MATRIX-BASED HIERARCHICAL FUZZY SYSTEMS

Santiago Aja-Fernández and Carlos Alberola-López*
Laboratorio de Procesado de Imagen
ETSI Telecomunicación, Universidad de Valladolid
Campus Miguel Delibes 47011 Valladolid, Spain)

Keywords: Hierarchical fuzzy systems, FITM, transition matrices, SAM.

Abstract: A matrix inference method for fuzzy systems is used to deal with hierarchical fuzzy systems (HFSs). A method to decompose a multiple input fuzzy system into a HFS is presented. This method is based in representing the structure of a fuzzy system using matrices. An example of such a conversion for a three-input system is included.

1 INTRODUCTION

In fuzzy systems (FSs) and fuzzy controllers with multiple inputs a rule-explosion problem arises when the number of inputs is large. As the number of rules increases, the rule base is less and less understandable to human users. This problem is known as the *curse of dimensionality*, which in fuzzy systems causes some well known problems related with computational and memory usage, real-time performance, and difficulty to properly define the system. One of the methods proposed to overcome this problem is the use of hierarchical fuzzy systems (HFSs); a complex system with many rules is decomposed into a number of hierarchically-connected low-dimensional FSs. These new systems, which turn out to be simpler than the original, have usually smaller rule bases.

In this paper we will present a new method to carry out this system decomposition. The method is based on a matrix procedure known as FITM (Fast Inference using Transition Matrices) (Aja-Fernández and Alberola-López, 2004a), which has been proposed as a methodology to perform inferences in SAM (Standard Additive Model (Kosko, 1997)) FSs efficiently. It is based on representing each input to the FS as a vector and the overall fuzzy inference procedure becomres a simple matrix operative. FITM is a procedure initially intended for *Computing with Words* (Zadeh, 1996) environments, but it may find applicability in other fields, such as control and feedback systems.

*The authors acknowledge the CICyT for research grant TEC2004-06647-C03-01, the FIS for grant PIO-41483, the Junta de Castilla y León for grant VA075A05 and the European Commission for the funds associated to the Network of Excellence SIMILAR (FP6-507609).

1.1 Hierarchical Fuzzy Systems

It is easy to see that if a system has n input variables and each variable has m terms (defined over m fuzzy sets) the system has a complete set of m^n different rules (Raju et al., 1991). Moreover, as previously stated, the rule base is less and less understandable to human users as the number of rules increases. Originally HFSs were proposed to overcome this problem. Instead of having one complex system with many rules, the system was decomposed into a number of hierarchically-connected low-dimensional FSs. These new systems, which turn out to be simpler than the original, have smaller rule bases, a fact that makes them more transparent to the designer. If the number of intermediate variables is *small enough* [2] the HFS has also fewer rules than the original system.

One of the first approaches is due to Raju, Zhou and Kisner (Raju et al., 1991; Raju and Zhou, 1993). An adaptive HFS is introduced in order to facilitate the tuning of the parameters of a controller. Their aim was to design a series of hierarchical fuzzy processors with a small number of input variables distributed in each processor. Yager in (Yager, 1993) suggested an extension of the basic FS modeling framework. The purpose of this extension was to allow for a prioriti-

[2] A simple criterion to quantify this "enough" can be found in (Lee et al., 2003).

Aja-Fernández S. and Alberola-López C. (2006).
MATRIX-BASED HIERARCHICAL FUZZY SYSTEMS.
In *Proceedings of the Third International Conference on Informatics in Control, Automation and Robotics*, pages 92-97
DOI: 10.5220/0001206500920097

zation of the rules by using hierarchical representation of the rules. This new representation is known as Hierarchical Prioritized Structure (HPS). In (Horáček and Binder, 1997) Horáček and Binder discussed two structures of hierarchically-organized groups of rules: series and parallel hierarchies . The aim of this work is to design a nonlinear controller whose rules were transparent to the designer. Many other approaches to overcome the dimensionality problem based on HFS have been proposed elsewhere (Holve, 2003; Lee et al., 2003). Additionally, some authors have focused on theoretical justifications of the approximation capability of HFS (Wang, 1998; Zeng and Keane, 2005).

