USE OF WIENER NONLINEAR MPC TO CONTROL A CSTR WITH MULTIPLE STEADY STATE

A. LUSSON CERVANTES, O.E. AGAMENNONI and J.L. FIGUEROA†

Departamento de Ingeniería Eléctrica, Universidad Nacional del Sur
† Also in Planta Piloto de Ingeniería Química - PLAPIQUI (UNS-CONICET)
Camino La Carrindanga Km. 7 (8000) Bahía Blanca. Argentina
jfigueroa@plapiqui.edu.ar

Abstract— In this paper a Nonlinear Model Predictive Control based on a Wiener Model with a Piecewise Linear gain is presented. The major advantages of this algorithm is that it retains all the interesting properties of the classical linear MPC and the computations are easy to solve due to the canonical structure of the nonlinear gain. The proposed control scheme is applied to a nonlinear CSTR that presents multiple steady states.

Keywords— Model Predictive Control, Wiener Models, Piecewise Linear Approximation.

I. INTRODUCTION

Model predictive control (MPC) refers to a class of computer control algorithms that control the future behavior of a plant through the use of an explicit process model. At each control interval the MPC algorithm computes an open-loop sequence of manipulated variable adjustments in order to optimize future plant behavior. The first input in the optimal sequence is injected into the plant, and the entire optimization is repeated at subsequent control intervals. MPC technology was originally developed for power plant and petroleum refinery applications, but can now be found in a wide variety of manufacturing environments including chemicals, food processing, automotive, aerospace, metallurgy and pulp and paper (Qin and Badgwell, 1997).

Though manufacturing processes are inherently nonlinear, the vast majority of MPC applications to date are based on linear dynamic models, the most common being step and impulse response models derived from the convolution integral. There are several potential reasons for this, for example, by using a linear model and a quadratic objective, the nominal MPC algorithm takes the form of a highly structured convex Quadratic Programming problem (QP), for which reliable solution algorithms and software can easily be found. This is important because the solution algorithm must converge reliably to the optimum in no more than few tens of seconds to be useful in manufacturing applications.

Nevertheless, there are cases where nonlinear effects are significant enough to justify the use of Nonlinear Model Predictive Control (NMPC). With the introduction of a dynamic nonlinear models within the NMPC algorithm, the complexity of the predictive control problem increases significantly. Review papers by Henson (1998) and Bequette (1991) elaborate on the various approaches to handling nonlinear systems via MPC. For example, many researches have used empirical models for NMPC (Norquay et al., 1998; Su and McAvoy, 1997; Zhu and Seborg, 1994).

In particular, Wiener models have a special structure that facilitates their application to NMPC. Its application has been proved both in academic (Norquay et al., 1998) and practical aspects (Norquay et al., 1999). These models represent a process with linear dynamic but a nonlinear gain, and can represent many of the memoryless nonlinear systems commonly encountered in industrial processes. Due to the static nature of the nonlinearities, they can be removed from the control problem. However some computational difficulty is potentially present and due to that an implicit inversion of the nonlinear static gain is needed. In this work the application of a discrete time nonlinear predictive control of Wiener systems to the control of a CSTR with multiple steady states is analyzed. More specifically, the Wiener structure consists of a linear dynamic element followed in series by a Static nonlinear element. The linear dynamic element uses a discrete state space model and the nonlinear element uses the Piecewise Linear approximation (Julian 1999). The paper is organized as follows: In section II a description of the proposed control scheme is included. In section III the case study is analyzed and finally, some conclusions are discussed in section IV.

II. NONLINEAR WIENER MPC USING PWL FUNCTIONS

Let us assume that the system to be controlled can be described by the following discrete-time, nonlinear, state-space model:

\[ x(k+1) = g(x(k), \Delta u(k)) \]
\[ y(k) = h(x(k)) + d(k) \]  

(1)
where the functions are twice continuously differentiable functions, \( x(k) \) is the vector of state variables, \( \Delta u(k) \) is the vector of control movements, \( d(k) \) is the vector of additive disturbance variables and \( y(k) \) is the vector of process output.

