ISSN 0327-0793

LATIN AMERICAN APPLIED RESEARCH

Chemical Engineering
Applied Chemistry
Heat & Mass Transfer
Applied Mechanics
Control and Information Processing

PESQUISA APLICADA LATINO AMERICANA

Engenharia Química
Química Aplicada
Transfêrencia de Calor e Matéria
Mecânica Aplicada
Controle e Processamento da Informação

INVESTIGACIÓN APLICADA LATINOAMERICANA

Ingeniería Química
Química Aplicada
Transferencia de Calor y Materia
Mecánica Aplicada
Control y Procesamiento de la Información

Vol. 31 N° 4
September 2001

PIECEWISE LINEAR MODELS IN MODEL PREDICTIVE CONTROL

J. L. FIGUEROA*

Planta Piloto de Ing. Química - UNS-CONICET,
Camino la Carrindanga Km 7 - 8000 - Bahia Blanca - ARGENTINA
e-mail: cofiguer@criba.edu.ar
**also in Depto. de Ing. Eléctrica -UNS. Avda. Alem 1253, 8000- Bahía Blanca, Argentina*

Keywords: Model Predictive Control, Canonical Piecewise Lineal Approximations.

Abstract

In this paper an efficient algorithm for Nonlinear Model Predictive Control is presented. The nonlinear problem is written as a simple Quadratic Programming problem with constraints by using a Canonical Piecewise Lineal approximation to the model.

1. Introduction

It is undoubtedly true that the most control applications require satisfaction of hard constraints on controls and states; actuators saturation and safe operation requires limitation on states such as velocity, acceleration, temperature and pressure. Efficient handling of constraints requires nonlinear control whether the system being controlled is linear or nonlinear. The closed loop is, therefore, nonlinear which is the reason for the slow development of an adequate theory for the solution of these problems.

Model predictive control (MPC) refers to a class of algorithms that compute a sequence of adjustments of manipulated variables, in order to optimize the future behavior of the plant in the presence of constraints. Originally, model predictive algorithms were based on linear model and were developed to meet the specialized control needs of power plants and petroleum refineries. Now, they have been tested extensively and are presently well accepted in industry (Garcia et al, 1989). MPC technology can now be found in a wide variety of application areas including chemicals, food processing, automotive, aerospace, metallurgy and pulp and paper (Qin and Badgwell, 1997).

In the last decade, a substantial increase in the use of optimization based predictive algorithms for process control, has been observed. Alternative formulations for predictive control using nonlinear process models have also been considered. However, the solution of these nonlinear model predictive control (NMPC) algorithms is complex. This fact causes the application of these NMPC to be only advantageous where a substantial need for improved control quality exists, e.g. due to the strong nonlinear nature of the process or if changes during routine operation are large, as in batch process or during start-up or shut-down of a continuos process.

In this paper we propose the use of a Canonical Piecewise Linear approximation for the process model in order to obtain an efficient algorithm for the solution of a generic nonlinear model predictive control

Let us assume that the system to be controller is described by

$$\dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u},\mathbf{d}) \qquad (1)$$

$$\mathbf{y} = h(\mathbf{x}) \qquad (2)$$

where $f:\Re^n \times \Re^{m_i} \times \Re^{m_d} \to \Re^n$ and $h:\Re^n \to \Re^{n_o}$ are twice continuously differentiable, $\mathbf{x} \in \Re^n$ is the vector of state variables, $\mathbf{u} \in \Re^{m_i}$ is the vector of inputs, $\mathbf{d} \in \Re^{m_d}$ is the vector of the disturbances and $\mathbf{y} \in \Re^{n_o}$ is the vector of the outputs.

For a constant sample time T, $t_k = k.T$ is the k^{th} sampling interval and the discrete time operator corresponding to (1-2) can be defined as

$$\mathbf{x}_{k+1} = \Xi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \qquad (3)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) \qquad (4)$$

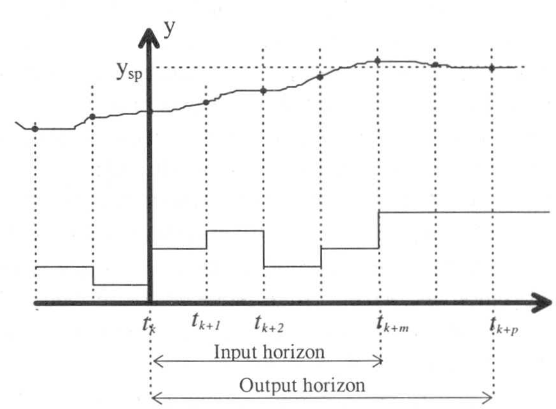

Figure 1: Input and Output Horizon in MPC

Let us assume that, this discrete model verifies, $\Xi(\mathbf{0},\mathbf{0},\mathbf{0}) = \mathbf{0}$, and the control and state are subject to hard constraints, $\mathbf{u}_k \in U$, $\mathbf{x}_k \in X$, $\mathbf{d}_k \in W$ for all k, where $U \subset \Re^{m_i}$ and $W \subset \Re^{m_d}$ are compact and convex sets and $X \subset \Re^n$ is a closed and convex set; and the origin lies in the interior of these sets.