In the HFS literature a fuzzy set with two inputs and one output is known as Fuzzy Logic Unit (FLU). Depending of the connection between the FLUs in a hierarchical decomposition of a MISO-FS, we can have three main types of hierarchical models (Horáček and Binder, 1997; Kikuchi et al., 1998) as shown in Fig. 1 (although one may think of more): *Serial HFS*, where the inputs to the FLUs are the outputs of the previous FLUs and an external input; *Parallel HFS*, where the inputs to the FLUs are the outputs of two FLUs of the previous layer and *Hybrid HFS*.

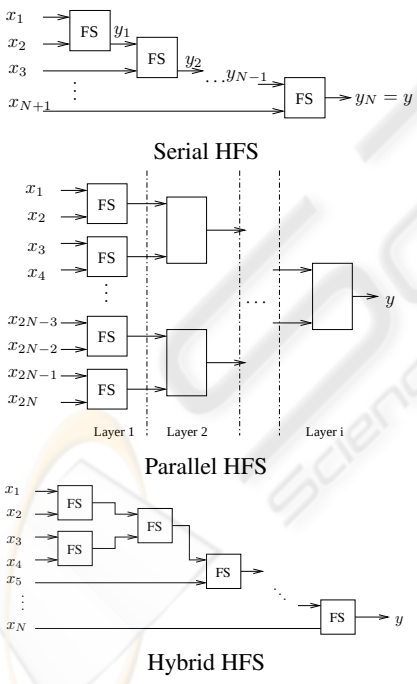

Figure 1: Different hierarchical models.

In the following sections we will show how to deal with HFSs using transition matrices. This method was previously introduced in (Aja-Fernández and Alberola-López, 2004b).

2 BACKGROUND ON FITM

The FITM procedure was originally proposed in (Aja-Fernández and Alberola-López, 2004a) to perform inferences efficiently in the SAM-FSs (Kosko, 1997); its result is proportional to the inference output given by a SAM-FS, and it is totally equivalent to the SAM-FS in terms of the output centroid. The benefit of FITM is the considerable reduction of the overall computational complexity in the inference process, provided that an initial assumption is held; specifically, inputs are required to be linear combinations of the fuzzy sets that the input linguistic variable (LV) consists of. If this is so, each input is represented as a vector in the input space, with components equal to the contribution of each fuzzy set of the input linguistic variable to the current input.

This requirement holds in *computing with words* (CWW) (Zadeh, 1996) applications, where the inputs to the FS are *words*, *concepts* or *labels*, modeled as fuzzy sets. Additionally, this sort of inputs can also be found in those problems where the output of one SAM-FS is the input of another one. This is clearly the case of hierarchical systems. General inputs and particularly, crisp inputs, can indeed be dealt with; in this case there would be no computational savings with FITM, but some benefits can be obtained from the matrix representation of the system. An extension of the procedure to non-linear operators (i.e. generic t-norms and t-conorms) has also been reported in (Aja-Fernández and Alberola-López, 2005). Following, we describe the method to build a FITM inference engine when linear operators (SAM) are used and (crisp) numerical inputs are considered; attention will be focused on the 2-input single output case (2-ISO). Only conclusions will be described here. Details can be found in (Aja-Fernández and Alberola-López, 2004a).

Assume a 2-input single output FS, with inputs X and Y and output Z. The inputs and the output are all (crisp) numbers. For each input and the output a linguistic variable (LV) is defined. Assume that the first input LV consists of M possible fuzzy sets A_k defined on the universe $U \subset R$; the second input Y is a LV consisting of N possible fuzzy sets B_l defined on the universe $V \subset R$ and the output LV consists of L possible fuzzy sets D_n defined on the universe $W \subset R$.

The whole SAM inference process may be expressed using transition matrices as

$$\boldsymbol{\gamma} = \left(\sum_{l=1}^{N} \alpha_l \boldsymbol{\Omega}_l \right) \boldsymbol{\beta} \tag{1}$$

with $\boldsymbol{\Omega}_l$ an array of transition matrices of the system. $\boldsymbol{\gamma}$ is the output vector and $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the input vec-

tors defined as

$$\boldsymbol{\beta} = \begin{bmatrix} A_1(x_0) \\ A_2(x_0) \\ \vdots \\ A_M(x_0) \end{bmatrix} = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_M \end{bmatrix} \quad (2)$$

being x_0 the numerical input. $\boldsymbol{\alpha}$ is defined in a similar way using sets B_i and the numerical input y_0. In order to build the transition matrices we must define some intermediate data structures that can be discarded once matrices $\boldsymbol{\Omega}_l$ have been calculated. How to build these matrices is now addressed.

