A NOE MODEL IDENTIFICATION ALGORITHM USING PIECEWISE LINEAR FUNCTIONS

Liliana R. Castro[1], Marcela P. Álvarez[1], Osvaldo E. Agamennoni[2]

[1]Universidad Nacional del Sur, Depto. de Matemática
Av. Alem 1253, B8000CPB Bahía Blanca, Argentina
[2]Universidad Nacional del Sur, Depto. de Ingeniería Eléctrica y de Computadoras and CIC
Av. Alem 1253, B8000CPB Bahía Blanca, Argentina

lcastro@uns.edu.ar(Liliana R. Castro)

Abstract

In this paper we propose a Nonlinear Output Error (NOE) identification algorithm based on High Level Canonical Piecewise Linear (HL CPWL) functions. Starting from a linear Output Error (OE) model, the proposed model structure allows the implementation of an identification algorithm in which the degrees of freedom (flexibility) of the model can be easily increased during the identification process, retaining the achieved approximation. This is done by increasing the number of divisions of the HL CPWL simplicial domain until obtaining a given approximation error. The parameters of the HL CPWL functions are updated using a simple algorithm based on a modified steepest descent method with an adaptive learning rate that also allows controlling de BIBO stability of the model. Taking into account the simplicity of the HL CPWL VLSI realization, we are interested in the hardware implementation of the identification algorithm. We also derive sufficient conditions for BIBO stability of the identification algorithm. Taking into account this condition, we derive minimum and maximum bounds that preserve BIBO stability of the model during the optimization of the parameters of the HL CPWL functions. This model structure is well suited for control applications that need a large simulation horizon. This is the case of different optimal control applications like, for example, Model Predictive Control (MPC).

Keywords: Nonlinear systems, identification, piecewise linear techniques.

Presenting Author's Biography

Liliana Castro received a bachelor and a Master in Mathematics and the Doctorate in Control Systems at the Universidad Nacional del Sur (UNS), Argentina. Since 1999 she is Associate Professor at the Departamento de Matemática, UNS and she is also with the Instituto de Investigaciones en Ingeniería Eléctrica "Alfredo Desages". Since 2000 she is head of the Modeling Group at the Laboratorio de Investicación y Desarrollo en Visualización y Computación Gráfica, Departamento de Ciencias e Ingeniería de la Computación, UNS. Her research interests include nonlinear system modeling, identification, piecewise linear approximation of nonlinear systems and geometric modeling using wavelets.

ISBN 978-3-901608-32-2

1 Introduction

In the present paper we are focused on developing a nonlinear model structure together with the associated parameter identification algorithm, oriented to applications that need a large simulation horizon. This is the case in different optimal control applications, like Model Predictive Control (MPC).

One of the main problems in system identification is to find a suitable model structure to describe the process. This problem becomes more complicated if this structure is allowed to go from a linear to a nonlinear, since the set of nonlinear models is much richer than the set of linear ones [1]. If a nonlinear finite impulse (NFIR) structure is used, the model order evaluation problem may be effectively addressed by using regularization theory [2]. If a Wiener like model structure is used, an aggregation approach can be easily implemented as in the Korenberg algorithm [3]. With the advent of Neural Networks and Fuzzy models a much wider class of systems can be handled [4]. In the Neural Networks literature, growing and pruning methods are used to deal with the size of a Neural Network during the training process [5]. Piecewise affine linear models have been widely used for nonlinear system identification (for example, [6, 7, 8]) as an interesting class of parametrized NARMAX models [9, 10, 11, 12]. These models are a special case of multiple model approaches (see, for example, [13, 14] for a survey on this field).

When NOE model structures are used, the problem becomes much more difficult due to the computational complexity involved in the recursive nature of the model (see, for example, [4]). In addition, model order variation (increase or decrease) in the NOE structure identification algorithms in the literature, is implemented restarting the process. There is no straightforward procedure to reduce the computational cost of evaluating the new set of parameters from the previous one. In this paper we consider a NOE model structure like the one proposed by Narendra and Parthasarathy [15] in the context of Neural Networks based on High Level Canonical Piecewise Linear (HL CPWL) functions [16, 17]. We develop an identification algorithm that offers a simple mechanism for increasing the model approximation capabilities (increase the model order) retaining the previously achieved approximation. This algorithm can be also modified in order to reduce the model order. In this way, it is possible to start the identification with a linear OE approximation and then progressively increase the model approximation capabilities to reduce the model mismatch up to an acceptable value. Finally, we derive simple sufficient conditions that ensures the BIBO stability of the identification algorithm. Taking into account this condition, one of the contributions of this paper is to derive minimum and maximum bounds that preserve BIBO stability of the model during the parameter optimization. This work is part of an ongoing project focused on the development of an identification oriented hardware. The VLSI realization properties of the HL CPWL functions [18] are specially suited for this purpose. This is the reason why, in a first approach, we are using simple steepest descent optimization algorithms.