The model predictive controller should predict inputs and outputs within the input horizon and output horizon respectively (Figure 1). So, an input horizon m and an output horizon p are defined, with $m<p$. Then the lengths of these horizons are given by $t_{ih}=mT$ and $t_{oh}=pT$, respectively. Within one sampling interval inputs, are constant and for $t_{ih} \le t_k \le t_{oh}$ the input are maintained constant and equal to its last value ($\mathbf{u}_{k+m}$). The basic idea of the model predictive control law is to minimize deviations of the controlled variables $\mathbf{y}$ from their setpoints $\mathbf{y}^{sp}$ within the output horizon, using the input vector $\mathbf{u}$ within the input horizon as decision variable. This optimization problem is constrained by a set of equality equations represented by the process model and a set of inequalities given by upper and lower bounds for inputs, outputs and states. The mathematical formulation of this problem is given as

$$
\begin{aligned}
\min_{\mathbf{u}_i, i=1,\dots,m} \Im_2 &= \sum_{i=1}^{p} \left(\mathbf{y}_i - \mathbf{y}_i^{sp}\right)^T \mathbf{Q}_y^i \left(\mathbf{y}_i - \mathbf{y}_i^{sp}\right) \\
&+ \sum_{i=1}^{p} \left(\mathbf{u}_i - \mathbf{u}_i^r\right)^T \mathbf{Q}_u^i \left(\mathbf{u}_i - \mathbf{u}_i^r\right) \\
s.t.\ & \mathbf{x}_{k+1} = \Xi(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k), k = 1,\dots,p \\
& \mathbf{y}_k = h(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \\
& \mathbf{u}_l \le \mathbf{u}_k \le \mathbf{u}_u \\
& \mathbf{x}_l \le \mathbf{x}_k \le \mathbf{x}_u \\
& \mathbf{y}_l \le \mathbf{y}_k \le \mathbf{y}_u \\
& \mathbf{u}_j = \mathbf{u}_{j-1}, j = m+1,\dots,p \\
& \mathbf{d}_j = \mathbf{d}_{j-1}, j = m+1,\dots,p
\end{aligned}
\tag{5}
$$

where $\mathbf{Q}_y^i$ and $\mathbf{Q}_u^i$ are weighting matrices. Here the subscript l stands for lower bound and the subscript u for upper bound. In this formulation the objective function does not only weight the deviations of the outputs from their setpoints, but contains an additional term to take into account the deviation of inputs from a reference input trajectory ($\mathbf{u}^r$), which accomplishes the same role as the setpoints $\mathbf{y}^{sp}$ for the outputs. It is chosen so, that in absence of disturbances the input sequence, $\mathbf{u}_{k+1} = \mathbf{u}_{k+1}^r$ with appropriated initial conditions keeps the outputs at their set points within the predictive horizon. In absence of future values for disturbances we assume that they are constant for the output horizon, i.e. $\mathbf{d}_k = \mathbf{d}$ for all . $\forall k = 1,\dots,p$

To simplify the notation, augmented vectors $\mathbf{U}(k)$, $\mathbf{D}(k)$, $\mathbf{X}(k)$ and $\mathbf{Y}(k)$ containing all values of the corresponding variables inside a predictive horizon, beginning at sample k, are introduced as,

$$\mathbf{U}(k) \in \Re^{pm_i} \equiv \left[\mathbf{u}_k^T \quad \mathbf{u}_{k+1}^T \quad \cdots \quad \mathbf{u}_{k+p-1}^T\right]^T, \tag{6}$$

$$\mathbf{D}(k) \in \Re^{pm_d} \equiv \left[\mathbf{d}_k^T \quad \mathbf{d}_{k+1}^T \quad \cdots \quad \mathbf{d}_{k+p-1}^T\right]^T, \tag{7}$$

$$\mathbf{X}(k) \in \Re^{pn} \equiv \left[\mathbf{x}_{k+1}^T \quad \mathbf{x}_{k+2}^T \quad \cdots \quad \mathbf{x}_{k+p}^T\right]^T, \tag{8}$$

$$\mathbf{Y}(k) \in \Re^{pn_o} \equiv \left[\mathbf{y}_{k+1}^T \quad \mathbf{y}_{k+2}^T \quad \cdots \quad \mathbf{y}_{k+p}^T\right]^T, \tag{9}$$

$\mathbf{Q}_y = diag\left\{\mathbf{Q}_y^i, i = 1,\dots,p\right\}$ and $\mathbf{Q}_u = diag\left\{\mathbf{Q}_u^i, i = 1,\dots,p\right\}$.

There is not an available algorithm to solve the problem (5) for on-line use due to its computational complexity. This fact has moved different authors to use some approximations to the model (for example using Neural Networks, Wienner or Hammerstein models) to solve this problem in an efficient way. Examples of this could be found in Su and McAvoy (1997), Zhu and Seborg (1994), Fruzzetti *et al.* (1997), Norquay *et al.* (1998), Genceli and Nikolaou (1995), Hernandez and Arkun (1993), Sriniwas and Arkun (1997).

Our approach in this paper is to use a Canonical Piecewise Linear approximation for the nonlinear model and use the properties of these models to solve (5). A description of these functions can be found in Chua and Deng (1986) or Figueroa and Desages (1998).

2. Problem Representation as CPWL.

Let the sets $X \subset \Re^n$, $U \subset \Re^{m_i}$, $W \subset \Re^{m_d}$ be the domains of the **x**, **u** and **d** variables respectively, and consider the set

$$\aleph = \left\{\left[\mathbf{x}^T, \mathbf{u}^T, \mathbf{d}^T\right]^T : \mathbf{x} \in X, \mathbf{u} \in U, \mathbf{d} \in W\right\}$$

on which we want to approximate to the given system (3)-(4). Consider also the following partition in the set $\aleph$ such that $\aleph = \bigcup_{j=1}^{\sigma} \aleph^j$, where $\aleph^j$ is called the "j^{th} *partition*" of the set $\aleph$.