First step is to create the *activation matrix* of each input. As both inputs will be crisp, they will make use of identity matrices, i.e. $\mathbf{R_A}$ and $\mathbf{R_B}$, will be the $M \times M$ and the $N \times N$ matrices respectively. Next step is to calculate the set of matrices $\mathbf{G}_l$, $1 \leq l \leq N$, that bear the relation between inputs

$$\mathbf{G}_l = \mathbf{R_A} \otimes [\mathbf{R_B}\mathbf{E}_l] \quad (3)$$

where $\otimes$ is the Kronecker tensor product (Golub and Loan, 1996). $\mathbf{E}_l$ is a column selection vector, i.e., a column vector with all entries zero but the one at row l, $1 \leq l \leq N$, the value of which is unity. This vector has the purpose of extracting column l from matrix $\mathbf{R_B}$.

The rule base of the system is coded in matrix $\mathbf{C}$. This is a selection matrix with as many rows as rules in the rule base and, for row j, all the entries are zero but the one at column i if the output consequent for rule j is D_i.

$$\mathbf{C} = \begin{matrix} R_1 \\ R_2 \\ \vdots \\ \vdots \\ R_{M\times N} \end{matrix} \begin{matrix} D_1 & D_2 & \cdots & D_L \end{matrix} \begin{pmatrix} 0 & 0 & \cdots & 1 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & & \cdots \\ 1 & 0 & \cdots & 0 \end{pmatrix} \quad (4)$$

As for matrix $\mathbf{C}$, rules in the rule base are ordered by varying the second antecedent for each value of the first antecedent (see Appendix I in (Aja-Fernández and Alberola-López, 2004a)).

Finally, the transition matrices are calculated by

$$\boldsymbol{\Omega}_l = \mathbf{C}^T\mathbf{G}_l \; l = 1, \cdots N \quad (5)$$

The output centroid, if desired, can be calculated from the output vector $\boldsymbol{\gamma}$ by

$$c_{\text{out}} = \frac{[c_1 \; c_2 \cdots c_L]\boldsymbol{\gamma}}{[1 \; 1 \cdots 1]\boldsymbol{\gamma}} = \frac{\mathbf{c}^T\boldsymbol{\gamma}}{\mathbf{1}^T\boldsymbol{\gamma}} \quad (6)$$

with $\mathbf{c}$ the vector of the output set centroids c_i ($1 \leq i \leq L$) and $\boldsymbol{\gamma}$ as defined in equation (1). one from the conventional SAM-FS.

For the case of a multiple-input single-output (MISO) FS the expressions are extended accordingly. The relation among coefficients is now given by:

$$\boldsymbol{\gamma} = \left(\sum_{i_1=1}^{N_1} \cdots \sum_{i_F=1}^{N_F} \alpha_{i_1}^1 \cdots \alpha_{i_F}^F \boldsymbol{\Omega}_{\prod_{j=1}^F i_j} \right) \boldsymbol{\beta} \quad (7)$$

where $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}$ are the input vectors and $\boldsymbol{\Omega}_{\prod_{j=1}^F i_j}$ are the transition matrices of the system.

3 HIERARCHICAL DECOMPOSITION OF FS

Consider the serial HFS shown in Fig. 1. A reinterpretation of this system using transition matrices and vectors is shown in fig. 2. The different inputs and outputs to the system can be viewed as vectors in their respective spaces. As it has been indicated in section 2, these vectors represent a linear combination of the sets of each linguistic variable, or, in the case of crisp inputs, the activation of each set by this input. Input variables x_i will be represented by vectors $\boldsymbol{\alpha}^i$ and output variables y_i by vectors $\boldsymbol{\gamma}^i$. Each FLU can be characterized by an array of matrices $\boldsymbol{\Omega}_l^i$.

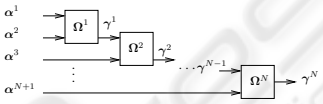

Figure 2: A HFS in FITM notation.

