An Experimental Study of Gradient Descent Learning of Fuzzy Rule Bases Aimed at the Approximation of One Variable Functions

Mohand Si-Fodil, Patrick Siarry and Jean-Luc Tyran*
Ecole Centrale Paris
Grande Voie des Vignes
92295 Châtenay-Malabry, France
Phone: +33 1 41 13 13 05, Fax: +33 1 41 13 14 37
Email: sifodilm@cti.ecp.fr, siarry@lpi.ecp.fr
*EDF, Direction des Etudes et Recherches
Département CCC
6, Quai Watier, 78401 Chatou, France
Email: Jean-Luc.Tyran@der.edfgdf.fr

ABSTRACT: In fuzzy control, the choice of membership functions has a direct influence on the way the system will approach its output function. The most commonly used input membership functions are the triangular functions, the trapezoidal functions, the bell-shaped functions... In our work we try to approach a target output function. An optimization using a gradient descent method is performed to adjust the parameters of the various input membership functions tested. In this paper, a comparative survey of approximations thus achieved is exposed and we show the impact of an arbitrary choice of a rule base of which parameters are optimized as the system performs a "defuzzification".

KEYWORDS: Fuzzy logic, Fuzzy control, Optimization, Function approximation, Sugeno model.

INTRODUCTION

A fuzzy system : $F : IR^n \longrightarrow IR^p$ puts in memory Kosko (1992) the "inferences" by using its premises and tries to get the most adapted conclusion while approximating a function. The chosen input membership functions affect in various, more or less adapted, ways the output function F(x) of the system. F(x) can be in the form Huweindick (1999) k_1+k_2x.

In our study, we tried to approach a function f(x), considered as the output of a fuzzy system. An optimization using a gradient descent method was performed to adjust the parameters of the membership functions used. This gradient descent Guély (1993), Jang (1995, 1997) and Babuska (1998) changes the aspect of the output membership function allowing to stabilize the system more quickly and in a better way. It also appears that the error decreases with the number of the input fuzzy sets used.

CHOICE OF THE FUZZY SETS AND APPROXIMATION

A fuzzy system establishes an output of Sugeno type, while assembling all the rules having the same conclusion to make one single output. It proceeds by connecting all the premises by means of operators " T_norm " and / or " T_conorm ". The most commonly used operators are Min. and Max. Zadeh (1965) defined as follows :

$$T\_conorm : F_{A \grave{E} B}(x) = max.(F_A(x), F_B(x))$$
$$T\_norm : F_{A \c{C} B}(x) = min.(F_A(x), F_B(x))$$

In our system, the defuzzification is given by F(x) Kosko (1992) defined as follows :

$$F(x) = centroid\left(\sum_{j=1}^{n} w_j a_j(x) B_j\right)$$

$a_j(x)$ stands for the membership of x to A_j Ì IR^n, and is defined by :

$$a_j(x) = \prod_i a_j^i(x_i)$$

$a_j(x) = a^1_j(x_1)\, a^2_j(x_2)$ .... $a^n_j(x_n)$. Usually the w_i are chosen such that :
$w_1 = w_2 = ... = w_n = 1$. In this work, we set w_i ³ 0.

X Ì IR^n and Y Ì IR^p represent the definition domain of the rule (called the fuzzy partition or fuzzy set). It appears that the number of pieces which define the function grows up with the number of subsets, the discretization is then finer and allows a better approximation of the target function (see on Figures 1 and 2).

