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
Avda. Alem 1253, 8000 Bahía Blanca. Argentina.
† Also in Planta Piloto de Ingenieria Quimica - PLAPIQUI (UNS-CONICET)
Camino La Carrindanga Km. 7 (8000) Bahia 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:

$$\begin{aligned} x(k+1) &= g\left(x(k),\Delta u(k)\right) \\ y(k) &= h\left(x(k)\right)+d(k) \end{aligned} \tag{1}$$

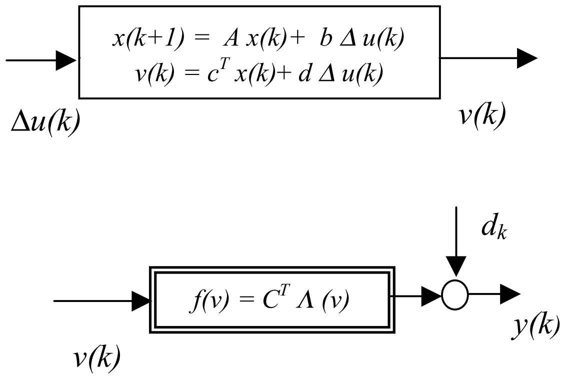
$x(k+1) = A\,x(k)+ b\,\Delta\,u(k)$
$v(k) = c^T x(k)+ d\,\Delta\,u(k)$
$\Delta u(k)$
$v(k)$
d_k
$f(v) = C^T \Lambda\,(v)$
$y(k)$
$v(k)$

Figure 1: The Wiener model for MPC

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 Λ as $f(v)=C^T\Lambda(v)$. 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 $v=f^{-1}(f(v))$. This function is also unique and it is a PWL.

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

$$J=\sum_{j=1}^{p}\left\|y(k+j)-r\right\|_{Q_j}+\sum_{j=0}^{m-1}\left\|\Delta u(k+j)\right\|_{R_j} \tag{2}$$

subject to model equations and to inequality constraints

$$\begin{aligned} y^l \le y(k+j)\le y^u \qquad & \forall j=1,..,p-1 \\ u^l \le u(k+j)\le u^u \qquad & \forall j=1,..,m-1 \end{aligned} \tag{3}$$

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.* $\Delta u(k+j)=0$, $j=m,\dots,p$). 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 ($d(k)=d(k+j)$, $j=1,..,m$). To accomplish this, a bias term that compares the current predicted output to the current measured is computed (*i.e.,* $d(k)=y(k)-y^m(k)$).

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 $\boldsymbol{v}(k)$, $\boldsymbol{\Delta U}(k)$, $\boldsymbol{y}(k)$ and $\boldsymbol{r}(k)$; then, the predicted output for the linear model is

$$\hat{\boldsymbol{v}}(k)=\boldsymbol{\beta}\,\boldsymbol{\Delta U}(k)+\boldsymbol{\xi}\,x(k) \tag{4}$$

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

$$\boldsymbol{\beta}=\begin{bmatrix} c^Tb+d & 0 & \cdots & 0 \\ c^TAb & c^Tb+d & \cdots & 0 \\ \vdots & \vdots & \ddots & c^Tb+d \\ \vdots & \vdots & \cdots & \vdots \\ c^TA^{p-1}b & c^TA^{p-2}b & \cdots & c^TA^{p-m}b \end{bmatrix} \quad \boldsymbol{\xi}=\begin{bmatrix} c^T \\ c^TA \\ \vdots \\ c^TA^{p-1} \end{bmatrix}$$

and the predicted output for the complete model is

$$\hat{\boldsymbol{y}}(k)=\boldsymbol{f}(\hat{\boldsymbol{v}}(k))=\left[\boldsymbol{f}(\boldsymbol{v}(k+1))^T\cdots\boldsymbol{f}(\boldsymbol{v}(k+p))^T\right]^T$$

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

- Since the PWL function $\boldsymbol{f}$ 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^u and y^l are the upper and lower bounds defined in (3), then, we can translate this magnitudes to the linear model as $v^u=f^{-1}(y^u)$ and $v^l=f^{-1}(y^l)$, respectively.