The paper is organized as follows. In Sections 2 and 3 we present the model and the identification algorithm, analyze its advantages and drawbacks; in Section 4 we define and give sufficient conditions for the identification algorithm to be BIBO stable; in Section 5 we develop two different examples using the proposed methodology and in Section 6, we draw some conclusions and outline future works. Finally, in Appendix 7 we present the algorithm to find the set of parameters of the HL CPWL approximation corresponding to a given number of divisions of the simplicial partition of the domain from the set of parameters of the HL CPWL approximation corresponding to the previous number of divisions of the simplicial partition of the domain.

2 Identification structure

Let $(\mathbf{u}, \mathbf{y})$ the input/output vectors corresponding to a given Lipschitz continuous, SISO system. If $\tilde{\mathbf{y}}$ is the estimated value corresponding to the input $\mathbf{u}$, and we note

$$\begin{aligned} \mathbf{u}^{k,M+1} &= [u_k, \ \ldots, u_{k-M}] \\ \tilde{\mathbf{y}}^{k-1,N} &= [\tilde{y}_{k-1}, \ldots, \tilde{y}_{k-N}], \end{aligned}$$

then we propose the following black-box identification structure

$$\begin{aligned} \tilde{y}_k &= f_{pwl}\left(u_k, \ldots, u_{k-M}, \tilde{y}_{k-1}, \ldots, \tilde{y}_{k-N}\right) \\ &= \mathbf{c}\Lambda\left(\mathbf{u}^{k,M+1}, \tilde{\mathbf{y}}^{k-1,N}\right), \end{aligned} \quad (1)$$

where $f_{pwl}(\mathbf{x}) = \mathbf{c}\Lambda(\mathbf{x})$ is the HL CPWL function as defined in [16, 17]. From Eq. (1), the regression vector of the proposed black-box identification structure is $\varphi_k = \left[\mathbf{u}^{k,M+1}, \tilde{\mathbf{y}}^{k-1,N}\right]$. It is worth to mention that a linear OE model is a particular case of f_{pwl} that will be noted as f_{lin}. The model given by Eq. (1) is pictured in Fig. 1.

Assumption 2.1 *The model orders M and N are given. Also, throughout the paper it is assumed that the number of input/output available data is L.*

Assumption 2.2 *The domain of the function f_{pwl} is a compact set $\mathbf{S} \subset \mathbb{R}^m, m = M + N + 1$, defined as follows*

$$\begin{aligned} \mathbf{S} &= \{\mathbf{x} \in \mathbb{R}^m : a_i \leq x_i \leq a_i + \delta.ndiv, \\ &\quad i = 1, 2, \ldots, m\}, \end{aligned} \quad (2)$$

δ being the fixed grid size.

As a consequence of Eq. (2), each dimension is divided into a number of subintervals of equal length δ. Then, when the grid size δ decreases, the number of divisions $ndiv$ on each direction increases. As a consequence, using HL CPWL functions for the nonlinear approximation, $ndiv$ allows to go from a linear model ($ndiv = 1$) to a nonlinear one with a coarse to a finer

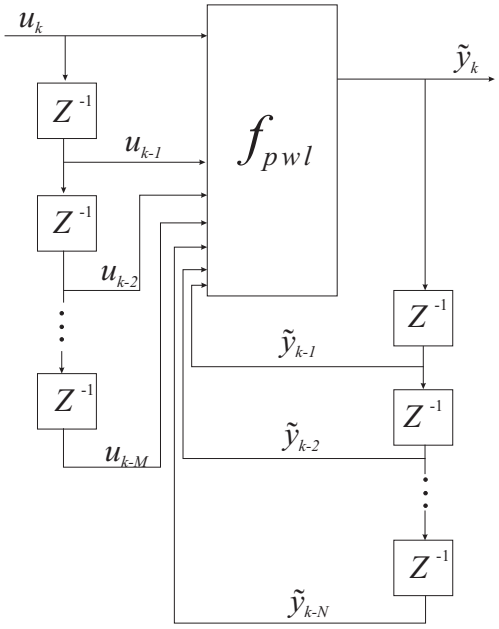


Fig. 1 HL CPWL NOE model.

partition of $\mathbf{S}$. The advantages of using this kind of models is pointed out in [1, Ch. 1].

The set defined by Eq. (2) is partitioned into polyhedrical regions using a simplicial boundary configuration [16, 17]. The f_{pwl} constructed using the methodology described in [16, 17] is linear on each simplex and continuous on the adjacent boundaries of the simplices. Besides, the HL CPWL functions uniformly approximate any Lipschitz continuous function in $\mathbf{S}$ [16, 17].