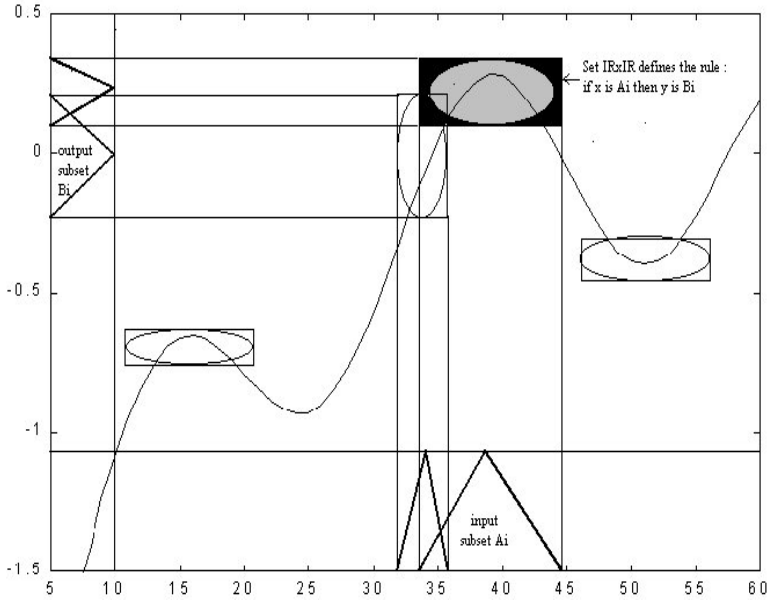

Figure 1: Rough approximation for the Inference in a fuzzy system.

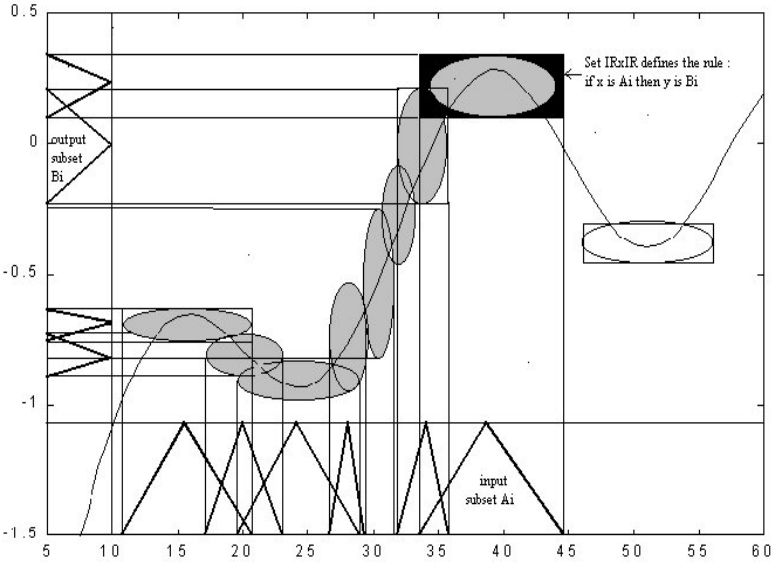

Figure 2: Finer approximation for the Inference in a fuzzy system.

MEMBERSHIP FUNCTIONS AND PARTITION OF THE REFERENTIAL

We describe first the case of triangular membership functions (see on Figure 3) :

$$f_{ij}(X_j)=\begin{cases} 1-\dfrac{X_j-a_{ij}}{b_{ij}}, & if\ X_j\in[l_j \quad m_j] \\ 1-\dfrac{a_{ij}-X_j}{b_{ij}}, & if\ X_j\in[m_j \quad r_j] \\ 0 \quad , & else \end{cases}, \qquad bj=\begin{cases} mj-lj & if\ X_j\in[l_j \quad m_j] \\ rj-mj & if\ X_j\in[m_j \quad r_j] \end{cases}$$