Finally, the WMPC can be posed as

$$\begin{aligned} &\min_{U} J=\left(\hat{\boldsymbol{v}}(k)-\boldsymbol{r}^*\right)^T\boldsymbol{Q}\left(\hat{\boldsymbol{v}}(k)-\boldsymbol{r}^*\right)+\ \boldsymbol{U}^T\boldsymbol{R}\ \boldsymbol{U} \\ &\text{s.t.} \\ &\hat{\boldsymbol{v}}(k)=\quad \boldsymbol{U}+\qquad x(k) \\ &v^l\le\hat{\boldsymbol{v}}(k)\le v^u \\ &u^l\le u(k)\le u^u \end{aligned} \tag{5}$$

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.

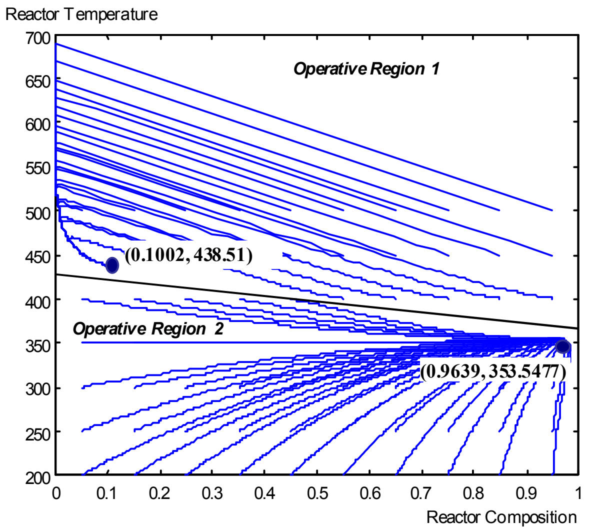

Figure 2: Steady State Space

Table 1. Nominal CSTR parameter values.

Product concentration	C_A	0.1 mol / l
Reactor temperature	T	438.54 K
Coolant flow rate	Q_c	103.41 l/min
Process flow rate	Q	100 l/min
Feed concentration	C_{A0}	1 mol / l
Feed temperature	T_0	350 K
Inlet coolant temperature	T_{c0}	350 K
CSTR volume	V	100 l
Heat transfer term	HA	7 x10^5 cal/min K
Reaction rate constant	K_0	7.2 x 10^{10} 1/min
Activation energy term	E/R	1 x 10^4 K
Heat of reaction	ΔH	-2 x 10^5 cal/mol
Liquid densities	ρ, ρ_c	1 x 10^3 g/l
Specific heats	C_p, C_{pc}	1 cal/g K

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).

$$\dot{C}_A(t) = \left({q(t)}\big/{V}\right)\left(C_{A0}(t) - C_A(t)\right) - \; k_0 C_A(t)\exp\left(-{E}\big/{RT(t)}\right)$$

$$\dot{T}(t) = \left({q(t)}\big/{V}\right)\left(T_0(t) - T(t)\right) - \left({k_0 \Delta H}\big/{\rho C_p}\right) C_A(t).$$
$$\exp\left(-{E}\big/{RT(t)}\right) + \left({\rho_c C_{pc}}\big/{\rho C_p V}\right) q_c(t).$$
$$\left[1 - \exp\left(-{hA}\big/{q_c(t)\rho_c C_{pc}}\right)\right]\left(T_0(t) - T(t)\right)$$

The measured concentration has a time delay t_d=0.5min, 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: OR_1=(0.10,438.51) and OR_2=(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,

$$H_1 = \left[\begin{array}{c|c} A & b \\ \hline c & d \end{array}\right] = \left[\begin{array}{cccc|c} 2.86 & 1 & 0 & 0 & 0.4837\times10^{-3} \\ -3.1249 & 0 & 1 & 0 & -0.6655\times10^{-3} \\ 1.5423 & 0 & 0 & 1 & 0.2852\times10^{-3} \\ -0.2914 & 0 & 0 & 0 & -0.0539\times10^{-3} \\ \hline 1 & 0 & 0 & 0 & -1.8492\times10^{-4} \end{array}\right]$$

$$H_2 = \left[\begin{array}{c|c} A & b \\ \hline c & d \end{array}\right] = \left[\begin{array}{cccc|c} 2.6756 & 1 & 0 & 0 & 0.1284\times10^{-5} \\ -2.4051 & 0 & 1 & 0 & -0.1536\times10^{-5} \\ 0.7442 & 0 & 0 & 1 & 0.0356\times10^{-5} \\ -0.0161 & 0 & 0 & 0 & -0.0008\times10^{-5} \\ \hline 1 & 0 & 0 & 0 & 0 \end{array}\right]$$

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.

