FLAS: A Fuzzy Linear Adaptive System for identification of non-linear noisy functions

M.J. Araúzo Bravo*, E. Gómez Sánchez**, J.M. Cano Izquierdo†,
Y.A. Dimitriadis**, J. López Coronado†,

*Dept. of Electromechanical Engineering, University of Burgos, Spain
e-mail: marara@cid.cid.ubu.es
**School of Telecommunications Engineering, University of Valladolid, Spain
e-mail: edugom,yannis@tel.uva.es
†School of Industrial Engineering, University of Murcia, Spain
e-mail: joscan,coronado@plc.um.es

ABSTRACT

FLAS (Fuzzy Linear Adaptive System) is a self-organizing fuzzy system for non-linear function identification, that uses a learning method based on clustering to generate fuzzy rules and tune their parameters. This method reduces the influence of pattern presentation order permits building prototypes with physical meaning, allows measuring the importance of each variable, and therefore reduces the influence of noise. FLAS fuzzy membership functions are defined as barycentric coordinates in a simplex, yielding equivalence between Mandami and Takagi-Sugeno defuzzification methods. This allows FLAS to make piecewise linear interpolation and thus facilitates a rule fusion procedure. In simulations done for noisy non-linear function identification tasks, FLAS showed better results than other comparative systems yielding smaller identification error and number of rules. In the difficult task of bioprocesses variable identification FLAS also outperforms other systems. FLAS theoretical features and good identification performance provide good expectations for its implementation within different Model Based Controllers.
Keywords: Fuzzy systems, simplicial topology, clustering, non-linear systems, noisy functions.

1. INTRODUCTION

Identification and control of non-linear processes has been subject of much research due to its difficulty and industrial interest [2] [6]. Moreover, this task often involves time varying parameters, noisy data and variables of difficult measurement [4]. This work is devoted to identification task, which is required by Model Based Controllers. In problems without uncertainties, satisfactorily results can be achieved by traditional interpolation methods, such as linear regressions or splines [9]. However, in a more realistic case where measurement involve uncertainties, several existing techniques (statistics, fuzzy systems, neural networks...) can be applied, though not completely resolving the problem. All of them have advantages and drawbacks:

Statistical methods are very efficient, but need large quantities of data, and impose several properties and conditions to the problem that cannot always be verified [1].

Fuzzy systems allow to cope with these uncertainties using techniques inspired in human reasoning [3], and allow introducing explicit human knowledge [20], but their construction need expert knowledge to be transformed into rules.

Neural networks extract knowledge from data using learning methods [11], but store this knowledge in a manner difficult to use in others systems such as controllers.

There are many hybrids systems [18] that combine the advantages of these techniques with different objectives. FLAS, the system proposed here is aimed to the application of identifiers to non-linear Model Based Controllers in problems with uncertainty. FLAS combines the facilities of fuzzy systems to integrate human knowledge, a clustering algorithm to extract knowledge from noisy examples, and a representation of this knowledge in a piecewise linear reasoning surface that makes control design easier.

The rest of this paper is organized as follows: section 2 introduces some concepts of complex simplicial topology, and describes FLAS; section 3 shows experimental results for SISO and MISO identification; section 4 discusses FLAS application to Model Based Controllers; finally section 5 presents the conclusions.

2. FLAS DEFINITION

FLAS (Fuzzy Linear Adaptive System) is a self-organizing fuzzy system for function identification, that maps functions coding knowledge in m piecewise linear reasoning surfaces $f:\Re^n \rightarrow \Re^{n+1}$, where n is the input space dimension, m is the output space dimension and $N=n+m$, is the problem dimension.

Although there exist fuzzy systems that implement piecewise linear mappings [13], they use triangular memberships functions over rectangular supports, and therefore the support of a fuzzy set is constrained to have its limits coincident with the triangle vertices of all neighbor sets. This restriction is very hard to implement in self-organizing systems, since it means that triangle vertices should lay on a hyperectangular grid. To overcome such restriction while retaining the piecewise linear mapping capability, FLAS is proposed here based on the properties of the complex simplicial topology. To clarify the ideas involved, some basic definitions related to complex simplicial topology are presented below.