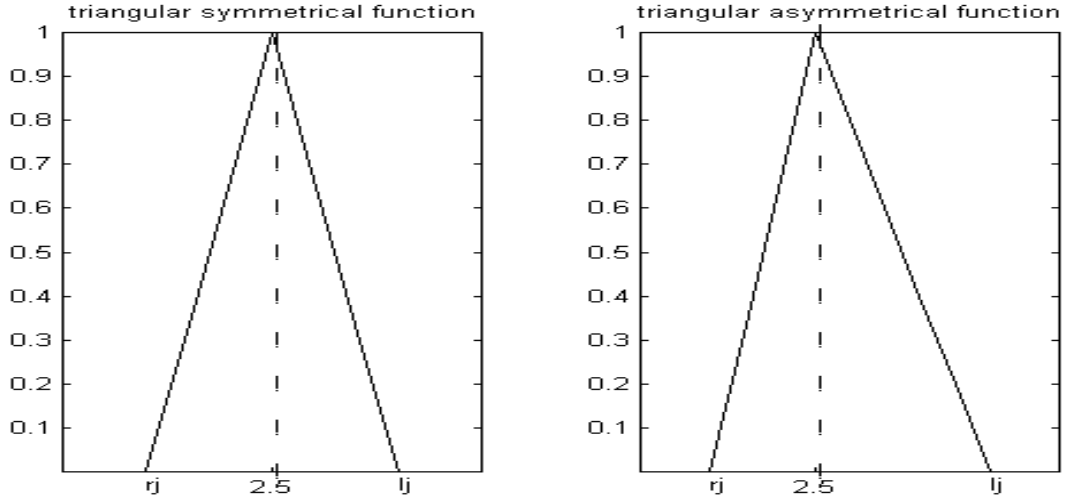

Figure 3: Symmetrical and asymmetrical triangular functions.

The trapezoidal functions are as follows (see on Figure 4)

$$f_{ij}(X_j)=\begin{cases} 1-\dfrac{X_j-a_{ij}}{b_{ij}}, & if\ X_j\in[l_j \quad m_j] \\ 1, & if\ X_j\in[m_j \quad m'_j] \\ 1-\dfrac{a_{ij}-X_j}{b_{ij}}, & if\ X_j\in[m'_j \quad r_j] \\ 0, & else \end{cases} \qquad bj=\begin{cases} mj-lj & si\ X_j\in[l_j \quad m_j] \\ rj-m'j & si\ X_j\in[m'_j \quad r_j] \end{cases}$$

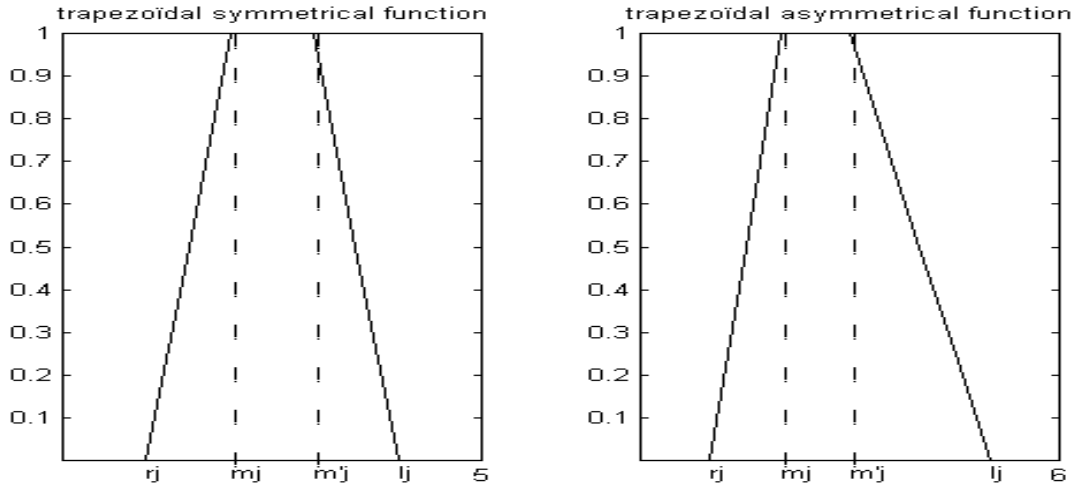

Figure 4: Symmetrical and asymmetrical trapezoidal functions.

GAUSSIAN FUNCTIONS:

$f_{ij}(X_j)=\exp\left\{-\left(\frac{x-m_j}{d_j}\right)^2\right\}$ (see on Figure 5), where m_j stands for the mean and d_j for the variance.

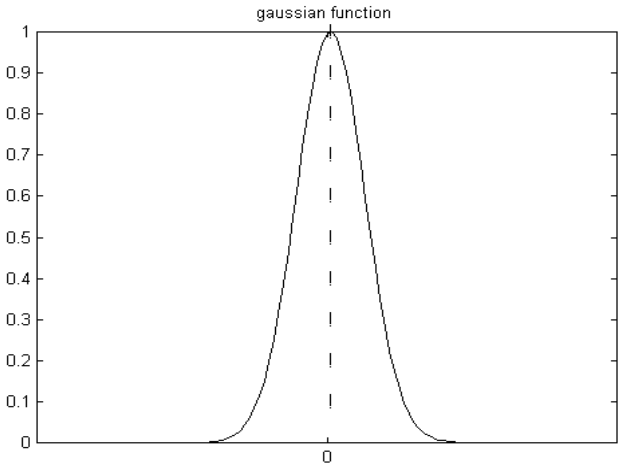

Figure 5: Gaussian function.