Then, the CPWL representation of the system (3)-(4) is:

$$
\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{a}_x + \mathbf{B}_{xx}\mathbf{x}_k + \mathbf{B}_{xu}\mathbf{u}_k + \mathbf{B}_{xd}\mathbf{d}_k + \\
&\sum_{i=1}^{\sigma} \mathbf{c}_{xi} \left|\alpha_{xi}\mathbf{x}_k + \alpha_{ui}\mathbf{u}_k + \alpha_{di}\mathbf{d}_k - \beta_i\right|
\end{aligned}
\tag{10}
$$

$$
\begin{aligned}
\mathbf{y}_k &= \mathbf{a}_o + \mathbf{B}_{ox}\mathbf{x}_k + \mathbf{B}_{ou}\mathbf{u}_k + \mathbf{B}_{od}\mathbf{d} + \\
&\sum_{i=1}^{\sigma} \mathbf{c}_{oi} \left|\alpha_{xi}\mathbf{x}_k + \alpha_{ui}\mathbf{u}_k + \alpha_{di}\mathbf{d}_k - \beta_i\right|
\end{aligned}
\tag{11}
$$

where all the matrices and vectors have appropriate dimensions with elements in the real field.

If the system is constrained to the j^{th} region (i.e. $(\mathbf{x},\mathbf{u},\mathbf{d}) \in \aleph^j$), the equations (10-11) can be reformulated as follows:

$$\mathbf{x}_{k+1} = \xi_{xx}^j.\mathbf{x}_k + \xi_{xu}^j.\mathbf{u}_k + \xi_{xd}^j.\mathbf{d}_k + \eta_x^j \tag{12}$$

$$\mathbf{y}_k = \xi_{ox}^j.\mathbf{x}_k + \xi_{ou}^j.\mathbf{u}_k + \xi_{od}^j.\mathbf{d}_k + \eta_o^j \tag{13}$$

where $\xi_{xx}^j = \mathbf{B}_{xx} + \sum_{i=1}^{\sigma} \mathbf{c}_{xi}.\alpha_{xi}.\gamma_i^j$, $\xi_{xu}^j = \mathbf{B}_{xu} + \sum_{i=1}^{\sigma} \mathbf{c}_{xi}.\alpha_{ui}.\gamma_i^j$, $\xi_{xd}^j = \mathbf{B}_{xd} + \sum_{i=1}^{\sigma} \mathbf{c}_{xi}.\alpha_{di}.\gamma_i^j$, $\eta_x^j = \mathbf{a}_x + \sum_{i=1}^{\sigma} \mathbf{c}_{xi}.\beta_i.\gamma_i^j$,

$\xi^j_{ox} = \mathbf{B}_{ox} + \sum_{i=1}^{\sigma} \mathbf{c}_{oi}.\alpha_{xi}.\gamma_i^j$, $\xi^j_{ou} = \mathbf{B}_{ou} + \sum_{i=1}^{\sigma} \mathbf{c}_{oi}.\alpha_{ui}.\gamma_i^j$, $\xi^j_{od} = \mathbf{B}_{od} + \sum_{i=1}^{\sigma} \mathbf{c}_{oi}.\alpha_{di}.\gamma_i^j$, $\eta^j_o = \mathbf{a}_o + \sum_{i=1}^{\sigma} \mathbf{c}_{oi}.\beta_i.\gamma_i^j$ with $\gamma_i^j = sign(\alpha_{xi}\mathbf{x} + \alpha_{ui}\mathbf{u} + \alpha_{di}\mathbf{d} - \beta_i)$.

Note that the *sign* function in the last expression determines the ***Sector Belonging Condition,*** i.e., the sign vector γ^j defined as $\gamma^j = [\gamma_1^j, \gamma_2^j, ..., \gamma_\sigma^j]$ is univocally related to the j^{th} partition (Figueroa and Desages, 1998). Consequently, a point $(\mathbf{x}^h, \mathbf{u}^h, \mathbf{d}^h)$ will lie in $\aleph^j$ if and only if it satisfies the inequality

$$\mathbf{z}^j_\aleph = \xi^j_{\aleph x}.\mathbf{x}^h + \xi^j_{\aleph u}.\mathbf{u}^h + \xi^j_{\aleph d}.\mathbf{d}^h + \eta^j_\aleph \le \mathbf{0} \tag{14}$$

where $[\xi^j_{\aleph x}]_i = -\gamma_i^j \alpha_{xi}$, $[\xi^j_{\aleph u}]_i = -\gamma_i^j \alpha_{ui}$, $[\xi^j_{\aleph d}]_i = -\gamma_i^j \alpha_{di}$, $[\eta^j_\aleph]_i = \gamma_i^j \beta_i$ and $[.]_i$ means the i^{th} row in the matrix [.].

Now, we will analyze a way to obtain a predictive model for the system described by Equations (10)-(11). Consider the system at any initial condition $(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \in \aleph^0$. While the system is in sector $\aleph^0$ (suppose that this occurs for n^0 samples), it is easy to see that the state vector will be

$$\begin{aligned}\mathbf{x}_{k+i} = \xi^0_{xx}\mathbf{x}_{k+i-1} + \xi^0_{xu}.\mathbf{u}_{k+i-1} + \\ \xi^0_{xd}\mathbf{d}_{k+i-1} + \eta^0_x \qquad \forall i = 1,..,n^0\end{aligned} \tag{15}$$