3 Identification algorithm

With the HL CPWL model structure presented in the previous section, it is possible to develop a nonlinear identification algorithm that starts the identification process with a linear approximation model (a special case of a HL CPWL fuction). Then it is possible to increase the number of divisions $ndiv$ of the simplicial domain partition and straightforwardly evaluate the new set of parameters (see Section 7) corresponding to the achieved approximation function but with a finer division of the input domain partition. Afterwards, update the vector of parameters using a suitable optimization algorithm until a minimum is reached. This three steps process (grid subdivision, parameter update according to the new grid subdivision and parameter optimization) can be repeated until a desired model error is achieved. Similarly, it is possible to go from a fine approximation to a coarser one by decreasing the value of $ndiv$. We will now formally present this idea.

Let M, N the orders of the model (see Assumption 2.1) and $\mathbf{S}$ the compact domain that contains the complete set of data $(u_k, y_k)_{1\leq k\leq L}$. Next we set the notation that will be used in the algorithm.

Notation

$ndiv = 2^d, d \geq 0$: number of divisions of the region $\mathbf{S}$. Equal number of divisions in each dimension is assumed.

V^d: the set of vertices of the simplicial partition H of the set $\mathbf{S}$ with $ndiv = 2^d$ number of divisions.

Λ^d: The HL CPWL basis defined on $\mathbf{S}$ with vertices belonging to V^d.

$\mathbf{c}^{d,*}$: the vector of parameters associated with best HL CPWL approximations using the basis Λ^d. From [16, 17] it can be easily concluded that the number of parameters is $(ndiv+1)^{M+1+N}$.

$(A)_i$: the i-th row of a matrix A.

$Niter$, $Niter \in \mathbb{N}$: maximum number of iterations of the optimization algorithm.

$Maxerror$: maximum allowable approximation error.

$\mathbf{lr}^r$: learning rate corresponding to iteration r; $lr_i^r > 0\,\forall i$ (typically, $lr_i = 0.0001$).

mom: momentum, $mom > 0$ (typically, $mom = 0.9$).

lr_{inc}: learning rate increment, $lr_{inc} > 1$ (typically, $lr_{inc} = 1.05$).

lr_{dec}: learning rate decrement, $0 < lr_{dec} < 1$ (typically, $lr_{dec} = 0.9$).

Algorithm

Step 1. $d = 0$*: Linear Approximation.*

Evaluate a linear OE model approximation. Then compute the parameters $\mathbf{c}^{d,*}$ of the HL CPWL representation from the hyperplane defined by the OE parameters. This is a straightforward process since an hyperplane is a particularly simple case of a HL CPWL function [16, 17].

Step 2. $d \leftarrow d+1$*: Evaluation of* $\mathbf{c}^{d,*}$ *from* $\mathbf{c}^{d-1,*}$.

Evaluate $\mathbf{c}^{d,*}$ from $\mathbf{c}^{d-1,*}$ according to the algorithm described in Section 7; set

$$r = 0, \qquad \mathbf{c}^{d,r} = \mathbf{c}^{d,*}, \qquad \Delta\mathbf{c}^{d,r} = [0,\ldots,0].$$

Step 3. $r \leftarrow r+1$*: Error and gradient evaluation.*

$$E^r = \frac{1}{2}\sum_{i=1}^{L}\left[y_i - \mathbf{c}^{d,r-1}\Lambda^d\left(\mathbf{u}^{i,M+1}, \tilde{\mathbf{y}}^{i-1,N}\right)\right]^2, \tag{3}$$

$$\begin{aligned}\left(\nabla E^r\right)_j &= \frac{\partial E^r}{\partial\left(c_j^{d,r-1}\right)} \\ &= -\sum_{i=1}^{L}\left[y_i - \mathbf{c}^{d,r-1}\Lambda^d\left(\mathbf{u}^{i,M+1}, \tilde{\mathbf{y}}^{i-1,N}\right)\right] \\ &\qquad \left(\Lambda^d\left(\mathbf{u}^{i,M+1}, \tilde{\mathbf{y}}^{i-1,N}\right)\right)_j . \end{aligned} \tag{4}$$

Step 4. Parameter update.

If $E^r \leq Maxerror$ then STOP, otherwise,

$$\Delta c_j^{d,r} = \eta\left(-\nabla E_j^r lr_j^r + \Delta c_j^{d,r-1} mom\right), \quad (5)$$

$$\mathbf{c}^{d,r} = \mathbf{c}^{d,r-1} + \Delta \mathbf{c}^{d,r}, \quad (6)$$

where the constant η lies in the interval defined by Eq. (9) and (10) given in Section 4 and the components of the learning rate vector $\mathbf{lr}^r$ are modified as follows

$$lr_j^r = \begin{cases} lr_j^{r-1} \times lr_{inc} & \text{if sign}\left(\nabla E_j^r\right) = \text{sign}\left(\nabla E_j^{r-1}\right) \\ lr_j^{r-1} \times lr_{dec} & \text{if sign}\left(\nabla E_j^r\right) \neq \text{sign}\left(\nabla E_j^{r-1}\right) \end{cases}.$$

If $r < Niter$, go to Step 3;

else $\mathbf{c}^{d,*} = \mathbf{c}^{d,r}$ and go to Step 2.