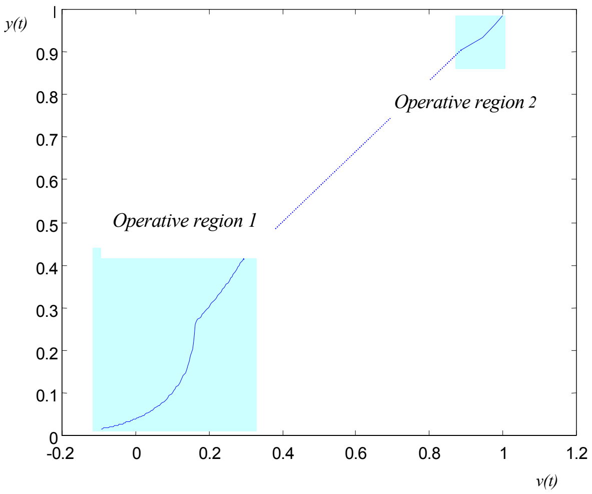

Figure 3. PWL static Gain.

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 β_i (for i=1,..,10) the limits of each segment.

Then, the basis Λ for each domain are defined as

$$\Lambda(v) = \left[1 \quad v-\beta_1 - |v-\beta_1| \quad \cdots \quad v-\beta_{10} - |v-\beta_{10}|\right]$$

and the vectors of parameters, computed using the PWL Toolbox are

$$c_1 = [.958 \quad 2.5599 \quad -.6078 \quad -.3905 \quad -.2787 \quad \cdots$$
$$-.2266 \quad -.1342 \quad -.1419 \quad -.1042 \quad -.0729 \quad -.0643]$$

and

$$c_2 = [.0152 \quad .1931 \quad .0768 \quad .1036 \quad .1607 \quad \cdots$$
$$.2331 \quad .6483 \quad 2.1016 \quad -2.8079 \quad .5079 \quad .1171]$$

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

	OR1	OR2
$Q-R$	100 I - 10 I	100 I- 0.001 I
$P-M$	200 – 10	200 – 10
$y_l - y_u$	0 – 0.2	0.9 – 0.968
$u_l - u_u$	60 – 120	50 – 200

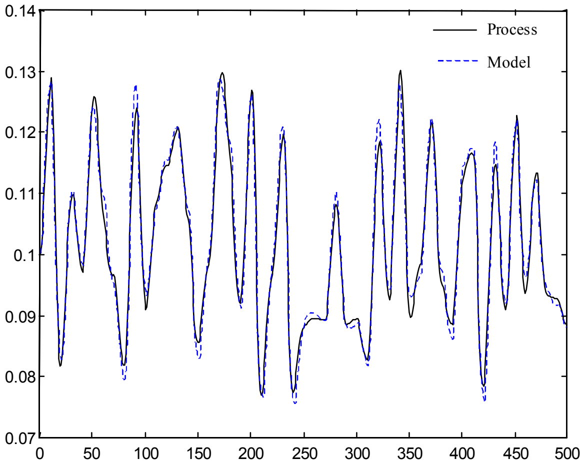

Figure 4. Performance of the Wiener model.

The values Q and R were evaluated through simulation, trying to solve the trade off between smooth control action and fast response. The constraints were defined to guarantee that the operating region of the reactor is always inside the identified regions. The prediction horizon p is chosen to capture the complete process behavior (*i.e.*, to allow the model reach the steady-state).

Note that it is necessary to change the parameter R when the reactor moves from one operating point to the other. This is necessary due to the large difference of gain in each case. The simulation results are depicted in Fig. 5 and 6 for operating conditions 1 and 2, respectively. The response of the NMPC is compared to the response of the linear controllers having the same set of parameters. This has been done because the linearization effects of the NMPC controller are to be remarked. Fig. 5 and 6 illustrate how the closed loop system with NMPC controller achieves a similar time constant for upper and lower step inputs. The LMPC step response to a lower set point (at OP1) is considerably slower because the system is driven to a lower gain region (see Fig. 3). On the other hand (at the same operating point), the behavior of the LMPC is faster than the NMPC for an upper step input because the system is driven to a higher gain region. From Fig. 5 and 6 it is clear that the NMPC response with the same time constant in both operating points and without overshoot. This behavior does not occur with the LMPC scheme.