This expression will be valid till the moment in which the system reach next sector (called $\aleph^1$), when the value of the state is

$$\begin{aligned}\mathbf{x}_{k+n^0} &= \xi^0_{xx}\mathbf{x}_{k+n^0-1} + \xi^0_{xu}\mathbf{u}_{k+n^0-1} + \xi^0_{xd}\mathbf{d}_{k+n^0-1} + \eta^0_x \\ &= \xi^0_{xx}\left(\xi^0_{xx}\mathbf{x}_{k+n^0-2} + \xi^0_{xu}\mathbf{u}_{k+n^0-2} + \xi^0_{xd}\mathbf{d}_{k+n^0-2} + \eta^0_x\right) + \\ &\quad \xi^0_{xu}\mathbf{u}_{k+n^0-1} + \xi^0_{xd}\mathbf{d}_{k+n^0-1} + \eta^0_x \\ &= \left(\xi^0_{xx}\right)^{n^0}\mathbf{x}_k + \xi^0_{xu}\mathbf{u}_{k+n^0-1} + \xi^0_{xx}\xi^0_{xu}\mathbf{u}_{k+n^0-2} + ... \\ &\quad + \left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xu}\mathbf{u}_k + \xi^0_{xd}\mathbf{d}_{k+n^0-1} + \xi^0_{xx}\xi^0_{xd}\mathbf{d}_{k+n^0-2} + \\ &\quad \left(\xi^0_{xx}\right)^2\xi^0_{xd}\mathbf{d}_{k+n^0-3} + ... + \left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xd}\mathbf{d}_k + \\ &\quad + \eta^0_x + \left(\xi^0_{xx}\right)\eta^0_x + \left(\xi^0_{xx}\right)^2\eta^0_x + ... + \left(\xi^0_{xx}\right)^{n^0-1}\eta^0_x\end{aligned}$$

and using this state as an initial condition for sector $\aleph^1$ it is possible to compute

$$\begin{aligned}\mathbf{x}_{k+i} = \xi^1_{xx}\mathbf{x}_{k+i-1} + \xi^1_{xu}.\mathbf{u}_{k+i-1} + \\ \xi^1_{xd}\mathbf{d}_{k+i-1} + \eta^1_x \qquad \forall i = n^0+1,..,n^1\end{aligned} \tag{16}$$

expression that will be valid till the time in which the system reaches sector $\aleph^2$ (at n^1). Then, it is possible to obtain a Predictive Model by using the following algorithm,

Algorithm 1: Predictive Model

Data: A set of control variables $[\mathbf{u}_k, \mathbf{u}_{k+1}, \cdots, \mathbf{u}_{k+m}]$, disturbances $[\mathbf{d}_k, \mathbf{d}_{k+1}, \cdots, \mathbf{d}_{k+m}]$ and the initial state vector $\mathbf{x}_k$, set j=0 and $n^{-1}=0$.

Step 1: Determine in which sector, $\aleph^j$, point $(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k)$ lies and compute the linear model valid in this sector.

Step 2: If $k<p$ continue; Otherwise ***stop.***

Step 3: Compute $\mathbf{x}_{k+1} = \xi^j_{xx}\mathbf{x}_k + \xi^j_{xu}.\mathbf{u}_k + \xi^j_{xd}\mathbf{d}_k + \eta^j_x$.

Step 3: If no entry of the vector,

$\mathbf{z}^j_\aleph = \xi^j_{\aleph x}\mathbf{x}_{k+1} + \xi^j_{\aleph u}\mathbf{u}_k + \xi^j_{\aleph w}\mathbf{d}_k + \eta^j_\aleph$;

changes sign, make k=k+1 and return to Step 2. Otherwise, make $n^j = k - n^{j-1}$ and j=j+1 and return to Step 1 to proceed similarly in next sector.

Using the results of this algorithm, a generic expression for the predictive model when the system goes through sectors [$\aleph^0$, $\aleph^1$, $\aleph^2$, …., $\aleph^h$] could be written as

$$\mathbf{X} = \Phi_{xx}\mathbf{x}_k + \Phi_{xu}\mathbf{U} + \Phi_{xd}\mathbf{D} + \Phi_x \tag{17}$$

$$\mathbf{Y} = \Phi_{yx}\mathbf{x}_k + \Phi_{yu}\mathbf{U} + \Phi_{yd}\mathbf{D} + \Phi_y \tag{18}$$

where the matrices are defined in Appendix A.

In some applications, it is not necessary to consider changes of the manipulated vectors in each period. Moreover, it could unnecessarily increase the dimension of the vector **U** for the solution of the Model Predictive Control. To avoid this problem, we propose to modify the matrices involved in Eqns. (17)-(18) in order to consider another sampling time. To do this, we will consider that the process input (manipulated variable) changes each t sampling times. Then, for an input horizon m (the input signal changes m times) the length of this horizon is $t_{ih}=m.t.T$. Similarly, the length of the output horizon is $t_{oh}=p.t.T$. Note that in this case, the matrices involved in Eqns. (17)-(18) could be easily computed by adding up all t columns inside each r rows. For example, if the original matrix is $\Phi_{xu} = [\Phi_{xu}]_{i,j}$ for i,j= 1,..,$p.t$, the resulting matrix will be

$$\tilde{\Phi}_{xu} = [\tilde{\Phi}_{xu}]_{k,h} = \sum_{i=1}^{t-1} [\Phi_{xu}]_{k.t,(h-1).t+i} \quad \text{for } k,h=1,..,p$$

Now, let us rewrite the Model Predictive Control Problem using this predictive model. In absence of information about future disturbances we will consider it constants, i.e. $\mathbf{d}_k = \mathbf{d} \quad \forall k = 1,....,p$. Then, from the analysis of problem (5) and the predictive model (17-18); and considering that the disturbance vector **D** is constant and the initial state vector $\mathbf{x}_k$ is known a priori, for our application we can pose this model as,

$$\mathbf{X} = \tilde{\Phi}_x(\mathbf{x}_k) + \Phi_{xu}\mathbf{U} \tag{19}$$

$$\mathbf{Y} = \tilde{\mathbf{\Phi}}_y(\mathbf{x}_k) + \mathbf{\Phi}_{yu}\mathbf{U} \tag{20}$$

with

$$\tilde{\mathbf{\Phi}}_x(\mathbf{x}_k) = \mathbf{\Phi}_{xx}\mathbf{x}_k + \mathbf{\Phi}_{xd}\mathbf{D} + \mathbf{\Phi}_x$$

and $\tilde{\mathbf{\Phi}}_y(\mathbf{x}_k) = \mathbf{\Phi}_{yx}\mathbf{x}_k + \mathbf{\Phi}_{yd}\mathbf{D} + \mathbf{\Phi}_y$.