Remark 3.1 *In Step 4, any of the well known stop conditions based on the error evolution may be applied.*

From the formulation, locally convergence of the method to a minimum immediately follows. Like in any optimization process, the drawback is that the achieved minimum may not be a global minimum but a local one. In spite of this, the advantages of using HL CPWL functions enumerated below make it worth to define this identification structure.

1. Computing the gradient is linear in the parameters and straightforward.
2. The canonical HL CPWL approximation uses the least number of parameters.
3. The number of divisions $ndiv$ is progressively increased retaining the achieved approximation and introducing, in the identification process, extra degrees of freedom to improve the model approximation.

4 BIBO stability conditions of the model

Let us suppose that $\mathbf{u} \in U \subset \mathbb{R}^{M+1}$, $\mathbf{y} \in O \subset \mathbb{R}^N$, U and O given compact sets, $Q \subset U \times O$, Q compact and $I = \left[\underline{y}, \overline{y}\right] \subset \mathbb{R}$, with $\underline{y} = \min \mathbf{y}, \overline{y} = \max \mathbf{y}$.

Definition 4.1 *We say that the model defined by (1) is BIBO stable if $f_{pwl}(Q) \subset I$.*

This definition means that the model output remains within the output values when the input is any signal $\mathbf{u} \in U$.

The expression in Eq. (1) defines a mapping $f_{pwl} : Q \to I$. As Q is a compact set and f_{pwl} is continuous on Q, then it attains its maximum and minimum values on Q. Moreover, since f_{pwl} is linear on each simplex, the extreme values are attained on V_Q, the set of vertices of Q. Then the NOE identification structure given by Eq. (1) will be BIBO stable if for every fixed d and any r, the following conditions are simultaneously fulfilled

$$\begin{cases} \min_{\mathbf{v}\in V^d}\left(\mathbf{c}^d \Lambda(\mathbf{v})\right) \geq \underline{y}, \\ \max_{\mathbf{v}\in V^d}\left(\mathbf{c}^d \Lambda(\mathbf{v})\right) \leq \overline{y}, \end{cases} \quad (7)$$

where E^r is given by Eq. (3) and V^d is defined in Section 3.

Let $\mathbf{c}^d$, d fixed, a vector of parameters such that the model is BIBO stable. Then we must guarantee that, for any r, $\tilde{\mathbf{y}}_k = \mathbf{c}^{d,r}\Lambda\left(\mathbf{z}^k\right) \in I, 1 \leq k \leq L$, where $\mathbf{c}^{d,r}$ is obtained from Step 4 of the identification algorithm described in Section 3.

Now we state the following sufficient condition for BIBO stability.

Proposition 4.1 *Let us suppose that, for d and $r-1$ fixed, the model is BIBO stable. Then the model will be BIBO stable for d and r if the following condition is satisfied*

$$\underline{y} - \min_{\mathbf{v}\in V_d}\left(\mathbf{c}^{d,r-1}\Lambda(\mathbf{v})\right) \leq \Delta \mathbf{c}^{d,r}\Lambda(\mathbf{v}) \leq \overline{y} - \max_{\mathbf{v}\in V_d}\left(\mathbf{c}^{d,r-1}\Lambda(\mathbf{v})\right), \quad (8)$$

where $\Delta \mathbf{c}^{d,r}$ is given by (6) and $\mathbf{v} \in V^d$.

Proof: See [19].

Corollary 4.1 *With the hypothesis of Proposition 4.1, let us note $a = \underline{y} - \min_{\mathbf{v}\in V^d}\left(\mathbf{c}^{d,r-1}\Lambda(\mathbf{v})\right)$, $b = \overline{y} - \max_{\mathbf{v}\in V^d}\left(\mathbf{c}^{d,r-1}\Lambda(\mathbf{v})\right)$. Then*

$$\eta \geq \frac{a}{\min_{\mathbf{v}\in V^d}\left[\left(-\nabla E^r \mathbf{lr}^r + \Delta \mathbf{c}^{d,r-1} mom\right)\Lambda(\mathbf{v})\right]}, \quad (9)$$

$$\eta \leq \frac{b}{\max_{\mathbf{v}\in V^d}\left[\left(-\nabla E^r \mathbf{lr}^r + \Delta \mathbf{c}^{d,r-1} mom\right)\Lambda(\mathbf{v})\right]}, \quad (10)$$

simultaneously.

Proof: The proof immediately follows from the proposition and the fact that $\Delta \mathbf{c}^{d,r-1}\Lambda\left(\mathbf{z}^k\right)\Lambda(\mathbf{z}) = \eta\left(-\nabla E^r\right)\Lambda(\mathbf{z})$ attains its extreme values on V^d.