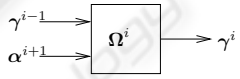

Figure 3: Single FITM block. For $i = 1$ $\boldsymbol{\gamma}^{i-1} = \boldsymbol{\alpha}^i$.

A single block of the whole system (a FLU rewritten in FITM terminology) is showed in 3. For the i^{th} ($i \geq 2$) block the inputs will be the vectors $\boldsymbol{\gamma}^{i-1}$ and $\boldsymbol{\alpha}^{i+1}$, and the output the vector $\boldsymbol{\gamma}^i$. According to eq. (1) the inference process is performed following the equation

$$\boldsymbol{\gamma}^i = \left(\sum_l \alpha_l^{i+1} \boldsymbol{\Omega}_l^i \right) \boldsymbol{\gamma}^{i-1} \quad (8)$$

Vector $\boldsymbol{\gamma}^{i-1}$ is the output of the previous block and $\boldsymbol{\alpha}^{i+1}$ is an external input. According to eq. (8) we can write the output of the overall system as

$$\boldsymbol{\gamma} = \left(\sum_{l_N} \alpha_{l_N}^{N+1} \boldsymbol{\Omega}_{l_N}^N \right) \cdots \left(\sum_{l_1} \alpha_{l_1}^2 \boldsymbol{\Omega}_{l_1}^1 \right) \boldsymbol{\alpha}^1 \quad (9)$$

which can be rewritten as

$$\begin{aligned} \boldsymbol{\gamma} &= \left(\sum_{l_N} \cdots \sum_{l_1} \alpha_{l_N}^{N+1} \cdots \alpha_{l_1}^2 \boldsymbol{\Omega}_{l_N}^N \cdots \boldsymbol{\Omega}_{l_1}^1 \right) \boldsymbol{\alpha}^1 \\ &= \left(\sum_{l_N} \cdots \sum_{l_1} \alpha_{l_N}^{N+1} \cdots \alpha_{l_1}^2 \boldsymbol{\Omega}_{l_N \cdots l_1} \right) \boldsymbol{\alpha}^1 \quad (10) \end{aligned}$$

This is the FITM inference equation for a MISO sys-

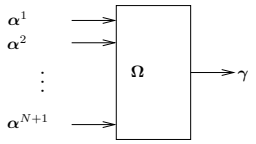

Figure 4: Multiple Input Single Output fuzzy system.

tem as the one shown in fig. 4, with its transition matrices defined as

$$\mathbf{\Omega}_{l_N \cdots l_1} = \mathbf{\Omega}^N_{l_N} \cdots \mathbf{\Omega}^2_{l_2} \mathbf{\Omega}^1_{l_1} \tag{11}$$

There is an equivalent relation between the transition matrices of a MISO FS and the matrices of each FLU of the derived HFS. Knowing the matrices of the FLU of the HFS we can easily calculate the matrix of the overall system. The inverse procedure is also possible.

3.1 From MISO to HFS. Special Case: 3ISO

Consider a fuzzy system with 3 inputs and 1 output, as shown in fig. 4. The imputs will be x_1 (M possible fuzzy sets, vector $\boldsymbol{\alpha}^1$), x_2 (N possible fuzzy sets, vector $\boldsymbol{\alpha}^2$) and (x_3: P possible fuzzy sets, vector $\boldsymbol{\alpha}^3$). The output is y (Q possible fuzzy sets, vector $\boldsymbol{\gamma}$). The inference can be carried out using an array of transition matrices $\mathbf{\Omega}_{ij}$, with $i = 1, \ldots, N$, $j = 1, \ldots, P$ and size $Q \times M$. An equivalent HFS to this system will be the one shown in fig. 5. Two FLUs replace the 3-ISO FS. The output of the first block is modeled by a vector $\boldsymbol{\beta}$, defined over a space with K possible fuzzy sets. The value of K will be discussed later. Both systems are defined by its transition matrices, $\mathbf{\Omega}^A_i$ ($Q \times K$ size, $i = 1, \cdots, P$), and $\mathbf{\Omega}^B_j$ ($K \times M$ size, $j = 1, \cdots, N$).

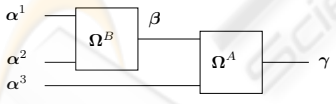

Figure 5: Equivalent HFS.