OPTIMIZATION OF FUNCTION PARAMETERS

Parameters to be adjusted are : a_{ij}, b_{ij}, and w_i. The gradient descent method applies the following training rule to all parameters :

$w(t+1)=w(t)+d_t\frac{\partial e}{\partial w}$, d_t stands for the coefficient of the gradient descent.

The aim is therefore the minimization of the mean quadratic error :

$e(x)=\frac{1}{2}\left(f(x)-F(x)\right)^2$. $f(x)$ = test function ; F(x) = output function

with : $F(x) = \frac{\sum_{j=1}^{n} w_j\, a_j(x)\, S_j\, c_j}{\sum_{j=1}^{n} w_j\, a_j(x)\, S_j}$, S_j is the area.

$$\frac{\partial e}{\partial w}=\frac{\partial e}{\partial F}\frac{\partial F}{\partial a_j}\frac{\partial a_j}{\partial w_j};$$

$$\frac{\partial e}{\partial c_j}=\frac{\partial e}{\partial F}\frac{\partial F}{\partial c_j},$$

$$\frac{\partial e}{\partial F}=-(f(x)-F(x))$$

$$\frac{\partial F}{\partial a_j}=\frac{\left(\sum_{i=1}^{n} a_j(x)\, S_i\right)\left(S_j c_j\right)-S_j\left(\sum_{i=1}^{n} a_j(x)\, S_i\, c_i\right)}{\left(\sum_{i=1}^{n} a_j(x)\, S_i\right)^2}$$

$$\frac{\P F}{\P a_j} = \frac{\left[c_j - F(x)\right] S_j}{\sum_{i=1}^{n} a_j(x) S_i} = \left[c_j - F(x)\right] \frac{p_j(x)}{a_j(x)}$$

LEARNING RATES FOR TRIANGULAR FUNCTIONS:

- $$m_j(t+1) = \begin{cases} m_j(t) - d_t(x)\left[c_j - F(x)\right]\dfrac{p_j(x)}{a_j(x) l_j}, & \text{if } m_j - l_j < x < m_j \\ m_j(t) + d_t(x)\left[c_j - F(x)\right]\dfrac{p_j(x)}{a_j(x) r_j}, & \text{if } m_j < x < m_j + r_j, \\ m_j(t), & \text{else} \end{cases}$$

- $$l_j(t+1) = \begin{cases} l_j(t) + d_t(x)\left[c_j - F(x)\right]\dfrac{p_j(x)}{a_j(x)}\dfrac{m_j - x}{l_j^2}, & \text{if } m_j - l_j < x < m_j, \\ l_j(t), & \text{else} \end{cases}$$

- $$r_j(t+1) = \begin{cases} r_j(t) + d_t(x)\left[c_j - F(x)\right]\dfrac{p_j(x)}{a_j(x)}\dfrac{x - m_j}{r_j^2}, & \text{if } m_j < x < m_j + r_j, \\ r_j(t), & \text{else} \end{cases}$$

where $\delta_t(x)$ stands for a function of the two variables t and x.

The figure 6 shows how the shape of the input function changes according to the learning triangular parameters. It seems also that the output function changes according to its parameters and according to the input function learning. The a, b and r parameters have been set respectively to 4, 2 and 5. To begin the learning parameters of the input function and the output function can hold any value belonging to their set domain.
After 156 iterations the stop test (error threshold) is met, then the learning program is stopped. New parameters which appear on the right side of the figure, then we obtain the new shape of the input-output function. This approximation corresponds to one input one output function with one rule in the form : *If x is A then y is B.* The relative error of this approximation is shown on figure 7. It decreases according to the number of iterations.