Remark 4.1 *From both bounds for η given by (9) and (10), the only one with practical interest is the least positive one and is the bound used in Step 4 of the identification algorithm.*

5 Examples

5.1 Example 1

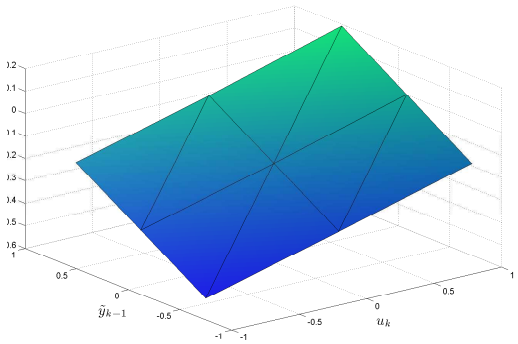
In this example, we consider the well known logistic discrete nonlinear dynamic system with a highly nonlinear exogenous input given by

$$y_{k+1} = 0.4 y_k (1 - y_k) + u_k^7. \quad (11)$$

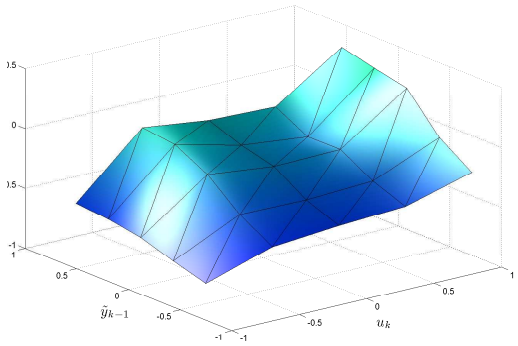
In order to stress the modeling capabilities of the proposed algorithm, we have chosen this highly nonlinear input even though it can not be associated to a real system.

The input $\mathbf{u}$ is defined as a 600 length random signal with uniform distribution between -0.9 and 0.9. According to the proposed methodology, we define the regression vector with $M = 0$ and $N = 1$ as $\varphi_k = [u_k,\ \tilde{y}_{k-1}]$. We have used the first 300 samples for the identification process and the remaining 300 samples for validation.

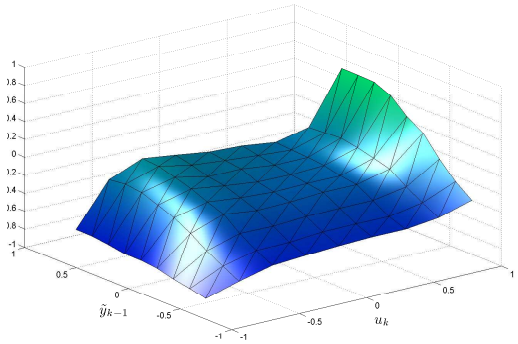
We first evaluate a linear OE model to generate the initial parameters. We start the algorithm described in Section 3 with



(a)



(b)

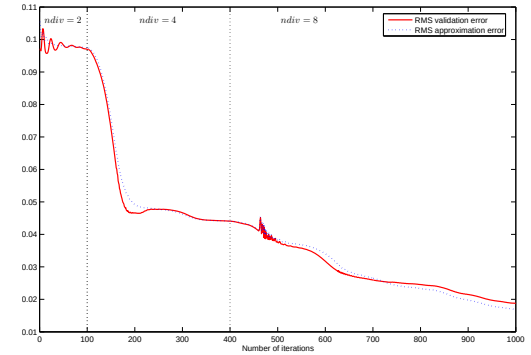


(c)

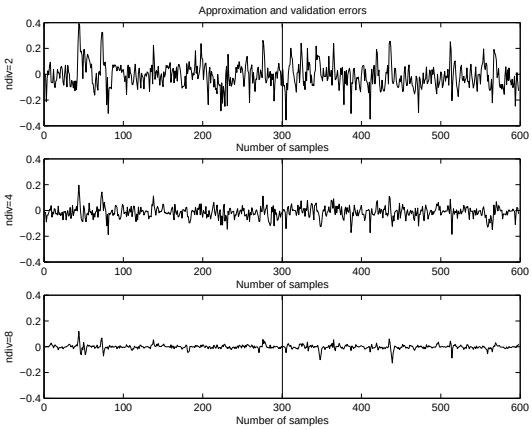
Fig. 2 Resulting f_{pwl} of Example 5.1 using (a) $ndiv = 2$, (b) $ndiv = 4$ and (c) $ndiv = 8$.

$ndiv = 2$ and then, following this algorithm, the number of divisions of the set $\mathbf{S}$ is increased to 4 and 8. The number of parameters $(ndiv + 1)^2$ is 9, 25 and 81, respectively. The resulting approximated f_{pwl} of the solution corresponding to the different validation surfaces on $\tilde{y}_{k-1}$ and u_k can be seen in Fig. 2. These surfaces were evaluated using the set of parameter that showed the best performance in the validation set during the approximation process.