There is a relation between the matrices of the 3-ISO and the hierarchical systems:

$$\mathbf{\Omega}^A_j \mathbf{\Omega}^B_i = \mathbf{\Omega}_{ij} \tag{12}$$

This relation can be rewritten using block matrices as

$$\begin{pmatrix} \mathbf{\Omega}^A_1 \\ \vdots \\ \mathbf{\Omega}^A_P \end{pmatrix} \begin{pmatrix} \mathbf{\Omega}^B_1 & \cdots & \mathbf{\Omega}^B_N \end{pmatrix} = \begin{pmatrix} \mathbf{\Omega}_{11} & \cdots & \mathbf{\Omega}^{N1} \\ \cdots & & \cdots \\ \mathbf{\Omega}_{1P} & \cdots & \mathbf{\Omega}^{NP} \end{pmatrix}$$

$$\mathbf{\Omega}^A_{T\,[QP \times K]} \mathbf{\Omega}^B_{T\,[K \times MN]} = \mathbf{\Omega}_{T\,[QP \times MN]} \tag{13}$$

The solutions of this system are determined by the rank of matrix $\mathbf{\Omega}_T$. It will also determine the number of sets in the transition space, as the minimum number of columns of matrix $\mathbf{\Omega}^A_T$ is the rank of matrix $\mathbf{\Omega}_T$. Specifically $K \geq \text{rank}(\mathbf{\Omega}_T)$. We choose the equality to obtain the minimum-size matrices. If a decomposition of matrix $\mathbf{\Omega}_T$ according to eq. (13) is feasible, the hierarchical system determined by matrices $\mathbf{\Omega}^A_i$ and $\mathbf{\Omega}^B_j$ will be totally equivalent to the original system.

3.2 Optimizing the HFS

The decomposition process described in section 3.1 assumes that the inputs are arranged in a particular order, but this order is somehow arbitrary. A different arrangement will give rise to a different matrix $\mathbf{\Omega}_T$, and therefore to a different decomposition in FLUs.

For a 3ISO system, as the one in the previous section, there are three possible arrangements; each of them gives rise to a different system (see fig. 5):

- $[x_1, x_2, x_3]$: x_3 will be the external input to the second FLU (this is the one indicated in Fig. 5).
- $[x_1, x_3, x_2]$: x_2 will be the external input to the second FLU.
- $[x_2, x_3, x_1]$: x_1 will be the external input to the second FLU.

We can also think in other possibilities, such as $[x_2, x_1, x_3]$, but the final HFS will be the same one as in the first case. Assuming that the rearranged inputs are $[x'_1, x'_2, x'_3]$, with size N'_1, N'_2 and N'_3 the total number of rules in the 3ISO system and in the HFS will be $N_{3ISO} = N'_1 \times N'_2 \times N'_3$ and $N_{HFS} = N'_1 \times N'_2 + K \times N'_3$, being $K = \text{rank}(\mathbf{\Omega}_T)$. In order to have the minimum number of rules, we must choose the arrangement of the inputs that minimize N_{HFS}.

3.3 An Example

Suppose a fuzzy system with 3 inputs x, θ and $\dot{\theta}$ and one output y. Each input and the output are defined as a linguistic variable with 3 possible fuzzy sets: Positive (P), Zero (Z) and Negative (N). The rule base for this system is on table 1.

The inference can be carried out using FITM. The system will be a 3ISO system as the one in fig. 4. The output will be the centroid of

$$\boldsymbol{\gamma} = \left(\sum_i \sum_j \mathbf{\Omega}_{ij} \alpha^i_2 \alpha^j_3 \right) \boldsymbol{\alpha}_1$$

being $\boldsymbol{\gamma}$ the output vector, and $\boldsymbol{\alpha}_i$ the input vectors. Accepting that the inputs are crisp values, these vectors are defined as

$$\boldsymbol{\alpha}_i = [P(x_i)\ Z(x_i)\ N(x_i)]^T$$

$$\mathbf{C}^T = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix} \tag{14}$$

Table 1: Rule base for the example.