In this work, we will use a specific Wiener approximation to represent the process model. In general, a Wiener model consists of a dynamic linear block in cascade with a static non-linearity at the output. In our application, a discrete state space model is used to describe the linear dynamic element as shown in Fig. 1. For the static nonlinear element, we propose the use of a Continuous Piecewise Linear (PWL) function \( y=f(v) \). The PWL functions have proved to be a very powerful tool in the modeling and analysis of nonlinear systems. In particular, any function \( f \) can be written as a linear combination of the elements of a basis \( \Lambda \) as \( f(v)=\sum_{i=1}^{\infty} \lambda_i \Lambda_i(v) \). This kind of function (Lusson Cervantes et al., 2001) allows developing a systematic and accurate treatment of approximating functions. For details of the definition and properties of PWL see Appendix A.

In our application, the domain and the image of the PWL function shares the same dimension. Moreover, if we assume that the function \( f \) of the system is bijective (this is a reasonable assumption for a large set of process models), it is possible to define the inverse function as \( f^{-1} \), such that \( \Delta u(v)=f^{-1}(f(v)) \). This function is also unique and it is a PWL.

The control problem to be solved is to compute a sequence of inputs \( \Delta u(k) \) \( \{k=1,\ldots,m\} \) that will minimize the following dynamic objective:

\[
J = \sum_{j=1}^{J} \left| v(k+j) - r \right|_{b} + \sum_{j=p}^{m} \left\| \Delta u(k+j) \right\|_{b} ,
\]

subject to model equations and to inequality constraints:

\[
y' \leq y(k+j) \leq y^* \quad \forall j = 1, \ldots, p-1
\]

\[
u' \leq u(k+j) \leq u^* \quad \forall j = 1, \ldots, m-1
\]

where \( p \) is the prediction horizon, \( m \) is the control horizon, \( r \) is the desired set point, and the relative importance of the objective function contributions is controlled by setting the time dependent weight matrices \( Q_j \) and \( R_j \). Beyond the control horizon the control signal is assumed to be constant (i.e. \( \Delta u(k+j)=0, j=m,\ldots,p \)). Disturbances are typically handled by assuming that a step disturbance has entered at the output and that it will remain constant for all future time (\( d(k)=d(k+j), j=1,\ldots,m \)). To accomplish this, a bias term that compares the current predicted output to the current measured is computed (i.e., \( d(k)=y(k)-y^*(k) \)).

Once \( \Delta u(k) \) is computed, following the receding horizon principle, only the first element of the optimal control sequence is used as the current control value. Then, the horizons shift one step forward in time and the whole procedure is repeated. If we assume to know the state vector at the actual time and write the future behavior of the variables for samples \( k+1 \) to \( k+m \) as vectors \( \mathbf{v}(k), \mathbf{\Delta U}(k), y(k) \) and \( r(k) \); then, the predicted output for the linear model is

\[
\hat{y}(k) = \mathbf{\beta} \mathbf{U}(k) + \xi \mathbf{x}(k)
\]

where \( \hat{y}(k) \) is an estimation for the output of the lineal model and

\[
\mathbf{\beta} = \begin{bmatrix} c^T b + d & 0 & \cdots & 0 \\ c^T A b & c^T b + d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c^T A^{m-1} b & c^T A^{m-2} b & \cdots & c^T A^{m-1} b \end{bmatrix}, \quad \mathbf{\xi} = \begin{bmatrix} c^T \\ c^T A \\ \vdots \\ c^T A^{m-1} \end{bmatrix}
\]

and the predicted output for the complete model is

\[
\hat{y}(k) = f(\hat{y}(k)) = \left[ f(v(k+1)) \cdots f(v(k+p)) \right] \mathbf{U}(k)
\]

Let us define some points in relation with the MPC structure,

- Since the PWL function \( f \) was assumed invertible, it is possible to write the desired signal referred to the output of the linear model as a transformation of the set point \( r(k) \) as, \( r'(k)=f^{-1}(r(k)) \).