As expected, the approximation to the nonlinear system improves when the number of divisions of the set $\mathbf{S}$ increases as is clearly shown in Fig. 3(a) and (b) In Fig. 3(a) the approximation and validation RMS errors *versus* the number of iterations for each number of divisions is depicted, showing that the decreasing rate is high each time the number of divisions is augmented. On the other hand, in Fig. 3(b) we have plotted the approximation and validation errors for the NOE HL CPWL models with $ndiv = 2, 4$ and 8. It is easy to appreciate that there is a significant reduction of both, the approximation and validation errors, when the number of divisions is increased. In order to compare the evolution, the algorithm is stopped after 100, 300 and 600 iterations for each $ndiv = 2$, $ndiv = 4$ and $ndiv = 8$, respectively. From Fig. 3(a) is clear that with extra iterations a better approximation could be obtained with $ndiv = 8$. Taking into account its simplicity, this performance shows the potentials of the identification algorithm.



(a)



(b)

Fig. 3 Error performance in Example 5.1: (a) RMS approximation and validation errors for the HL CPWL approximation functions using $ndiv = 2, 4, 8$. (b) Approximation (samples 1 to 300) and validation (samples 301 to 600) errors.

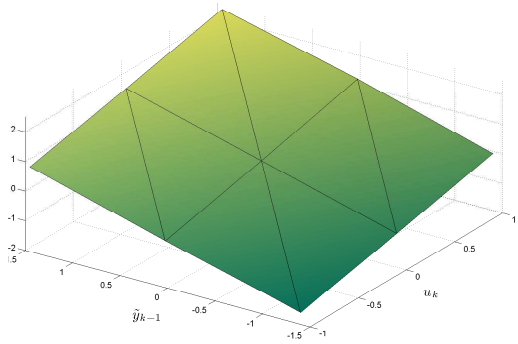
5.2 Example 2

This example, due to Yazdizdeh and Khorasani [20], is given by the following equations

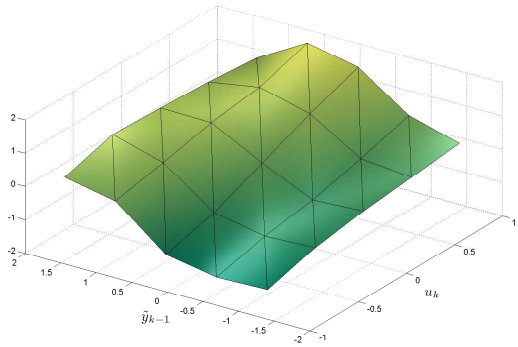
$$\begin{aligned} y(k+1) &= \frac{y(k)+u(k)}{1+y(k)^2}+\varepsilon(k), \\ \varepsilon(k+1) &= 0.25\,(\varepsilon(k)+\zeta); \end{aligned}$$

where the input $\mathbf{u}$ is a 2000 length random signal with uniform distribution between -1 and 1 and ζ is a uniform randomly generated number between -1 and 1; then $\varepsilon(k)$ can be considered as a colored added measurement noise. This noise was added to the first 1800 samples used for the identification process. In order to clearly show the approximation to the real system, 200 noisless samples were used for validation.

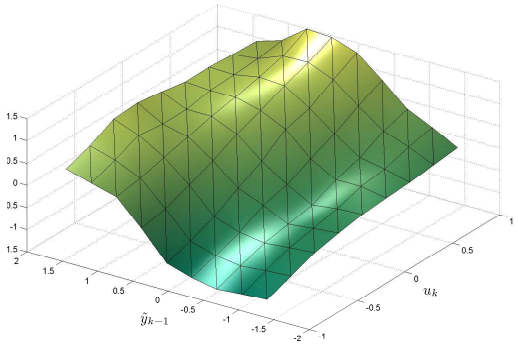
In this case, we define the regression vector with $M = 0$ and $N = 1$ as $\varphi_k = [u_k,\ \tilde{y}_{k-1}]$. Following the identification process explained in Section 3, we evaluate a OE model to generate the initial parameters. Then the number of divisions of the set $\mathbf{S}$ is increased to $ndiv = 2, 4$ and 8, following the algorithm described in that section, giving 9, 25 and 81 number of parameters, respectively. The resulting approximated f_{pwl} of the solution corresponding to the different validation surfaces on $\tilde{y}_{k-1}$ and u_k can be seen in Fig. 4. As before, these surfaces were evaluated using the set of parameter that, during the approximation process, showed the best performance in the validation set. Due to the colored noise, the surface shows some artifacts for $ndiv = 8$. In Fig. 5(a) the RMS appproximation and identification errors are displayed as a function of the number of iterations being 50, 600 and 900 the number of iterations for $ndiv = 2$, $ndiv = 4$ and $ndiv = 8$, respectively. Fig. 5(b) shows the approximation and identification errors between the system and the noisy models for $ndiv = 2, 4, 8$ using the proposed methodology. Better solutions might be obtained if both the number of data and the number of iterations were increased. This figure illustrates the very well known small noise sensibility of the NOE algorithm.