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.

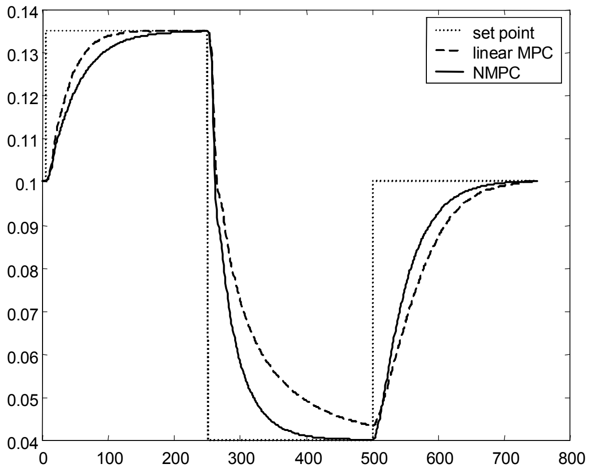

Figure 5. Closed loop response for set point changes OP1.

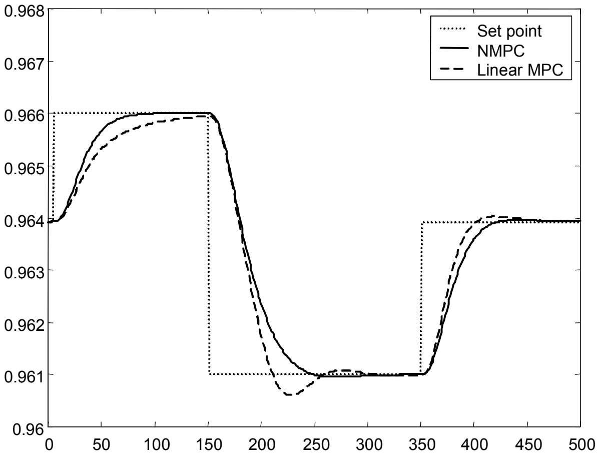

Figure 6. Closed loop response for set point changes OP2.

REFERENCES

Bequette, B.W. Nonlinear control of chemical processes: A review. *Ind. Eng. Chem. Res.,* **30**, 1391-1413 (1991).

Chua, L.O. Efficient computer algorithms for piecewise-linear analysis of resistive nonlinear networks. IEEE Trans. Circuits Theory, **CT-18**, 73-85 (1971).

Giovanini L. Control Predictivo Robusto con Restricciones, *PhD Thesis*. Universidad Nacional del Litoral (2000).

Henson, M. Nonlinear model predictive control: current status and future directions. *Computers and Chemical Engineering*, **23**, 187 (1998).

Julian, P. High Level Canonical Piecewise Linear Representation: Theory and Applications. *PhD thesis in Systems Control*, Universidad Nacional del Sur (1999).

Lusson Cervantes, A., O.E. Agamennoni and J.L. Figueroa, A Wiener Nonlinear Model Predictive Control under Uncertainties", *Second Pan American Workshop on Process Systemms Engineering,* Guaruja SP, Brasil, September (2001).

Morningred, J.D., B.E. Paden and D.A. Mellichamp, An adaptive nonlinear predictive controller. *Chemical Engineering Science*, **46**, 755 (1992).

Norquay, S., Palazoglu, A., Romagnoli, J.A. Model predictive control based on wiener models. *Chemical Engineering Science*, **53**, 75. (1998).

Norquay, S.J., A. Palazoglu and J.A. Romagnoli, Application of Wiener Model Predictive Control to an Industrial C2-Splitter, *J. of Process Control*, **9**, 461-473 (1999).

Qin, S.J., Badgwell, T.A. An overview of industrial model predictive control technology, *AIChE Symposium Series*, **93**, 232. (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).

Westwick, A.D. and M. Verhaegen, Identifying MIMO wiener systems using subspace model identification methods. *Signal Processing*, **52**, 235 (1996).