Complex Simplicial Topology

Definition. *Let $\vec{p}^{\,0},...,\vec{p}^{\,n}$ be independent points in the Euclidean space $\Re^N$, then the set of all points $\vec{r}=\sum_{i=1}^{n}\lambda_i \cdot \vec{p}^{\,i}$ with $\sum_{i=1}^{n}\lambda_i = 1$ and $\lambda_i > 0$, is called open n-dimensional* *simplex* $\sigma = (\sigma^n) = (\vec{p}^0,...,\vec{p}^n)$, *with vertex the* $\vec{p}^0,...,\vec{p}^n$ *points. If the condition* $\lambda_i > 0$ *is relaxed to* $\lambda_i \geq 0$, *the resulting set is called closed n-dimensional simplex:* $[\sigma] = [\sigma^n] = [\vec{p}^0,...,\vec{p}^n]$, *and the set* $\bar{\sigma}^n = [\sigma^n] - (\sigma^n)$ *is called the simplicial boundary of* σ^n.

It is easy to prove that the simplex $[\sigma^n]$ and (σ^n) are convex sets [10]. Then the following alternative definition for closed simplex is also valid:

Alternative definition. *If* $\vec{p}^0,...,\vec{p}^n, \in \Re^N$, $1 \leq n \leq N$, *the convex hull*

$$\sigma = \left\langle \vec{p}^0,...,\vec{p}^n \right\rangle = \left\{ \sum_{i=0}^{n} \lambda_i \cdot \vec{p}^i : \sum_{i=0}^{n} \lambda_i = 1, \lambda_i \geq 0 \right\} \quad (1)$$

of the set $\{\vec{p}^0,...,\vec{p}^n\}$ *is called a closed n-simplex if its (directed) n-dimensional volume:*

$$\mathrm{Vol}_n \left\langle \vec{p}^0,...,\vec{p}^n \right\rangle = \frac{1}{n!} \begin{vmatrix} 1 & p_1^0 & \cdots & p_n^0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & p_1^n & \cdots & p_n^n \end{vmatrix} \quad (2)$$

is nonzero, where $\vec{p}^i = \left(p_1^i,...,p_n^i\right)$, *i = 0,...,n, and* λ_i *are the coefficients of the convex combination.*

The coefficients λ_i can also be considered the coordinates of a point $\vec{r}$ in the subspace generated by vertices $\vec{p}^i$, and they are called *barycentric coordinates** with respect to the reference points p_i. Then if $\langle \vec{p}^0,...,\vec{p}^n \rangle$ is a *n*-simplex, any $\vec{r} = (r_1,...,r_n)$ in $\Re^N$ can be expressed by a $(n+1)$-tuple $(\lambda_0,...,\lambda_n)$, where

$$\lambda_i = \lambda_i(\vec{r}) = \frac{vol_n \left\langle \vec{p}^0,...,\vec{p}^{i-1}, \vec{r}, \vec{p}^{i+1},...,\vec{p}^n \right\rangle}{vol_n \left\langle \vec{p}^0,...,\vec{p}^n \right\rangle} \quad (3)$$

The *n*-simplex is for the degenerated case where *n*=0 a point; for *n*=1 an interval; for *n*=2 a triangle; for *n*=3 a tetrahedron. The simplex σ^n can be considered as the simplest *n*-dimensional figure of $\Re^N$ geometry.

Definition. *The r-dimensional simplex generated by r+1 arbitrary vertexes taken from the n+1 vertexes of a simplex* σ *are called lateral r-dimensional simplex of* σ, *or simply faces* σ^r *of* $\sigma = \sigma^n$.

Definition. *A set **K**, made of a finite number of simplexes in an Euclidean space* $\Re^N$ *is called a simplicial complex, or more briefly a complex, when it verifies the next two conditions:*

[S1] *If a simplex belongs to **K**, all its lateral simplex also belong to **K**.*

[S2] *If* σ_1 *and* σ_2 *are two simplex in **K**, then* $(\sigma_1) \cap (\sigma_2) = \varnothing$.

Definition. *The field of a simplicial complex **K** in* $\Re^N$, *i.e. the set made of all the simplex in **K**, provided with by the topology induced in it by that of* $\Re^N$, *is called rectilinear polyhedron R=|**K**|. **K** is said to be one simplicial decomposition or triangulation of R.*

Definition. *If* $\vec{r}$ *is a point in a polyhedron R=|**K**| with the triangulation **K**, the simplex in **K** univocally determined as the open simplex that contains* $\vec{r}$, *is called simplex support of* $\vec{r}$, *or briefly support of* $\vec{r}$.

Definition. *Let* $\vec{p}$ *be a vertex of a simplicial complex **K**, then the set of all the polyhedron points whose simplex supports have vertex* $\vec{p}$, *is called the (open) star* **st** $\vec{p}$ *of* $\vec{p}$.