Replacing these expressions in each term of the problem 5, we can obtain the objective function,

$$\begin{aligned}\Im_2 &= \sum_{i=1}^{p}(\mathbf{y}_i - \mathbf{y}_i^{sp})^T \mathbf{Q}_y^i(\mathbf{y}_i - \mathbf{y}_i^{sp}) + \\ &\sum_{i=1}^{p}(\mathbf{u}_i - \mathbf{u}_i^r)^T \mathbf{Q}_u^i(\mathbf{u}_i - \mathbf{u}_i^r) \\ &= \mathbf{U}^T\mathbf{\Theta}_{uu}\mathbf{U} + \mathbf{\Theta}_u\mathbf{U} + \mathbf{\Theta}\end{aligned} \tag{21}$$

where $\mathbf{\Theta} = \left((\tilde{\mathbf{\Phi}}_x - \mathbf{Y}^{sp})^T \mathbf{Q}_y(\tilde{\mathbf{\Phi}}_x - \mathbf{Y}^{sp}) + \mathbf{U}^{rT}\mathbf{Q}_u\mathbf{U}^r\right)$,

$\mathbf{\Theta}_{uu} = (\mathbf{\Phi}_{xu}^T\mathbf{Q}_y\mathbf{\Phi}_{xu} + \mathbf{Q}_u)$ and $\mathbf{\Theta}_u = 2\left((\tilde{\mathbf{\Phi}}_x - \mathbf{Y}^{sp})^T\mathbf{Q}_y\mathbf{\Phi}_{xu} + \mathbf{U}^{rT}\mathbf{Q}_u\right)$.

$$\begin{aligned}\Im_2 &= \sum_{i=1}^{p}(\mathbf{y}_i - \mathbf{y}_i^{sp})^T \mathbf{Q}_y^i(\mathbf{y}_i - \mathbf{y}_i^{sp}) + \sum_{i=1}^{p}(\mathbf{u}_i - \mathbf{u}_i^r)^T \mathbf{Q}_u^i(\mathbf{u}_i - \mathbf{u}_i^r) \\ &= (\mathbf{Y} - \mathbf{Y}^{sp})^T\mathbf{Q}_y(\mathbf{Y} - \mathbf{Y}^{sp}) + (\mathbf{U} - \mathbf{U}^r)^T\mathbf{Q}_u(\mathbf{U} - \mathbf{U}^r) \\ &= (\tilde{\mathbf{\Phi}}_x + \mathbf{\Phi}_{xu}\mathbf{U} - \mathbf{Y}^{sp})^T\mathbf{Q}_y(\tilde{\mathbf{\Phi}}_x + \mathbf{\Phi}_{xu}\mathbf{U} - \mathbf{Y}^{sp}) + (\mathbf{U} - \mathbf{U}^r)^T\mathbf{Q}_u(\mathbf{U} - \mathbf{U}^r) \\ &= \mathbf{U}^T(\mathbf{\Phi}_{xu}^T\mathbf{Q}_y\mathbf{\Phi}_{xu} + \mathbf{Q}_u)\mathbf{U} + \left((\tilde{\mathbf{\Phi}}_x - \mathbf{Y}^{sp})^T\mathbf{Q}_y\mathbf{\Phi}_{xu} + \mathbf{U}^{rT}\mathbf{Q}_u\right)\mathbf{U} + \\ &\quad \left((\tilde{\mathbf{\Phi}}_x - \mathbf{Y}^{sp})^T\mathbf{Q}_y(\tilde{\mathbf{\Phi}}_x - \mathbf{Y}^{sp}) + \mathbf{U}^{rT}\mathbf{Q}_u\mathbf{U}^r\right) \\ &= \mathbf{U}^T\mathbf{\Theta}_{uu}\mathbf{U} + \mathbf{\Theta}_u\mathbf{U} + \mathbf{H}\end{aligned}$$

Similarly, the constraints are now expressed as

$$\mathbf{H}\,\mathbf{U} \le \mathbf{h} \tag{22}$$

with

$$\mathbf{H} = \begin{bmatrix}\mathbf{I} \\ -\mathbf{I} \\ \mathbf{\Phi}_{xu} \\ -\mathbf{\Phi}_{xu} \\ \mathbf{\Phi}_{yu} \\ -\mathbf{\Phi}_{yu}\end{bmatrix} \text{ and } \mathbf{h} = \begin{bmatrix}\mathbf{U}_u \\ -\mathbf{U}_l \\ \mathbf{X}_u - \tilde{\mathbf{\Phi}}_x \\ -\mathbf{X}_l + \tilde{\mathbf{\Phi}}_x \\ \mathbf{Y}_u - \tilde{\mathbf{\Phi}}_y \\ -\mathbf{Y}_l + \tilde{\mathbf{\Phi}}_y\end{bmatrix},$$

Then, the Model Predictive Control problem could be solved as,

$$\begin{aligned}&\min_{\mathbf{U}} \Im_2 = \mathbf{U}^T\mathbf{\Theta}_{uu}\mathbf{U} + \mathbf{\Theta}_u\mathbf{U} + \mathbf{\Theta} \\ &s.t. \\ &\quad \mathbf{HU} \le \mathbf{h}\end{aligned} \tag{23}$$

which is a typical quadratic problem, that could be solved using any commercial algorithm. By solving this problem, we can suggest the following algorithm for MPC

Algorithm 2: Predictive Control Algorithm

Step 0: Check actual system state.(i.e., vectors $\mathbf{x_0}$ and **d**). Fix an initial guess for the control vector **U**. Set $k=0$.