x	θ	$\dot{\theta}$	y
P	P	P	P
P	P	Z	N
P	P	N	N
P	Z	P	N
P	Z	Z	P
P	Z	N	Z
P	N	P	Z
P	N	Z	N
P	N	N	N
Z	P	P	Z
Z	P	Z	N
Z	P	N	N
Z	Z	P	N
Z	Z	Z	P
Z	Z	N	P
Z	N	P	P
Z	N	Z	P
Z	N	N	N
N	P	P	P
N	P	Z	P
N	P	N	N
N	Z	P	P
N	Z	Z	Z
N	Z	N	Z
N	N	P	Z
N	N	Z	N
N	N	N	N

As indicated before, the activation matrices $\mathbf{R_x}$, $\mathbf{R}_\theta$ and $\mathbf{R}_{\dot{\theta}}$ are the 3×3 identity matrix. The selection matrix $\mathbf{C}$ is on eq. (14). The resulting transition matrices are (if an ordering $[x, \theta, \dot{\theta}]$ is assumed)

$$\mathbf{\Omega}_{11} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \mathbf{\Omega}_{21} = \mathbf{\Omega}_{12} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

$$\mathbf{\Omega}_{31} = \mathbf{\Omega}_{23} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad \mathbf{\Omega}_{22} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{\Omega}_{32} = \mathbf{\Omega}_{13} = \mathbf{\Omega}_{33} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Once the whole system is defined we can make the HFS decomposition. To build the $\mathbf{\Omega}_T$ matrix in eq. (13) we have three different ways of ordering the inputs:$[x, \theta, \dot{\theta}]$, $[x, \dot{\theta}, \theta]$ and $[\theta, \dot{\theta}, x]$. So, three different $\mathbf{\Omega}_T$ matrices can be built:

$$\mathbf{\Omega}_T^1 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{\Omega}_T^2 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\mathbf{\Omega}_T^3 = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}$$

the ranks of which are rank($\mathbf{\Omega_T^1}$) $= 6$, rank($\mathbf{\Omega_T^2}$) $= 6$ and rank($\mathbf{\Omega_T^3}$) $= 5$. The number of rules in the original system is $3 \times 3 \times 3 = 27$. In the HFS we have $3 \times 3 +$ rank($\mathbf{\Omega_T}$) $\times 3$ rules. So we choose the third matrix (the one with smaller rank) to build the HFS.

According to eq. (13) we may find two matrices $\mathbf{\Omega}_T^A$ and $\mathbf{\Omega}_T^B$ so that $\mathbf{\Omega}_T^A \mathbf{\Omega}_T^B = \mathbf{\Omega}_T$. One way to do it is using a LU factorization, and use some further algebra to eliminate redundant lines. By doing so, we obtain matrices $\mathbf{\Omega}_T^A$ and $\mathbf{\Omega}_T^B$ and the transition matrices for each FLU will be

$$\mathbf{\Omega}_1^A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{\Omega}_2^A = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{\Omega}_3^A = \begin{pmatrix} 1 & 1 & 0 & -1 & -1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{\Omega}_1^B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \mathbf{\Omega}_2^B = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{\Omega}_3^B = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

The inference using FITM can be done as

$$\boldsymbol{\gamma} = \left(\sum_i \mathbf{\Omega}_i^A \alpha_1^i \right) \boldsymbol{\beta}$$

$$\boldsymbol{\beta} = \left(\sum_j \mathbf{\Omega}_j^B \alpha_3^i \right) \boldsymbol{\alpha}_2$$

The system represented as a HFS is totally equivalent to the one we departed from.

3.4 General Case

When the number of inputs is greater than 3, the HFS can be obtained in a very similar way to the 3-ISO system. Suppose a generic MISO system with N inputs $x_1, \cdots, x_N$. The system can be represented using transition matrices

$$\mathbf{\Omega}_{l_2 \cdots l_N} \; l_2 = 1, \cdots, M_2 \; \cdots \; l_N = 1, \cdots, M_N$$

being M_i the dimension of the i-th input. This system is equivalent to a system with N inputs and another system with two inputs, as shown in fig. 6.

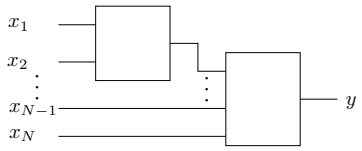

Figure 6: First step in hierarchical decomposition of a MISO-FS.