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: 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 $\Re^2$ 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 $\Re^m$. 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 $f: D \subset \Re^m \rightarrow \Re^m$, 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 $R^{(1)}$, $R^{(2)}$, ..., $R^{(q)}$ (such that $D = \bigcup_{i=1}^{q} R^{(i)}$) by a finite set of boundaries, $H = \{H_i \subset D, i = 1,.2.,...,h\}$, such that each boundary is either an (m-1)-dimensional hyperplane characterized by $H_i = \{x \in \Re^m : \pi_i(x) = \alpha_i^T x - \beta_i = 0\}$, where $\alpha_i \in \Re^m$, $\beta_i \in \Re$ for i=1:,2,...,h, or a subset of it, and cannot be covered by any (m-2)-dimensional hyperplane.

2. f is expressed by an affine representation of the form $f^{(i)}(x) := J^{(i)}x + w^{(i)}$, for any $x \in R^{(i)}$, where $J^{(i)} \in \Re^{m\times m}$ is the Jacobian matrix of f in the region $R^{(i)}$ and $w^{(i)} \in \Re^{m}$.
3. f is continuous on any boundary between two regions, namely, $J^{(p)}x + w^{(p)} = J^{(q)}x + w^{(q)}$, for any $x \in R^{(p)} \cap R^{(q)}, p \neq q$.

Julian (1999) formulates the canonical expression for the family of PWL functions defined over of a simplicial partition of the domain $D=\{(x_1,...,x_n):0 \leq x_i \leq m_i\delta, i \in \{1,...,m\}\} \subset \Re^m$, where δ is the grid size and m_i is a positive integer. This type of partition subdivides the domain D in a simplex set $\overline{S}^{(i)}$, i=1,…,q such that $D = \bigcup_{i=1}^{q} \overline{S}^{(i)}$. The corresponding set of vertexes for these simplexes is called $S^{(i)}$. Let us suppose, for simplicity, that the domain D belongs to $\Re^2$ (see Fig. A1 for a description of the partition in $\Re^2$). If one function value is associated to each vertex (Fig. A2), then, it is possible to define a PWL function with the following characteristics:

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 $f: D \rightarrow \Re^1$ 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 $\Re^m \rightarrow \Re^m$ is equivalent to m independent PWL maps f_1, …, f_m from $\Re^m \rightarrow \Re^1$ with the same boundary configuration).

Definition 2: Consider a compact domain $D \rightarrow \Re^m$ and a set of hiperplanes H. Then, $PWL_H[D]$ is defined as the set of all the PWL continuos 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 $PWL_H[D]$ leads to a *linear vector space* (*LVS*). Moreover, a set of HL CPWL functions which are a basis of $PWL_H[D]$ was found as a function of k nesting absolute values.

In addition, the elements of this base can be expressed in vector form as $\Lambda = \left[\Lambda^{0^T}, \Lambda^{1^T}, ...\Lambda^{m^T}\right]^T$, ordered according to its *nesting level* (*n.l.*), where Λ^i is the vector containing the *n.l.=i* functions. Accordingly, the main result in Julian (1999) said that any function $f \in PWL_H[D]$ can be written as

$$f(x) = c^T \Lambda(x)$$

where $c = \left[c_0^T, c_1^T, ...c_m^T\right]^T$ and every vector c_i is a parameter vector associated to the *n.l.=i* vector function Λ^i.

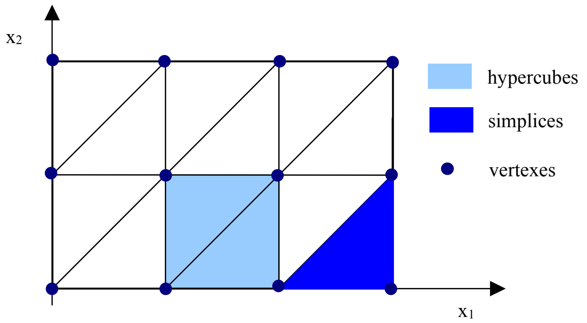

Figure A.1.: Simplicial Partition in $\Re^2$

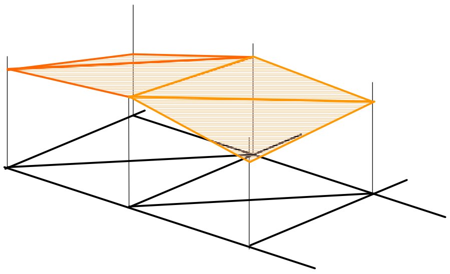

Figure A2. Constructive Approach in $\Re^2$

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