(a)



(b)



(c)

Fig. 4 Resulting f_{pwl} of Example 5.2 using (a) $ndiv = 2$, (b) $ndiv = 4$ and (c) $ndiv = 8$.

6 Conclusions and future work

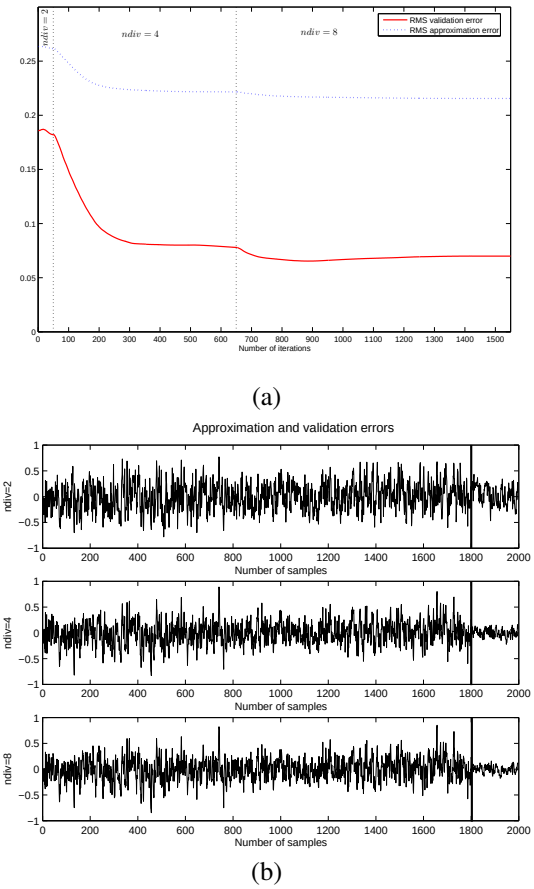
In this paper, a NOE identification algorithm based on HL CPWL functions approximation method is presented. The proposed methodology allows to approximate a NOE model from a linear OE one. The main features of this process are that it enables to go from a linear model to a nonlinear one straightforwardly and that BIBO stability can be guaranteed as long as the initial model is BIBO stable. The advantages of the algorithm are the following. In the first place, the HL CPWL functions have a simple hardware implementation in microprocessors. Secondly, the simplicity of the mechanism for increasing or decreasing the model's degrees of freedom, retaining the achieved model approximation. The model was defined for SISO system but can be easily generalized for MISO and MIMO ones. The potentials of our approach have been illustrated with two different examples.

Although the use of HL CPWL functions is simple, the number of parameters exponentially increase with the number of divisions. In order to avoid this, we are working on extending the methodology to locally adaptive grids. This would allow to refine the grid only in regions where the system is highly nonlinear.

This work is part of an ongoing project focused on the development of an identification oriented hardware. The VLSI realization properties of the HL CPWL functions are specially suited for this purpose. That's why, in a first approach, we are using simple steepest descent optimization algorithms for the parameter optimization process.

7 Update algorithm

In this section we give the algorithm the vector of parameters $\mathbf{c}^{d,*}$ from the vector $\mathbf{c}^{d-1,*}$. This algorithm can be modified to compute $\mathbf{c}^{d-1,*}$ from $\mathbf{c}^{d,*}$.



(a)

(b)

Fig. 5 Error performance in Example 5.2: (a) RMS approximation and validation errors for the HL CPWL approximation functions using $ndiv = 2, 4, 8$. (b) Approximation (samples 1 to 1800) and validation (samples 1801 to 2000) errors.

To deduce $\mathbf{c}^{d,*}$ from $\mathbf{c}^{d-1,*}$ is equivalent to say that it is possible to get the HL CPWL representation of a function defined on a simplicial partition H of a compact set $\mathbf{S}$ with $ndiv = 2^d$ number of divisions from its representation on the simplicial partition H of the compact set $\mathbf{S}$ with $ndiv = 2^{d-1}$ number of divisions, *i.e.* to solve the following system of linear equations

$$\mathbf{c}^{d,*}\Lambda^d\left(V^d\right) = \mathbf{c}^{d-1,*}\Lambda^{d-1}\left(V^d\right). \qquad (12)$$

In order to obtain the solution of the new representation given by Eq. (12), it is necessary to obtain it for different nesting levels (see [16, 17]).

Notation

d: nesting level, corresponding to the basis Λ^d.

$\mathbf{c}^{d,*}$: final vector of parameters associated with HL CPWL approximations using the basis Λ^d.

$n_{sec} = 2^{2d}$: number of sectors corresponding to $ndiv = 2^d$ number of divisions of the region $\mathbf{S}$ (equal number of divisions in every dimension has been assumed).

n_v: number of variables involved.

Algorithm

$d = 0$: *Nesting level 0.*
$\mathbf{c}^{d,*}$ is the solution of the linear approximation as described in Section 3.