To obtain the transition matrices of each system, the following equation must be solved

$$\mathbf{\Omega}^A_{l_3 \cdots l_N} \mathbf{\Omega}^B_{l_2} = \mathbf{\Omega}_{l_1 \cdots l_N} \tag{15}$$

Written in matrix form

$$\mathbf{\Omega}^A_T \mathbf{\Omega}^B_T = \mathbf{\Omega}_T$$

$$\begin{pmatrix} \mathbf{\Omega}^A_{11\cdots 1} \\ \mathbf{\Omega}^A_{21\cdots 1} \\ \vdots \\ \mathbf{\Omega}^A_{M_3 \cdots M_N} \end{pmatrix} \begin{pmatrix} \mathbf{\Omega}^B_1 & \cdots & \mathbf{\Omega}^B_{M_2} \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{\Omega}_{11\cdots 1} & \cdots & \mathbf{\Omega}_{M_2 1 \cdots 1} \\ \mathbf{\Omega}_{12\cdots 1} & \cdots & \mathbf{\Omega}_{M_2 2 \cdots 1} \\ \vdots & & \vdots \\ \mathbf{\Omega}_{1 M_3 \cdots M_N} & \cdots & \mathbf{\Omega}_{M_2 M_3 \cdots M_N} \end{pmatrix} \tag{16}$$

Matrices $\mathbf{\Omega}^B_{l_2}$ can be easily obtained from this equation. The same procedure is applied on the remaining MISO system, until only two inputs are left.

As in the 3ISO case, the proper election of the ordering of the inputs will give a better size of the rule base of each FLU. This size is related again with the rank of matrix $\mathbf{\Omega}_T$. When this optimization process is carried out, the final HFS system can be any of the types introduced in section 1.1: serial, parallel or hybrid.

4 CONCLUSIONS

In this paper we have shown a simple procedure to convert a large MISO FS into a serial HFS. Such a conversion leads to a number of smaller rule bases than the original, which, in turn, may facilitate human interpretation and/or fine tune the rule bases out of examples.

REFERENCES

Aja-Fernández, S. and Alberola-López, C. (2004a). Fast inference in SAM fuzzy systems using transition matrices. *IEEE Trans. Fuzzy Systems*, 12(2):170–182.

Aja-Fernández, S. and Alberola-López, C. (2004b). Fuzzy hierarchical systems wih FITM. In *Proc. of FUZZ-IEEE'04*, Budapest, Hungary.

Aja-Fernández, S. and Alberola-López, C. (2005). Fast inference using transition matrices: An extension to non-linear operators. *IEEE Trans. Fuzzy Systems*, 13(4):478–490.

Golub, G. and Loan, C. V. (1996). *Matrix Computations*. The Johns Hopkins University Press, Baltimore, ME.

Holve, R. (2003). Rule generation for hierarchical fuzzy systems. In *NAFIPS '97. 1997 Annual Meeting of the North American Fuzzy Information Processing Society*, pages 444–449, Syracuse, NY, USA.

Horáček, P. and Binder, Z. (1997). Hierarchical fuzzy controllers. *Annual Reviews in Control*, 21:93–101.

Kikuchi, H., Otake, A., and Nakanishi, S. (1998). Functional completeness of hierarchical fuzzy modeling. *Information Sciences*, 110:51–60.

Kosko, B. (1997). *Fuzzy Engineering*. Prentice-Hall International, New Jersey.

Lee, M. L., Chung, H., and Yu, F. (2003). Modeling of hierarchical fuzzy systems. *Fuzzy Sets and Systems*, 138:343–361.

Raju, G. and Zhou, J. (1993). Adaptive hierarchical fuzzy controller. *IEEE Trans. on System, Man and Cybernetics*, 23(4):973–980.

Raju, G., Zhou, J., and Kisner, R. A. (1991). Hierarchical fuzzy control. *Int. J. Control*, 54:1201–1216.

Wang, L. X. (1998). Universal approximation by hierarchical fuzzy systems. *Fuzzy Sets and Systems*, 93:223–230.

Yager, R. (1993). On a hierarchical structure for fuzzy modeling and control. *IEEE Trans. on System, Man and Cybernetics*, 23(5):1189–1197.

Zadeh, L. A. (1996). Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Systems*, 4(2):103–111.

Zeng, X. and Keane, J. (2005). Approximation capabilities of hierarchical fuzzy systems. *IEEE Trans. Fuzzy Systems*, 13(5):659–672.