- If \( y' \) and \( y' \) are the upper and lower bounds defined in (3), then, we can translate this magnitudes to the linear model as \( y'=f^{-1}(y') \) and \( y'=f^{-1}(y') \), respectively.

Finally, the WMPC can be posed as

\[
\min_{\mathbf{v}} J = \left\| \hat{y}(k)-r \right\|_{Q(\hat{y}(k)-r)} + \left\| U^T R U \right\|_{Q^T Q}
\]

s.t.

\[
\hat{y}(k) = \mathbf{U} + \mathbf{x}(k)
\]

\[
y' \leq \hat{y}(k) \leq y^* \quad \forall j = 1, \ldots, p-1
\]

\[
u' \leq u(k+j) \leq u^* \quad \forall j = 1, \ldots, m-1
\]

Note that minimization of (5) is a classical LMPC, then all properties about convergence to the global optimal solution and stability of the loop can be guaranteed.
III. CASE STUDY: STIRRED-TANK REACTOR (CSTR)

A. Process Description

The simulated continuous stirred-tank reactor (CSTR) process (Morningred et al., 1992) consists of an irreversible, exothermic reaction, A→B, in a constant volume reactor cooled by a single coolant stream, which can be modeled by the following equations (Giovanini, 2000).

\[
\dot{C}_A(t) = \frac{dq(t)}{dt} [C_{A0}(t) - C_A(t)] - k_2 C_A(t) \exp\left(-\frac{E}{RT(t)}\right)
\]

\[
\dot{T}(t) = \frac{q(t)}{\rho c_p} \left[ T_0(t) - T(t) \right] - k_2 \Delta H \rho \frac{C_A(t)}{\rho c_p} \exp\left(-\frac{E}{RT(t)}\right) \cdot \left[ 1 - \exp\left(-\frac{q(t)\rho c_p}{q(t)\rho c_p}T_C(t)\right)\right] + \frac{\rho c_p}{\rho c_p} q_C(t)\frac{q(t)}{\rho c_p
\]

The measured concentration has a time delay \(t_d=0.5\) min, then \(C_{A,meas}(t) = C_A(t - t_d)\). The objective is to control the measured concentration of A, \(C_A\), by manipulating the coolant flow rate \(q_C\). The nominal parameter values appear in Table 1. This reactor presents two steady states: OR1=(0.10,438.51) and OR2=(0.96,353.54). This is represented in Fig. 2, where both steady states are represented in the state space. Our control objective will be to define two controllers, one for each operating condition.

B. Process Identification

To describe the reactor using a Wiener model, we should obtain a linear description for the dynamic and a nonlinear gain. In our case, we will use a L-N approach. In this way, first the linear block is identified using a correlation technique; after that the intermediate signal \(v\) is generated from the input signal and the static non-linearity is estimated (Westwick and Verhaegen, 1996). This approach is chosen because it is straightforward and ensures an accurate description of the static nonlinearity. To perform the dynamic identification a Pseudo Random signal with a sample time of 0.1 minute was applied to the process. Then, using the input/output data a model for each operating point was computed using an ARX algorithm. The models obtained are as follows,

\[
H_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 2.86 & 1 & 0 & 0 \\ -3.1249 & 0 & 0 & -0.6655 \times 10^{-5} \\ 1.5423 & 0 & 0 & 0.2852 \times 10^{-5} \\ -0.2914 & 0 & 0 & -0.0539 \times 10^{-5} \\ 1 & 0 & 0 & -1.8492 \times 10^{-4} \end{bmatrix}
\]

\[
H_2 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 2.6756 & 1 & 0 & 0 \\ -2.4051 & 0 & 1 & -0.1536 \times 10^{-5} \\ 0.7442 & 0 & 0 & 0.0356 \times 10^{-5} \\ -0.0161 & 0 & 0 & -0.0008 \times 10^{-5} \\ 1 & 0 & 0 & 0 \end{bmatrix}
\]

Note that in this identified model the time delay is zero because the process time delay of 0.5 min is small compared to the pseudo-time constant of the system. This model results from a identification process, then the parameters were automatically adopted for the algorithm to achieve the best data adjustment.