With these definitions it is possible to derive a fuzzy system associated to a simplicial complex ***K***, by associating a fuzzy set to each vertex $\vec{p}$ in ***K***. The membership function μ_p of this set has the star of $\vec{p}$ as support. Therefore this support is formed by the union of the simplex supports of all the simplex with vertex $\vec{p}$. Once defined the fuzzy support, the membership function and its properties shall be studied.

Relation between fuzzy systems and simplicial topology

From the barycentric coordinates λ_i definition (3), it is immediate to see that these coordinates coincide with the convex combination parameters (1). Then, the necessary and sufficient condition for a point $\vec{x}$ to lay inside the simplex σ, is that its barycentric coordinates λ_i with respect to the simplex vertices verify:

$$0 \leq \lambda_i \leq 1, \text{for } i = 0,...,n \quad (4)$$

FLAS makes a piecewise linear mapping *f* from a *n*-dimensional complex ***K*** of $\Re^n$ to *m* *n*-dimensional complexes K_j (*j* = 1, ..., *m*) of $\Re^{n+1}$, where the pieces are *n*-simplexes. Then, if $\vec{x}$ lays inside a simplex $\sigma \subset K$, the prediction for the *j*-th output variable has to verify:

$$\hat{y}_j^{\sigma}(\vec{x}) = f(\vec{x}) = a_{o(0)} + \sum_{i=1}^{n} a_{\sigma(i)} \cdot x_i \quad (5)$$

where $a_{\sigma(i)}$ for *i* = 0 to 1, are the linear combination coefficients, that depend on the simplex σ vertices $(\vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(n)})$ and on $\vec{p}_j^{\sigma} = (p_{n+j}^{\sigma(0)},...,p_{n+j}^{\sigma(0)})^t$, where $p_{n+j}^{\sigma(i)}$ is the prediction for the *j*-th output variable made by the *i*-th vertex of the simplex σ, i.e. the (*n*+*j*)-component of the prototype associated to the *i*-th vertex of the simplex σ. Then, to obtain these coefficients is sufficient to solve the previous linear system (5), yielding:

$$a_{\sigma(i)} = \frac{vol_n \left\langle \vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(i-1)}, \vec{p}_j^{\sigma}, \vec{p}^{\sigma(i+1)},...,\vec{p}^{\sigma(n)} \right\rangle}{vol_n \left\langle \vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(n)} \right\rangle} \quad (6)$$

Using Mandami defuzzification method, the estimation of the *j*-th output variable corresponding *n*-dimensional input vector $\vec{x}$ inside a simplex $\sigma \subset K$, has the form:

* In mechanics, if masses λ_i that sum to one are located in points $\vec{p}^i$ then the point $\vec{r}$ is the gravity center of the total mass.

$$\hat{y}_j^{\sigma}(\vec{x}) = \frac{\sum_{i=1}^{c} p_{n+j}^{i} \cdot \mu^{i}(\vec{x})}{\sum_{i=1}^{c} \mu^{i}(\vec{x})} \tag{7}$$

However, as $\vec{x}$ is inside a simplex σ, only the n+1 fuzzy sets associated to the vertices $(\vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(n)})$ of this simplex σ can verify the convex conditions given by (4), and then (7) can be simplified to:

$$\hat{y}_j^{\sigma}(\vec{x}) = \frac{\sum_{i=0}^{n} p_{n+j}^{\sigma(i)} \cdot \mu^{\sigma(i)}(\vec{x})}{\sum_{i=0}^{n} \mu^{\sigma(i)}(\vec{x})} \tag{8}$$

where $\mu^{\sigma(i)}(\vec{x})$ is the part of the membership function of the fuzzy set associated to $\sigma(i)$ vertex, corresponding to the simplex support of σ. It is easy to prove that the necessary and sufficient condition for (8) to be a linear combination of the input variables $\vec{x}$, is that its denominator $\sum_{i=0}^{n} \mu^{\sigma(i)}(\vec{x})$ is 1. Then, comparing this expression to linear combination (5) and coefficients in expression (6) yields:

$$\mu^{\sigma(i)}(\vec{x}) = \frac{vol_n\left\langle \vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(i-1)},\vec{x},\vec{p}^{\sigma(i+1)},...,\vec{p}^{\sigma(n)}\right\rangle}{vol_n\left\langle \vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(n)}\right\rangle} \tag{9}$$

This expression is coincident with the barycentric coordinate definition (3), which implies that the membership functions over a simplex are the barycentric coordinates with respect to this simplex. Thus, the fuzzy set associated to a cluster prototype $\vec{p}^{i}$ has as support the star **st** $\vec{p}^{i}$, and as membership function a linear hyperpyramid of height 1 in the vertex $\vec{p}^{i}$, and 0 in the rest of star vertices.