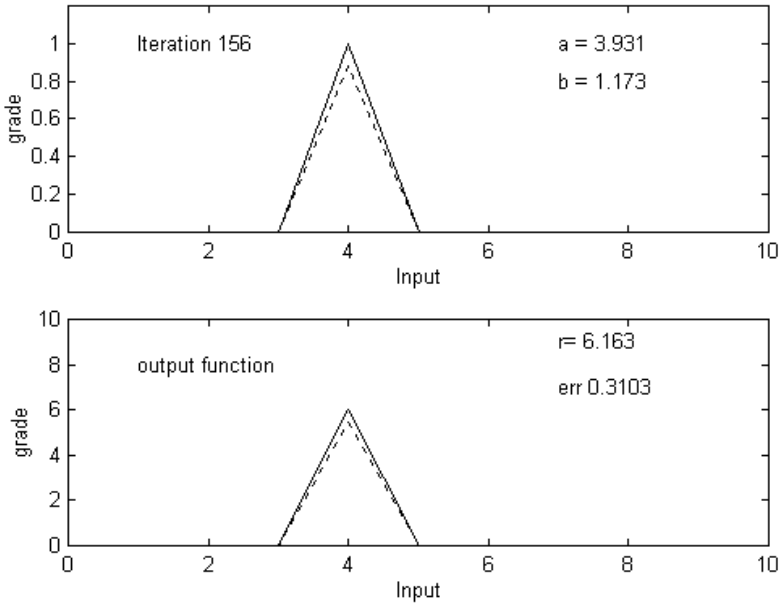

Figure 6: Learning input-output triangular functions.

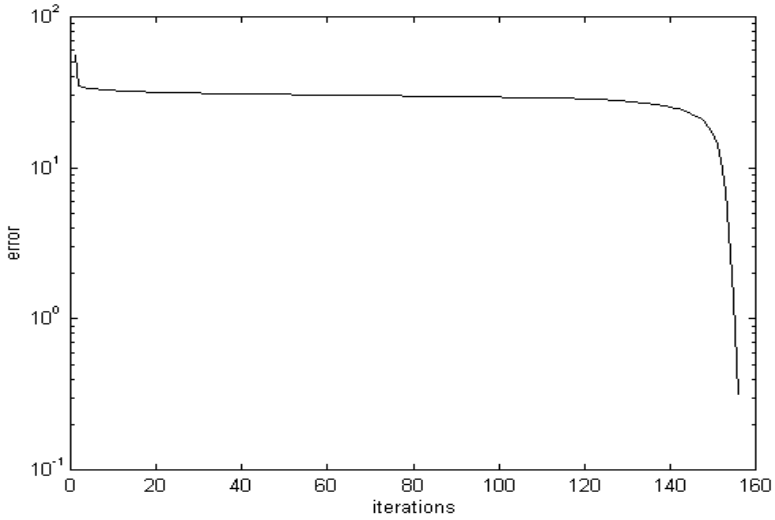

Figure 7: Evolution of the error during the learning (in semilog scale).

SIMULATION RESULTS

TEST FUNCTION : $f(x) = x+100* (\exp(-100 * (x-0.7)^2))* \sin(125/(x+1.5)) / (x+0.1)$

Various simulations have been performed to validate our work. The chosen function presents several picks (see on Figure 8).

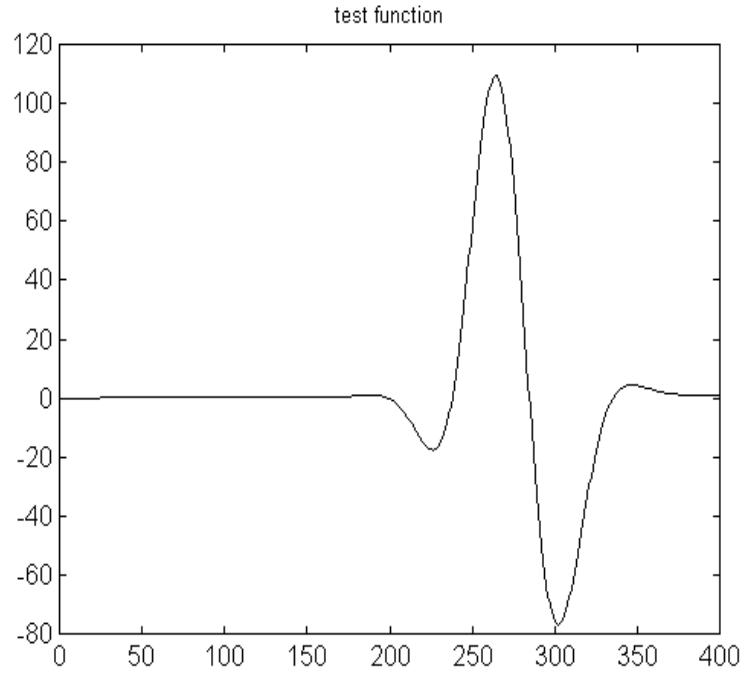

Figure 8: Test function.