Once that the linear model is obtained, several steady-state experiments were carry out and a PWL model is obtained using the PWL Toolbox proposed by Julian (1999), based on the Least Squares method. Using the same tool, a PWL description of the inverse of the gain is obtained. The static gain and the comparison from data and model output are shown in Fig. 3 and Fig. 4, respectively.
In this particular application, the dimension of the domain for the nonlinear gain is one. In this case, the PWL functions have the following particularities (see Appendix): the domains for each operative region are \( D_1 = [-0.095, 0.23] \) and \( D_2 = [0.96, 0.97] \). These domains are divided in 10 sectors defined as 10 segments of equal length. Let us call as \( \beta_i \) (for \( i = 1, \ldots, 10 \)) the limits of each segment.

Then, the basis \( \Lambda \) for each domain are defined as
\[
\Lambda(v) = \begin{bmatrix}
v - \beta_1 & v - \beta_2 & \ldots & v - \beta_9 & v - \beta_{10}
\end{bmatrix}
\]
and the vectors of parameters, computed using the PWL Toolbox are:
\[
c_1 = \begin{bmatrix}
0.958 & 2.5599 & -0.6078 & -0.3905 & -0.2787 & -0.2266 & -0.1342 & -0.1419 & -0.1042 & -0.0729 & -0.0643
\end{bmatrix}
\]
and
\[
c_2 = \begin{bmatrix}
0.0152 & 0.1931 & 0.0768 & 0.1036 & 0.1607 & 0.2331 & 0.6483 & 2.1016 & -2.8079 & 0.5079 & 0.1171
\end{bmatrix}
\]
for operating regions 1 and 2, respectively.

### C. Nonlinear Model Predictive Control

The control problem described in (5) was connected to the simulation model of the reactor in both operative regions. The weighing matrices and the bounds on the variables are described in Table 2.

#### Table 2: Model Predictive Control Parameters

<table>
<thead>
<tr>
<th></th>
<th>OR1</th>
<th>OR2</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Q - R )</td>
<td>100 I - 10 I</td>
<td>100 I - 0.001 I</td>
</tr>
<tr>
<td>( P - M )</td>
<td>200 - 10</td>
<td>200 - 10</td>
</tr>
<tr>
<td>( y_t - y_u )</td>
<td>0 - 0.2</td>
<td>0.9 - 0.968</td>
</tr>
<tr>
<td>( u_t - u_u )</td>
<td>60 - 120</td>
<td>50 - 200</td>
</tr>
</tbody>
</table>

### IV. CONCLUSIONS

In this paper a Nonlinear Model Predictive Control based on a Wiener Model with a Piecewise Lineal gain is applied to control a CSTR with two steady states. The performance of the controller to retain the process in each operative region is tested by simulation.
A. LUSSON CERVANTES, O. E. AGAMENNONI, J. L. FIGUEROA


APPENDIX: PIECEWISE LINEAR FUNCTIONS

The Piecewise Linear (PWL) Functions are used since 1965 in the area of nonlinear circuit theory. In the 70's they take special relevancy, from the works of Leon Chua (1971) in which the treatment of systems in $\mathbb{R}^2$ is properly solved. But only recently with the works of Julian et al. (1999) on High Level (HL) PWL, it is possible to have an adequate expression to solve general problems in $\mathbb{R}^m$. This representation is able to uniformly approximate any Lipschitz continuous function defined on a compact domain. In general, a PWL function is defined as follows:

**Definition 1 (PWL function):** A function $f : D \subseteq \mathbb{R}^m \rightarrow \mathbb{R}^n$, where $D$ is a compact set, is a PWL function if and only if it satisfies,

1. The domain $D$ is divided into a finite number of polyhedral regions $R^{(1)}, R^{(2)}, \ldots, R^{(q)}$ (such that $D = \bigcup_{i=1}^{q} R^{(i)}$) by a finite set of boundaries, $H = \{H_i \subseteq D, i = 1, 2, \ldots, q\}$, such that each boundary is either an $(m-1)$-dimensional hyperplane characterized by $H_i = \{x \in \mathbb{R}^m : \pi_i(x) = \alpha_i x + \beta_i = 0\}$, where $\alpha_i \in \mathbb{R}^m$, $\beta_i \in \mathbb{R}$ for $i = 1, 2, \ldots, q$, or a subset of it, and cannot be covered by any $(m-2)$-dimensional hyperplane. 