As a corollary of this result, considering that the coefficients of the convex conditions (4) coincide with the barycentric coordinates (3) and therefore with the membership values (9), it can be pointed out that the same conditions used to calculate which fuzzy sets are active, calculate also membership degrees to these sets.

Finally, an important result is derived: with the proposed membership function Mandami and Takagi defuzzification methods are equivalent. To prove this it should be noted that as it has been imposed to denominator in (8) to be 1, with the proposed membership functions (barycentric coordinates), the defuzzification equation (8) can be expressed as:

$$\hat{y}_j^{\sigma}(\vec{x}) = \frac{\sum_{i=0}^{n} p_{n+j}^{\sigma(i)} \cdot vol_n\left\langle \vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(i-1)},\vec{x},\vec{p}^{\sigma(i+1)},...,\vec{p}^{\sigma(n)}\right\rangle}{vol_n\left\langle \vec{p}^{\sigma(0)},...,\vec{p}^{\sigma(n)}\right\rangle} \tag{10}$$

This expression is linear in $\vec{x}$, and therefore equivalent to the Takagi-Sugeno defuzzification [12].

Clustering

Once the form of membership function has been defined, it becomes necessary to define its parameters. These are the vertices coordinates (class prototypes) $\vec{p}^{i}$, where i=1,...,c are the different classes. Traditionally values of fuzzy systems parameters are selected (i.e. fuzzy rules constructed) by the problem experts. FLAS allows this possibility, but as an self-organizing system it can automatically tune them by learning from examples. In order to do this, FLAS has a learning algorithm similar to some classical clustering algorithms [16], but with the following differences:

Distributed learning: Most classical clustering algorithms [16] are *winner-take-all*. This implies that each training pattern is learned by only one class. Thus, these algorithms are more sensitive to pattern presentation order. This influence can be reduced allowing each pattern to be learnt by all active clusters.

Use of variable learning rate: A typical learning equation used in several self-organizing systems [16] is:

$$W_J^{new} = \beta \cdot I + (1-\beta) \cdot W_J^{old} \tag{11}$$

where W_J^{new} and W_J^{old} are the new and old prototypes of class J, $\beta \in [0\ 1]$ is a constant learning rate and I is the training pattern. Parameter β determines the learning innovation level: the closer to 1, the more innovative. However, using this learning equation has the drawback that the resulting class prototype has not a clear physical meaning

Using for each class J a variable learning rate β_J of the form:

$$\beta_J = 1/n_J \tag{12}$$

where n_J is the number of patterns classified in the class J, it can be easily demonstrated that cluster prototype is the gravity center of patterns coded in class J. This method reduces the influence of pattern presentation order and provides a physical meaning to cluster prototypes, that can be useful to apply statistical techniques. It can also be observed that this choice makes learning more conservative in classes that have coded a large number of patterns.

Ellipsoidal reset: To control if a pattern belongs to a class, clustering algorithms use a reset algorithm. This may put limits to the perimeter of hyperrectangle associated to the class, as in FasArt [5], or to the radius of the hypersphere, as in the classical algorithms shown in [16]. This strategy is less sensitive to noise [19], but both suppose all features equally important. FLAS generalizes the latter strategy by allowing each feature to have different importance. In order to do this, the reset mechanism sets an ellipsoidal condition, where the ellipse semiaxis γ_i can be interpreted statistically as the uncertainty level of feature i.

These γ_i parameters are the only that need to be tuned by the user. However, considering that they are related to the uncertainty level in the variables, their values can be selected using statistical techniques. Moreover, it is also possible to used techniques inspired in fractal theory [14], establishing a relation between the uncertainty levels γ_i and the resolution scale η at which the function should be mapped. Therefore, the optimal γ_{i_opt} can be estimated as the scale η_{opt} at which the function starts to have a non-fractal behavior, i.e. the scale at which the hypervolume V of the mapping is invariant. Moreover this technique allows estimating experimentally the fractal dimension D of the problem, using Besicovic dimension definition [14]:

$$V = \eta^{D-1} \tag{13}$$

where V is the hypervolume of the reasoning surface, η is the resolution scale and D the fractal dimension. Then it is possible to calculate an estimate of the fractal dimension $\hat{D}$, approximating the resolution scale η to the average square radius $\gamma = (\sum_{i=1}^{n} \gamma_i^2)^{1/2}$ of the reset hyperellipsoid, i.e.:

$$\hat{D} = \mathrm{Ln}\left(\frac{L}{\gamma} + 1\right) \tag{14}$$

Tessellation
In two-dimensional problems (n=1, m=1), once the class prototypes $\vec{p}^{\,i}$ are known the identification problem is already solved. To predict the output associated to an input point it is only necessary to make a piecewise linear interpolation based on the two clusters prototypes closer to that input point. Unfortunately, in the general (n+m)-dimensional problem the generalization of the previous strategy, i.e. taking the n+1 closer prototypes to the input point, does not produce a simplicial complex, since condition [S2] is not verified. In fact the associated reasoning surface is non-continuous. Then, to solve this problem it becomes necessary to generate a simplicial complex $\boldsymbol{K}$ with vertices the cluster prototypes $\vec{p}^{\,i}$ before to make predictions, thus ensuring the continuity of the prediction function. FLAS allows the user to make these links manually, but it can do them automatically using any of the two algorithms described below.

Optimum algorithm: In Finite Elements Methods (FEM), there exist many criteria to define if a triangulation is optimum [17]. Among them, the *shortest diagonal criterion* has been selected since it is the easiest to implement. This is very important since FLAS is thought for high dimensional problems. To achieve the optimum solution with the *shortest diagonal criterion* implies to calculate all possible distances among all cluster prototypes $\vec{p}^{\,i}$, and to select the shortest links among them. Although this method has a high computational cost, it produces optimal simplicial complex $\boldsymbol{K}^*$ (in the sense that links are the shortest) and it is independent of construction order. The generated complex $\boldsymbol{K}^*$ is a convex set. This algorithm can be seen as a neural field in which all neurons are initially connected, but dynamically shorter connections inhibit larger connections that cross them. In order to reduce computational cost topological properties of simplicial complex can be applied.

Constructive algorithm: This method starts with an initial simplex σ_0 and in each step k introduces a new vertex $\vec{p}^{\,k}$, that is linked to all previously introduced vertices if condition [S2] is verified. This method generates a new $\boldsymbol{K}^k$ complex in each step k that is a convex set. The algorithms ends in step k=c-(n+1) when all cluster prototypes have been connected. It is verified that the boundary of the optimal complex $\overline{K}^*$ coincides with the boundary of the last complex generated by the constructive method $\overline{K}^{c-(n+1)}$. This method has the advantage of being very fast, but the prediction surface is generally suboptimal, and the simplex generated depends on the order of presentation of prototypes $\vec{p}^{\,k}$.

Seek for an optimal order of presentation of prototypes $\vec{p}^{\,k}$ for which complex $\boldsymbol{K}$ coincides with $\boldsymbol{K}^*$ is subject of ongoing research.

Rule Fusion
As previously mentioned, one of the disadvantages of self-organizing fuzzy systems is that they generally generate more fuzzy rules than those that would be selected by experts. FLAS design is specially oriented to perform an automatic post-processing rule fusion. This is achieved by an algorithm that makes use of the piecewise linear form of the reasoning surface. This algorithm is as explained below for MISO identification, but it can be easily extended to MIMO problems:

1. For each prototype $\vec{p}^{\,i}$, find out if it is included in a simplex σ_i, using convex conditions (4).
2. With the simplex σ_i make a prediction on $\vec{p}^{\,i}$, denoted by $\hat{\sigma}_i$, while the prediction of the prototype is p_{n+1}^{i}. Then the n+1 simplexes that form the $\vec{p}^{\,i}$ star can be substituted by only one simplex (the simplex σ_i) if $\left|\hat{\sigma}_i - p_{n+1}^{i}\right| \le f(\gamma_{n+1})$, where $f(\gamma_{n+1})$ is a function of the uncertainty level in variable n+1, usually the identity function. With this procedure in each iteration the fuzzy rule base is reduced in n rules.

This process is applied until no further fusion is achieved. The fusion algorithm can be generalized to cope with cases where not only a simplex σ_i containing the vertices of prototype $\vec{p}^{\,i}$ is searched, but also a sub-complex $\boldsymbol{S} \subset \boldsymbol{K}$ that contain these vertices.