COMMENTS ON THE RESULTS:

The figure 9 shows a rough approximation of the function, the referential being divided into 4. Even if the subsets were well chosen, they could not cover the whole area, thus the picks are still hidden to the approximation. In figure 10, the approximation seems better but not yet the one we expected: adding two subsets in the referential induces a better covering of the referential. In figure 11,and figure 12, the algorithm begins to recognize the function' extrema and to approximate them well.
Therefore, with a referential divided into 10 subsets (see on Figure 13), the approximation is better than in the figure 11 in spite of using the same referential, thanks to the different input function (gaussian). In figure 14, the approximation is almost indistinguishable from the test function, using just 12 subsets as in figure 12.

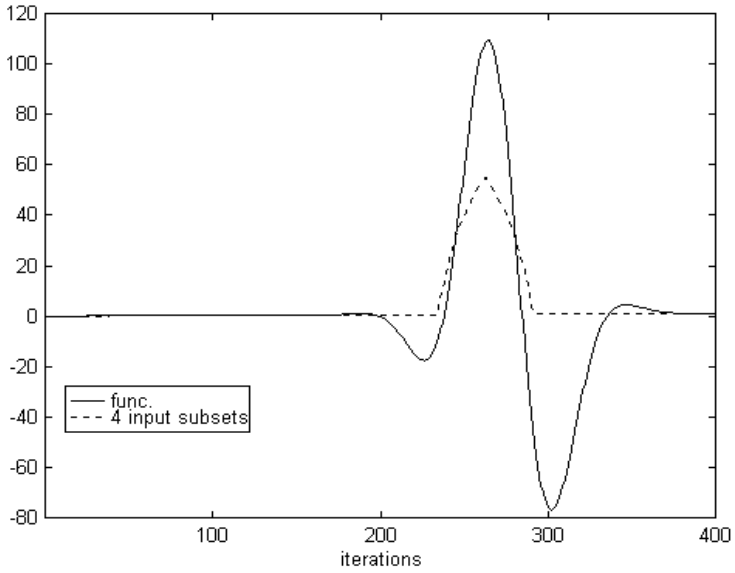

Figure 9: Approximation of f using 4 triangular input functions.

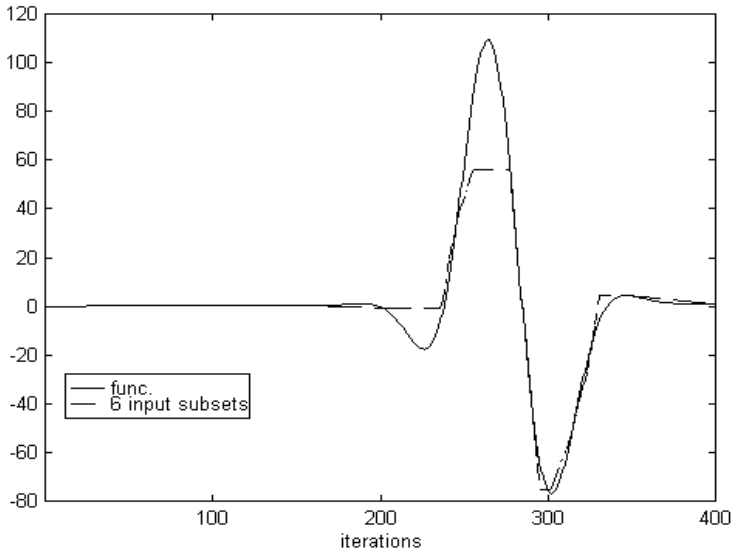

Figure 10: Approximation of f using 6 triangular input functions.

