEEE Trans. Autom. Control* 40 (1995), p. 1818