REFERENCES


2. \( f \) is expressed by an affine representation of the form \( f^{(i)}(x) = J^{(i)}w + w^{(i)} \), for any \( x \in R^{(i)} \), where \( J^{(i)} \in \mathbb{R}^{m \times n} \) is the Jacobian matrix of \( f \) in the region \( R^{(i)} \) and \( w^{(i)} \in \mathbb{R}^n \).

3. \( f \) is continuous on any boundary between two regions, namely, \( J^{(i)}x + w^{(i)} = J^{(q)}x + w^{(q)} \), for any \( x \in R^{(i)} \cap R^{(q)} \), \( p \neq q \).

Julian (1999) formulates the canonical expression for the family of PWL functions defined over a simplicial partition of the domain \( D \subset \mathbb{R}^n \), where \( m \) is a positive integer. This type of partition subdivides the domain \( D \) in a simplex set \( \mathcal{S}^{(i)} \), \( i = 1, \ldots, q \) such that \( D = \bigcup_{i=1}^{q} \mathcal{S}^{(i)} \). The corresponding set of vertexes for these simplexes is called \( \mathcal{S}^{(i)} \).

1. The function values assigned to each vertex defines a unique (and local) linear affine function for each simplex;
2. The local lineal expressions define a PWL continuous function because they are continuous on the boundaries of the partition.

The extension of this idea to an \( m \) dimensional domain leads us to define simplexes of \( m+1 \) vertices. Then, if we associate one function value to each vertex, it is possible to determine a unique linear affine (local) function for each simplex. In this way, a continuous PWL function is determined by the collection of all the local functions. From this procedure, it is clear that any arbitrary PWL function \( f : D \rightarrow \mathbb{R}^n \) defined over the simplicial boundary configuration introduced is uniquely determined by its values on the vertices.

Without loss of generality, in the following, it is assumed that the dimension of the image set is one (a PWL map from \( \mathbb{R}^n \rightarrow \mathbb{R}^n \) is equivalent to \( m \) independent PWL maps \( f_1, \ldots, f_m \) from \( \mathbb{R}^n \rightarrow \mathbb{R}^1 \) with the same boundary configuration).

**Definition 2:** Consider a compact domain \( D \rightarrow \mathbb{R}^n \) and a set of hyperplanes \( H \). Then, \( PWL_{(D)}[D] \) is defined as the set of all the PWL continuous mappings taking values on the domain \( D \) partitioned with the boundary configuration \( H \).

Julian (1999) proved that a proper definition of addition and multiplication by a scalar over the set \( PWL_{(D)}[D] \) leads to a linear vector space (LVS). Moreover, a set of HL CPWL functions which are a basis of \( PWL_{(D)}[D] \) was found as a function of \( k \) nesting absolute values.

In addition, the elements of this base can be expressed in vector form as \( \Lambda = \left[ \Lambda^T, \Lambda'^T, \ldots, \Lambda'^{\ell_T} \right] \), ordered according to its nesting level (n.l.), where \( \Lambda' \) is the vector containing the n.l.\( = i \) functions. Accordingly, the main result in Julian (1999) said that any function \( f \in PWL_{(D)}[D] \) can be written as

\[
 f(x) = c^T \Lambda(x)
\]

where \( c = [c_0^T, c_1^T, \ldots, c_{\ell}^T] \) and every vector \( c_i \) is a parameter vector associated to the n.l.\( = i \) vector function \( \Lambda' \).

Received: September 16, 2001.
Accepted for publication: November 02, 2002.
Recommended by Guest Editors J. Cerdá, S. Díaz and A. Bandoni.