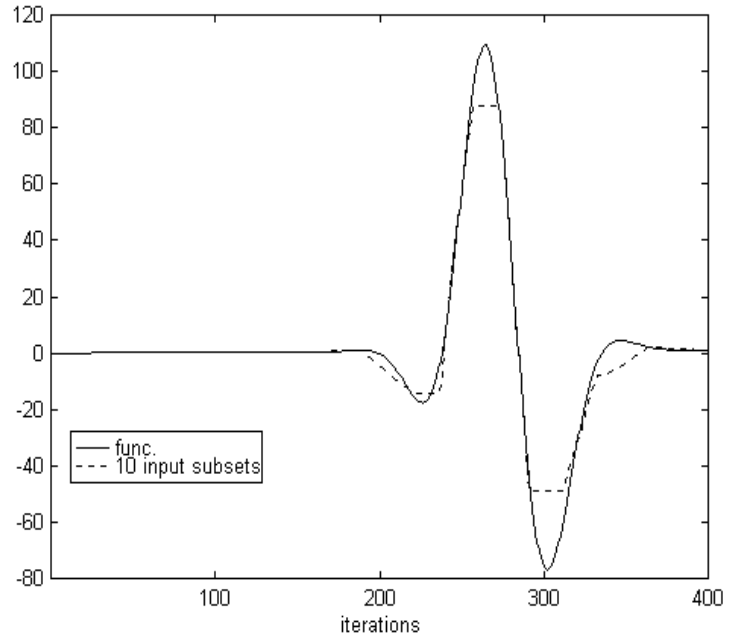

Figure 11: Approximation of f using 10 triangular input functions.

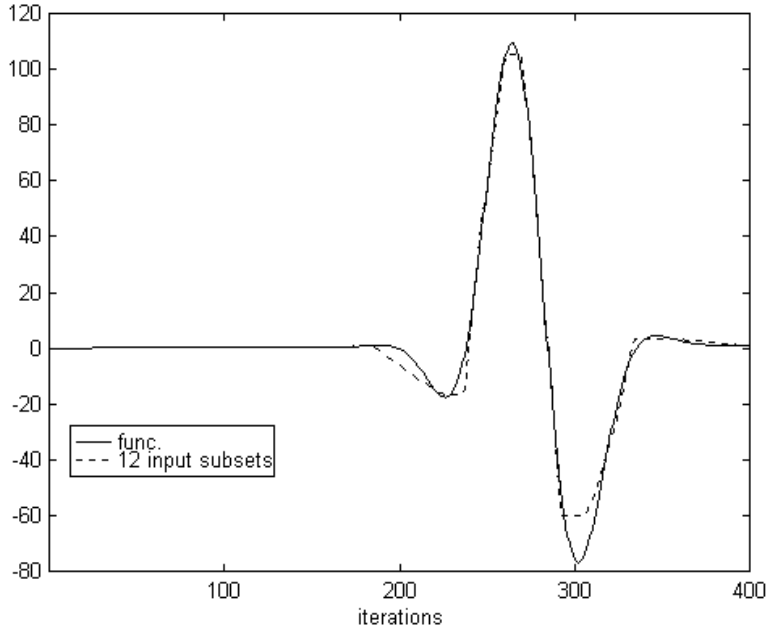

Figure 12: Approximation of f using 12 triangular input functions.

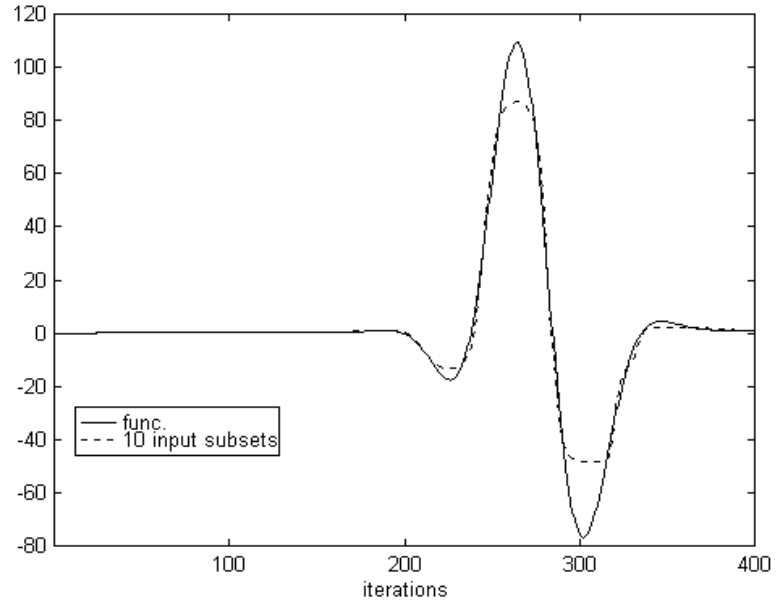

Figure 13: Approximation of f using 10 gaussian input functions.