Step 1: Compute the Predictive model of Eqns. (19-20).

Step 2: Compute the control vector **U** by solving the Model Predictive Control Problem of Eq. (23).

Step 3: If the model in Step 1 is not longer valid for the new vector **U**, return to Step 1. Otherwise, go on.

Step 4: Apply the control action $\mathbf{u_k}$ to the system.

Step 5: Estimate the resulting system state vector $\mathbf{x_{k+1}}$, set $k=k+1$ and return to Step 1.

3. Example

Performance of the Algorithm for MPC will be now analyzed by applying it to the control of a CSTR. Figure 2 shows the considered system. Within a CSTR, an isothermal pseudo first order reaction A+B→P, is conducted with an excess concentration of A. Two input streams of different concentrations of B are used in order to provide a good adjustment of the operating conditions. The reactor is assumed to be well mixed. The output flowrate is determined by the liquid level in the reactor. The process can be described by the following model,

$$\frac{dx_1}{dt} = u_1 + u_2 - k_1\sqrt{x_1}$$
$$\frac{dx_2}{dt} = (C_{B1} - x_2)\frac{u_1}{x_1} + (C_{B2} - x_2)\frac{u_2}{x_1} - \frac{k_2 x_2}{(1 + x_2)^2}$$

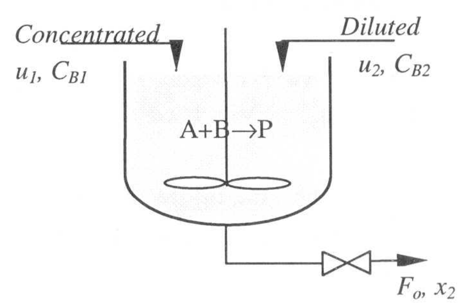

Figure 2: Stirred Tank

The desired steady state for this system is given for the following values of the parameters; u_1: flowrate of the inlet concentrated stream (nominal value 0.2344); u_2: flowrate of the inlet diluted stream (1.7656); C_{B1}: concentration of B in stream 1 (24.9); C_{B2}: concentration of B in stream 2 (0.1); k_1: valve constant (0.2); k_2: kinetic constant (1); x_1: liquid height in the reactor (100); x_2: concentration of B in the reactor (0.0669)

The states x_1 and x_2 are the controlled outputs, the variables u_1 and u_2 are the manipulated variables and the concentrations C_{B1} and C_{B2} are the disturbances inputs. A discrete CPWL model was used to solve the Model Predictive Control Problem. In this approximation, the space is divided in 75 Sectors. In this example, the sample time T is 0.1 minutes and the horizons are t=10 and p=m=50. The weighting matrices are defined as $Q_u^i = I$ and

$$Q_x^i = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 10^3 \end{bmatrix} & for \quad i = 1,\ldots,p-1 \\ \begin{bmatrix} 10 & 0 \\ 0 & 10^4 \end{bmatrix} & for \quad i = p \end{cases}$$

The manipulated variables are constrained to

$$0.9 * 0.2344 \le u_1 \le 1.1 * 0.2344$$
$$0.9 * 1.7656 \le u_1 \le 1.1 * 1.7656$$

The states are limited in the output horizon as

$$\left|x_1 - x_1^{sp}\right| \le \frac{100}{2} e^{-0.0025i} \qquad \forall i = 1,..,p$$
$$\left|x_2 - x_2^{sp}\right| \le \frac{0.0669}{2} e^{-0.0025i} \qquad \forall i = 1,..,p$$

where the superscript *sp* means the set point value.

For these variables, a set point change is produced in the variable x_1 from 100 to 115 at t=10 minutes. The performance of the MPC scheme defined in last section is shown in Figure 3. When disturbances of $\pm 10\%$ is applied to C_{b1} and C_{b2} the performance could be shown from Figure 4. From both proofs it is clear that the performance of this controller is satisfactory.

4. Conclusions

An algorithm for Nonlinear Model Predictive Control has been presented. It is based in a transformation of the nonlinear problem into a simple Quadratic Programming Problem by using a Canonical Piecewise Lineal approximation of the process model. The applicability of this method is tested in two examples.