FLAS identification algorithm
Once FLAS learning mechanisms have been introduced, it is easy to define its identification algorithm. If an input vector $\vec{x}$ = (x_1,...,x_n) is presented:

1. The fuzzification procedure consists of searching the n+1 fuzzy sets activated by $\vec{x}$, i.e. the n+1 vertices of the simplex σ that contain $\vec{x}$. This is the simplex whose n+1 barycentric coordinates λ_i, verify (4). These values are also the membership values μ^{σ} as shown in the demonstration of equation (9).
2. Based on the piecewise linear reasoning surface implemented by FLAS, the defuzzification procedure is equivalent to an interpolation for each of the m outputs. It has also been shown that this defuzzification is equivalent to Mandami and Takagi-Sugeno defuzzification methods.

3. EXPERIMENTAL RESULTS

SISO identification performance
A first experiment is carried out to test FLAS performance in a two-dimensional problem (n=1, m=1, i.e. a SISO system) taken from the literature [15], in order to compare with other identification systems such as Fuzzy ARTMAP [8], PROBART [15], FasArt [5] and FasBack [7]. All these systems are supervised neural architectures based on the Adaptive Resonance Theory (ART) [11]. Their general architecture consists of two self-organizing classifier that cluster the input space (with N_a nodes) and the output space (with N_b nodes), and a module linking the classifiers. They have been selected for comparison due to the fact that they all are self-organizing systems, and all but PROBART are fuzzy neural networks that allow knowledge introduction and interpretation, as well as FLAS. Moreover,

PROBART has been selected since it is specially designed to treat noisy problems.

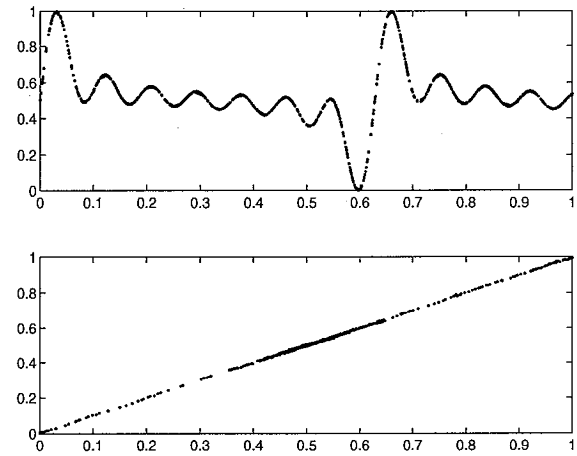

Figure 1: FLAS identification performance. Top: actual values and FLAS estimation; bottom: actual vs. estimated.

The proposed test, taken from [15], consists of the identification of a continuous noisy nonlinear signal:

$$f(x) = \frac{1}{20} \cdot \left[10 + \sum_{i=1}^{7} sin(10 \cdot i \cdot x) \right] \quad (15)$$

where x is in radians. The range of test function $f: \Re \rightarrow \Re$ is [0.2295, 0.7705] for the input domain [0 1]. Gaussian noise, derived from a zero mean source with unit variance, has been added to the signal with a scale factor of 0.02. Thus the corrupted output signal for pattern p is given by $y_p = y(p) = f(x_p) + 0.02\varepsilon_p$, where x_p is the x-coordinate and $\varepsilon_p \sim N(0,1)$ is the Gaussian noise added. The x-coordinates were randomly chosen from a uniform distributed source. The training an test sets were generated with different sets of x-coordinates, but test output samples were noise-free. Performance has been evaluated by the root mean square error (RMSE), maximum absolute error (MAXAE) and the number of clusters generated. The RMSE value is computed as:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{p=1}^{N} \left\| d_p - y_p \right\|^2} \quad (16)$$

where d_p is the desired output for pattern p, y_p is the actual output, and N is the number of patterns used for test. As previously mentioned, all identifiers were trained on noisy data. Both training and test sets consisted of 1000 data pairs. In all experiments FLAS parameters have been $\gamma_1 = \gamma_2 = 0.02$.

Experimental results are shown in table 1, where RMSE and MAXAE are the aforementioned error indices, N_a and N_b, are the number of nodes used by each system, which are an indication of their complexity. As FLAS only has one module N_b, is not given. In all systems N_a can be considered as the number of rules generated. Figure 1 shows FLAS estimation is very accurate. Furthermore, it can be seen that the clustering method implemented in FLAS achieves better results than all other identifiers, both in error indices and complexity.