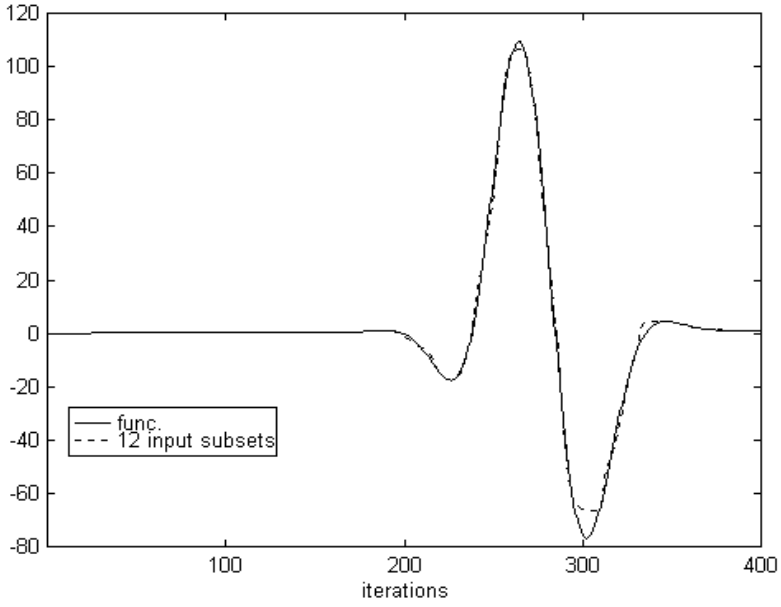

Figure 14: Approximation of f using 12 gaussian input functions.

CONCLUSION:

This work was aimed at the minimization of the mean quadratic error related to the learning of fuzzy rule bases. As it was shown in Mitaim (1998) it is easy to approximate a function when the number of input subsets is conveniently chosen. However, our study established that the right input membership functions, allowing to approximate as well as possible the output function, must be "manually" chosen. Results obtained for any kind of one variable output functions are quite satisfactory.

REFERENCES:

Babuska, R, 1998, "Fuzzy Modeling for Control", Kluwer Academic Publisher.

Guély, François ; Siarry, Patrick, 1993, "Gradient Descent Method for Optimizing Various Fuzzy Rule Bases", Proceedings of the Second IEEE International Conference on Fuzzy Systems FUZZ-IEEE'93, San Francisco CA, Vol. 2, pp. 1241-1246.

Huweindick, O ; Brockmann, W, 1999, "Function Approximation With Decomposed Fuzzy systems", Fuzzy Sets and Systems, Vol. 101, pp. 273-286.

Jang, Jhy-Shing Roger ; Sun, Chuen-Tsai, 1995, "Neuro-Fuzzy Modeling and Control", IEEE 83 Vol 3, pp 378-406.

Jang, Jhy-Shing Roger, 1997, "Neuro-Fuzzy Modeling and Soft Computing", Matlab Curriculum Series, Prentice Hall, Upper Saddle River, NJ, USA.

Kosko, Bart, 1992, "Neural Networks And Fuzzy Systems, A dynamical systems approach to machine intelligence", Prentice-Hall International Editions.

Kosko, Bart, 1994, "Fuzzy Systems as Universal Approximators", IEEE Transactions on Computers, Vol. 43, n° 11, pp. 1329-1333.

Mitaim, Sanya ; Kosko, Bart, 1998, "Neural Fuzzy Agents for Profile Learning and Adaptive Object Matching", Presence, Vol. 7, no. 5.

Sugeno, Michio ; Takagi, T, 1982, "A New Approach to Design of Fuzzy Controller", Advances in Fuzzy Sets Theory and Applications", Wang editor.

Zadeh, Lotfi, 1965, "Fuzzy Sets", Information and Control, Vol. 8, pp. 338-353.