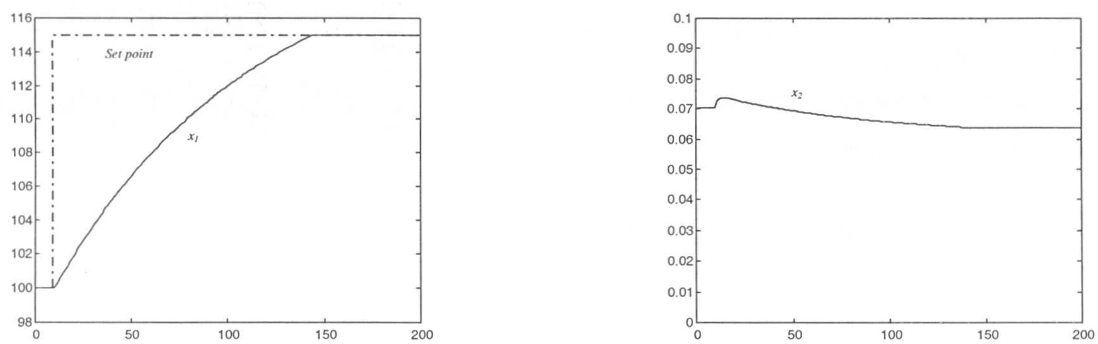

Figure 3: Performance for change in set point

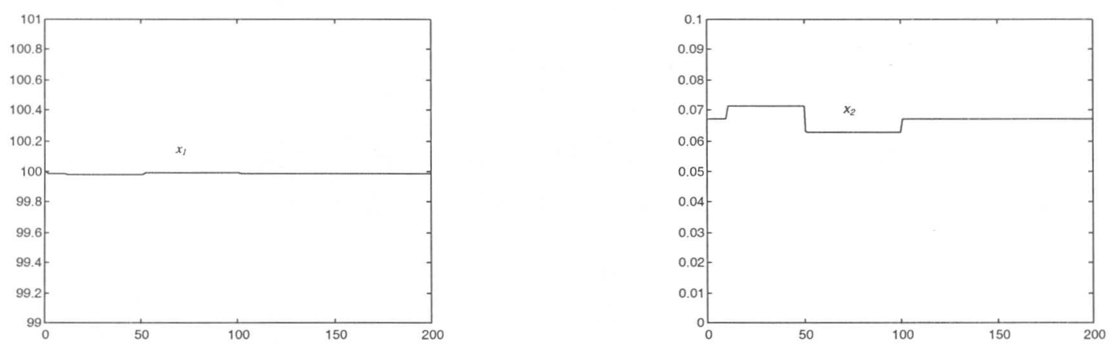

Figure 4: Performance for Disturbances

References

Chua L.O. and A.C. Deng, "Canonical Piecewise-Linear Modeling," *IEEE Trans. Circuits and Systems* **33**, 511-525. (1986)

Figueroa J.L. and A.C. Desages, "Use of Piecewise Linear Approximations for Steady-State Back-off Analysis," *Optimal Control: Appls and Methods* **19**, 93-100, (1998).

Fruzzetti K.P., A. Palazoglu and K.A. McDonald, "Nonlinear Model Predictive Control using Hammerstein Models," *Journal of Process Control* **7**, 31-41, (1997).

Garcia,E., D. Prett and M. Morari, "Model Predictive Control: Theory and Practice-a Survey," *Automatica* **25**, 335-348, (1989).

Genceli H. and M. Nikolaou, "Design of Robust Constrained Model-Predictive Controllers with Volterra Series," *AIChE Journal* **41**, pp. 2098-2107, (1995).

Hernandez E. and Y. Arkun, "Control of nonlinear systems using polynomial ARMA models," *AIChE Journal* **39**, 446-460, (1993).

Norquay S.J., A. Palazoglu and J.A. Romagnoli, "Model Predictive Control Based on Wiener

Models," *Chemical Engineering Science* **53**, 75-84. (1998).

Qin, S.J. and T.A. Badgwell, "An Overview of Industrial Model Predictive Control Technology," *Fifth International Conference on Chemical Process Control,* Ed. Kantor, Garcia and Carnahan, AIChE Symposium Series No. 316, **93**, 232-257. (1997).

Sriniwas G.R. and Y. Arkun, "A Global Solution to the Nonlinear Model Predictive Control Algorithms Using Polynomial ARX models," *Computers and Chemical Engineering* **21**, 431-439, (1997).

Su H.T. and T.J. McAvoy, "Artificial Neural Networks for nonlinear process identification and control," *Nonlinear Process Control.* Henson and Seborg (Eds.), Englewood Cliffs, NJ, Prentice-Hall, Chapter 7, 371-428, (1997).

Zhu X. and D.E. Seborg, "Nonlinear Model Predictive Control Based on Hammerstein Models," *Proc. Int. Symp. on Process System Engineering,* Seul, Korea, 995-1000, (1994).

Appendix A: CPWL Predictive Model

The matrices that defines de Predictive Model

$$\mathbf{X} = \mathbf{\Phi}_{xx}\mathbf{x}_k + \mathbf{\Phi}_{xu}\mathbf{U} + \mathbf{\Phi}_{xd}\mathbf{D} + \mathbf{\Phi}_x \qquad \text{(B1)}$$

are defined as

$$\mathbf{\Phi}_{xx} = \begin{bmatrix} \xi^0_{xx} \\ \left(\xi^0_{xx}\right)^2 \\ \vdots \\ \xi^1_{xx}\left(\xi^0_{xx}\right)^{n^0} \\ \vdots \\ \left(\xi^h_{xx}\right)^{n^h-1}\cdots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0} \\ \left(\xi^h_{xx}\right)^{n^h}\cdots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0} \end{bmatrix}$$

$$\mathbf{\Phi}_x = \begin{bmatrix} \eta^0_x \\ \xi^0_{xx}\eta^0_x + \eta^0_x \\ \vdots \\ \xi^1_{xx}\sum_{\iota=1}^{n^0}\left(\xi^0_{xx}\right)^{\iota-1}\eta^0_x + \eta^1_x \\ \vdots \\ \sum_{j=1}^{n^h-1}\left(\left(\xi^h_{xx}\right)^j\left(\ldots\sum_{\iota=1}^{n^0}\left(\xi^0_{xx}\right)^{\iota-1}\eta^0_x\ldots\right) + \left(\xi^h_{xx}\right)^{j-1}\eta^h_x\right) \\ \sum_{j=1}^{n^h}\left(\left(\xi^h_{xx}\right)^j\left(\ldots\sum_{\iota=1}^{n^0}\left(\xi^0_{xx}\right)^{\iota-1}\eta^0_x\ldots\right) + \left(\xi^h_{xx}\right)^{j-1}\eta^h_x\right) \end{bmatrix}$$