Table 1: Identification performance for the function given in [r13]

Model	N_a	N_b	RMSE	MAXAE
Fuzzy ARTMAP	806	61	0.0302	0.0679
PROBART	112	61	0.0202	0.0905
FasArt	275	30	0.0097	0.0427
FasBack	284	30	0.0078	0.0389
FLAS	128		0.0061	0.0262
FUSION FLAS	100		0.0061	0.0262

In a two-dimensional problem only test FLAS clustering performance has been evaluated, where tessellation is unnecessary. A three-dimensional problem is proposed here in order to deeply evaluate FLAS capabilities.

Viscosity Identification

Identification and control of biochemical processes is a difficult task due to their strong non-linear dynamics, time varying parameters and noisy variables, often of difficult and expensive measurement [2], [6]. Then, to illustrate FLAS identification performance on a real application case, the task of viscosity identification in the penicillin production process is studied here, using real data from Antibióticos S.A.U pilot plant. Results presented here correspond to MISO identification (n=2, m=1). For confidentiality reasons all the results are normalized in [0 1]. FLAS was trained on data from 28 fermentations, with parameters $\gamma_1 = \gamma_2 = \gamma_3 = 0.1$, to produce the reasoning surface shown in figure 2.

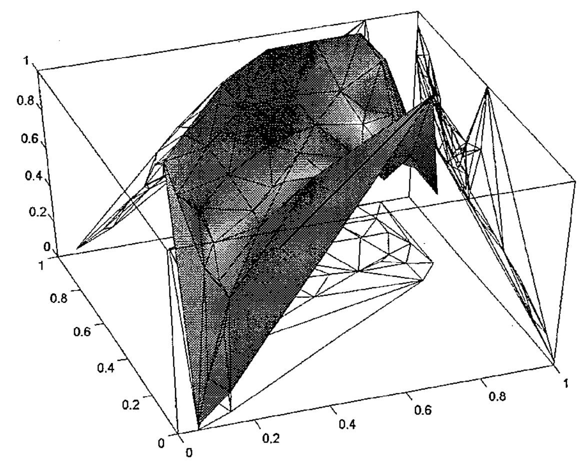

Figure 2: FLAS reasoning surface for viscosity identification.

Table 2: Identification performance in viscosity problem

Model	N_a	N_b	RMSE	MAXAE
FasArt	87	16	0.06633	0.174292
FLAS	44		0.05816	0.117020

Once trained, FLAS was tested on an unseen fermentation. For comparison with FLAS, FasArt neuro-fuzzy system has been selected because in table 1 it shows the best compromise between error and complexity among other systems taken from the literature. Results in table 2 show that FLAS can be used in multidimensional real problems outperforming other existing systems such as FasArt. While error is reduced, the generation of a more compact rule set help experts understand the process and introduce new rules, and thus facilitate the control design.

5. APPLICATION TO CONTROL DESIGN

As previously mentioned, FLAS has several properties that make it suitable for Model Based Controllers. These features derive from the piecewise linear reasoning surface FLAS implements. Its advantages to design modules for several control architectures are summarized below.

Internal Model Control: FLAS rules can be easily inverted.

Model Based Predictive Control: the application of linear programming optimization theories to develop an optimization procedure for predictive controllers based on FLAS is subject of ongoing research.

Fuzzy Control: Membership functions proposed in FLAS imply an equivalence between Mandami (used mainly in classification problems), and Takagi-Sugeno (used mainly in control problems) defuzzification methods. Furthermore, the interpretation of the barycentric coordinates as membership functions, facilities the explicit knowledge introduction in the controller by experts in the problem to solve.

6. CONCLUSIONS

FLAS has been presented as a self-organizing fuzzy system that performs piecewise linear mapping of non-linear functions. The membership functions of the fuzzy sets present in FLAS rules are defined as the barycentric coordinates over simplicial complexes. Theoretical properties derive from this definition, standing out the equivalence between Mandami and Takagi defuzzification methods, the capability to generate piecewise linear reasoning surfaces and the facility to reduce the number of rules using the proposed rule fusion method.

As many identification systems, FLAS design parameters are user tunable. However, these parameters have clear physical meaning, since they correspond to the maximal uncertainty level expected for each signal. This has at two-fold benefit: in one hand the identifier design can be done using knowledge of the problem; in the other hand it facilitates the analysis of the system, since there exists a relation between the values of these parameters and the fractal dimension of the problem.

Experimental results have been obtained for identification tasks using simulated and real data, showing that FLAS allows a reduction in both prediction error and system complexity.