$d = 1$: *Nesting level 1.*

$$n_{com} = \begin{pmatrix} n_v \\ 1 \end{pmatrix} = n_v,$$
$$i_v = i_n = 0,$$
$$n^{(1)} = 2n_{sec}.$$

{for $i = 1$ to n_{com}
  {for $k_1 = 1$ to n_{sec}
    $c^{d,*}_{i_n+2(k_1-1)+1} = c^{d-1,*}_{i_v+k_1}$,
  end}
  $i_v = i_v + n_{sec}$,
  $i_n = i_n + n^{(1)}$.
end}

$d \geq 2$: *Nesting level d.*

$$n_{com} = \begin{pmatrix} n_v \\ d \end{pmatrix},$$
$$i_v = i_n = 0,$$
$$n^{(2)} = 2^2 n_{sec},$$
$$n^{(d)} = 2n_{sec}^{d-1} \text{ for } d \geq 3.$$

{for $i = 1$ to n_{com}
  {for $k_d = d$ to n_{sec}
    {for $k_{d-1} = d-1$ to n_{sec}
      $\vdots$
      {for $k_1 = 1$ to n_{sec}
        $c^{d,*}_{i_n+2(k_1-1)+1} = c^{d-1,*}_{i_v+k_1}$,
      end}
      $i_v = i_v + n_{sec}$,
      $i_n = i_n + n^{(2)}$,
    end}
    $\vdots$
  $i_n = i_n + n^{(d-1)}$.
  end}
end}

8 Acknowledgments

This work was partially supported by grant 24/K011, SECyT, UNS and PICT 2003, 13468, FONCyT, ANPCyT.

9 References

[1] J. Sjöberg and L. S. H. Ngia. *Neural Nets and Related Model Structures for Nonlinear System Identification*, pages 1–28. Nonlinear Modeling: Advanced Black-Box Techniques. J. A. K. Suykens and J. Vandevalle, Eds., Kluwer Academic Publishers, 1998.

[2] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

[3] M. J. Korenberg and L. D. Paarmann. Orthogonal approaches to time-series analysis and system identification. *IEEE Signal Processing Magazine*, pages 29–43, 1991.

[4] O. Nelles. *Nonlinear System Identification*. Springer-Verlag, Berlin Heidelberg, Germany, 2001.
[5] S. Haykin. *Neural Networks: a comprehensive Foundation*. Macmillan, New York, 1994.
[6] C. Fantuzzi, S. Simani, S. Beghelli, and R. Rovatti. Identification of picewise affine models in noisy environment. *Int. J. of Control*, 75(18):1472–1485, 2002.
[7] S. Billings, H. Jamaluddin, and S. Chen. Properties of neural networks with applications to modelling non-linear dynamical systems. *Int. J. Control*, 56:319–346, 1992.
[8] E. Sontag. Nonlinear regulation: the piecewise linear approach. *IEEE Trans. on Automatic Control*, 26:346–358, 1981.
[9] S. Chen and S. Billings. Representation of non-linear systems: the NARMAX model. *Int. J. Control*, 49(3):1013–1032, 1989.
[10] R. Haber and H. Unbehauen. Structure identification of nonlinear dynamic systems - a survey on input/output approaches. *Automatica*, 26(4):651–677, 1990.
[11] J. Sjöberg, Q. Zhang, L. Ljung, A. Beneviste, B. Delyon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 35(12):1691–1724, 1995.
[12] T. Johansen and B. Foss. Constructing NARMAX models using ARMAX models. *Int. J. of Control*, 58:1125–1153, 1993.
[13] R. Murray-Smith and T.A. Johansen, editors. *Multiple Model Approaches to Modelling and Control*. Taylor and Francis, 1997.
[14] J. Roll. *Local and Piecewise Affine Approaches to System Identifiation*. PhD thesis, Linköping University, Sweden, 2003.
[15] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1:4–27, 1990.
[16] P. Julián, A. Desages, and O. Agamennoni. High level canonical piecewise linear representation using a simplicial partition. *IEEE Trans. on Circ. and Syst.*, 44:463–480, April 1999.
[17] Pedro M. Julián. *A High Level Canonical Piecewise Linear Representation: Theory and Applications*. PhD thesis, Universidad Nacional del Sur, Bahía Blanca, Argentina, UMI Dissertation Services, Michigan, USA, 1999.
[18] M. Parodi, M. Storace, and P. Julián. Synthesis of multiport resistors with piecewise-linear characteristics: A mixed-signal architecture. *Int. Journal on Circuit Theory and Applications*, 33(4):307–319, July 2005.
[19] L.R. Castro, J.L. Figueroa, and O.E. Agamennoni. BIBO stability for NOE model structure using HL CPWL functions. In *Proc. of the 24th IASTED Int. Conf., MIC 2005*, pages 91–96, Univ. California, Berkeley, 2005.
[20] A. Yazdizadeh and K. Khorasani. Adaptive time delay neural network structures for nonlinear system identification. *Neurocomputing*, 47:207–240, 2002.

ISBN 978-3-901608-32-2