$$\mathbf{\Phi}_{xu} = \begin{bmatrix} \xi^0_{xu} & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \xi^0_{xx}\xi^0_{xu} & \xi^0_{xu} & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \xi^1_{xx}\left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xu} & \xi^1_{xx}\left(\xi^0_{xx}\right)^{n^0-2}\xi^0_{xu} & \cdots & \xi^1_{xu} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \left(\xi^h_{xx}\right)^{n^h-1}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xu} & \left(\xi^h_{xx}\right)^{n^h-1}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-2}\xi^0_{xu} & \cdots & \left(\xi^h_{xx}\right)^{n^h-1}\ldots\left(\xi^1_{xx}\right)^{n^0-1}\xi^1_{xu} & \cdots & \xi^h_{xu} & 0 \\ \left(\xi^h_{xx}\right)^{n^h}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xu} & \left(\xi^h_{xx}\right)^{n^h}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-2}\xi^0_{xu} & \cdots & \left(\xi^h_{xx}\right)^{n^h}\ldots\left(\xi^1_{xx}\right)^{n^0-1}\xi^1_{xu} & \cdots & \xi^h_{xx}\xi^h_{xu} & \xi^h_{xu} \end{bmatrix}$$

$$\mathbf{\Phi}_{xd} = \begin{bmatrix} \xi^0_{xd} & 0 & \cdots & 0 & \cdots & 0 & 0 \\ \xi^0_{xx}\xi^0_{xd} & \xi^0_{xd} & \cdots & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \xi^1_{xx}\left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xd} & \xi^1_{xx}\left(\xi^0_{xx}\right)^{n^0-2}\xi^0_{xd} & \cdots & \xi^1_{xd} & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ \left(\xi^h_{xx}\right)^{n^h-1}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xd} & \left(\xi^h_{xx}\right)^{n^h-1}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-2}\xi^0_{xd} & \cdots & \left(\xi^h_{xx}\right)^{n^h-1}\ldots\left(\xi^1_{xx}\right)^{n^0-1}\xi^1_{xd} & \cdots & \xi^h_{xd} & 0 \\ \left(\xi^h_{xx}\right)^{n^h}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-1}\xi^0_{xd} & \left(\xi^h_{xx}\right)^{n^h}\ldots\left(\xi^1_{xx}\right)^{n^1}\left(\xi^0_{xx}\right)^{n^0-2}\xi^0_{xd} & \cdots & \left(\xi^h_{xx}\right)^{n^h}\ldots\left(\xi^1_{xx}\right)^{n^0-1}\xi^1_{xd} & \cdots & \xi^h_{xx}\xi^h_{xd} & \xi^h_{xd} \end{bmatrix}$$

where n^i is the sample time at which the system leaves sector $\aleph^i$. Note the dependence of the matrices $\mathbf{\Phi}_{xx}$, $\mathbf{\Phi}_{xu}$, $\mathbf{\Phi}_{xd}$ and $\mathbf{\Phi}_x$ on the sectors $[\aleph^0, \aleph^1, \aleph^2, \ldots, \aleph^h]$ and on the "times" $[n^0, n^1, n^2, \ldots, n^h]$. This means that, in general, expression (B1) is not longer valid if any change occurs in the inputs $\mathbf{u_k}$ or $\mathbf{d_k}$ for all k=1,..,p. Note that in these expressions $p=n^0 + n^1 + n^2 + \ldots, + n^h$.

Respect to the output equation, they are

$$\begin{aligned} \mathbf{Y} &= \mathbf{\Phi}_{yx}\mathbf{x}_k + \mathbf{\Phi}_{yu}\mathbf{U} + \mathbf{\Phi}_{yd}\mathbf{D} + \mathbf{\Phi}_y \\ &= \left(\mathbf{\Phi}_{ox}\mathbf{\Phi}_{xx}\right)\mathbf{x}_k + \left(\mathbf{\Phi}_{ox}\mathbf{\Phi}_{xu} + \mathbf{\Phi}_{ou}\right)\mathbf{U} + \\ &\quad \left(\mathbf{\Phi}_{ox}\mathbf{\Phi}_{xd} + \mathbf{\Phi}_{od}\right)\mathbf{D} + \left(\mathbf{\Phi}_{ox}\mathbf{\Phi}_x + \mathbf{\Phi}_o\right) \end{aligned}$$

where

$$\Phi_{ox} = diag\ block\Big[\underbrace{\xi^0_{ox}, \ \cdots \ \xi^0_{ox}}_{n^0-1}, \cdots, \underbrace{\xi^h_{ox}, \ \cdots \ \xi^h_{ox}}_{n^h+1}\Big],$$

$$\Phi_{ou} = diag\ block\Big[\underbrace{\xi^0_{ou}, \ \cdots \ \xi^0_{ou}}_{n^0-1}, \cdots, \underbrace{\xi^h_{ou}, \ \cdots \ \xi^h_{ou}}_{n^h+1}\Big],$$

$$\Phi_{od} = diag\ block\Big[\underbrace{\xi^0_{od}, \ \cdots \ \xi^0_{od}}_{n^0-1}, \cdots, \underbrace{\xi^h_{od}, \ \cdots \ \xi^h_{od}}_{n^h+1}\Big]$$

and

$$\Phi_{o} = \Big[\underbrace{(\eta^0_o)^t, \ \cdots \ (\eta^0_o)^t}_{n^0-1}, \cdots, \underbrace{(\eta^h_o)^t, \ \cdots \ (\eta^h_o)^t}_{n^h+1}\Big]^t$$

Received August 30 1999;
accepted for publication April 30 2001
Recommended by T. Pinto and Bolzan