Furthermore, FLAS theoretical features and good identification performance provide good expectations for its implementation within different Model Based Controllers: its rules can be easily inverted, as required by IMC; linear programming could be used to build optimizers on FLAS models in order to build MBPC; and its reduced rule set and easiness to introduce new rules, make it advantageous to build fuzzy controllers.

Acknowledgments
Authors would like to thank E. Baeyens, E. Parrado R, Sacristán and M. Martín-Merino. for their helpful suggestions, and to experts at Antibióticos S.A.U. pilot plant for their suggestions and collaboration in experiments shown. E. Gómez Sánchez was partially supported by ESPRIT 22416 "MONNET" with Cedetel.

7. REFERENCES

[1] R. G. Almond, "Discussion: fuzzy logic: better science or better engineering", *Technometrics*, Vol. 37, 1995, pp. 267-270.

[2] M.J. Araúzo, E. Gómez, J.M. Cano, J. López, M. López, A. Collados, "Control of the penicillin production with adaptive IMC using fuzzy neural networks", *Proceedings of 14th World Congress of IFAC International Federation of Automatic Control*, Beijing, China. Jul. 1999.

[3] P.P. Bonissone, "Discussion: fuzzy logic control tecnology: A personal perspective", *Technometrics*, Vol. 37, No. 3, 1995, pp. 262-266.

[4] J.D. Boskovic and K.S. Narendra, "Comparison of linear, nonlinear and neural-network-based adaptive controllers for a class of fed-batch fermentation processes", *Automatica*, Vol. 31, No.6, 1995, pp. 817-840,

[5] J.M. Cano, Y.A. Dimitriadis, M.J. Araúzo and J. López, "FasArt: a new neuro-fuzzy architecture for incremental learning in system identification", *Proceedings of 13th World Congress of IFAC International Federation of Automatic Control*, San Francisco, USA. Vol. F, Jul. 1996, pp. 133-138.

[6] J.M. Cano, Y.A. Dimitriadis, M.J. Araúzo, F. Abajo. and J. López, "Fuzzy adaptive system ART-based: theory and application to identification of biochemical systems", *Proceedings of CESA/96*, Lille, France, Vol 2, Apr. 1996, pp. 918-923.

[7] J.M: Cano, Y.A. Dimitriadis and J. López, "FasBack: matching-error based learning for automatic generation of fuzzy logic systems", *Proceedings of FUZZ-IEEE*, Barcelona, Spain, 1997.

[8] G. Carpenter, S. Grossberg, N. Markuzon and J. Reynolds, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps" *IEEE Transactions on Neural Networks*, Vol. 3, 1992, pp 698-713.

[9] C.K. Chui and M.J. Lai, "Multivariate vertex splines and finite elements", *Journal of Approximation Theory*, Vol. 60, No. 3, 1990, pp. 245-343.

[10] W. Franz, *Topologie*, Walter de Gruyter & Co., 1968. (in German, also translated).

[11] S. Grossberg, "Adaptive pattern classification and universal recoding. II: Feedback expectation, olfaction and illusions," *Biological Cybernetics*, Vol. 23, 1976, pp. 187-202.

[12] J. Joh and Y.H. Chen, "On the stability Issues of linear Takagi-Sugeno fuzzy models", *IEEE Transactions on Fuzzy Systems*, Vol. &, No. 3, 1998.

[13] F. L. Lewis, "Towards a paradigm for fuzzy logic control", *Automatica*, Vol. 32, No. 2, 1996, pp. 167-181..

[14] B. Mandelbrot, *Les objects fractals: forme, hasard et dimension*, Paris, Guillimard, 1987. (in French, also translated).

[15] S. Marriot and R.F. Harrison, "A modified Fuzzy ARTMAP architecture for the approximation of noisy mappings", *Neural Networks*, Vol. 8, No.4, 1996, pp. 619-641.

[16] B. Moore, "ART 1 and pattern clustering", *Proceedings of the 1988 Connectionist Models Summer School*, Morgan Kaufmann Publishers, 1989, pp. 174-185.

[17] L.L. Schumaker, "Triangulation methods", in *Topics in Multivariate Approximation*, C.K. Chui, L.L. Schumaker and F.I. Utreras, Eds., New York, Academic Press, 1987, pp. 219-332.

[18] L. Wang, *Adaptive fuzzy systems and control*, New York, NY: Prentice Hall, 1994.

[19] J. Williamson, "Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps", *Neural Networks*, Vol. 9, No. 5, 1996, pp. 881-897.

[20] L. Zadeh, "Fuzzy sets". *Information and Control*, Vol. 8, 1965.