Scientific Research and Essays Vol. 7(13), pp. 1415-1431, 9 April, 2012
Available online at http://www.academicjournals.org/SRE
DOI: 10.5897/SRE11.1960
ISSN 1992-2248 ©2012 Academic Journals

Full Length Research Paper

Nonlinear system identification using clustering algorithm and particle swarm optimization

TROUDI Ahmed*, HOUCINE Lassad and CHAARI Abdelkader

University of Tunisia, Higher School of Sciences and Techniques of Tunis (ESSTT) (Research unit C3S), 5 Avenue Taha Hussein Tunis 1008, Tunisia.

Accepted 17 February, 2012

The identification of nonlinear systems operating in a stochastic environment is an important problem in various discipline science and engineering. Fuzzy modeling and especially the T-S fuzzy model draw the attention of several researchers in recent decades this is due to their potential to approximate highly nonlinear behavior. An algorithm allowing the identification of the premise and consequent parameters intervening in the T-S fuzzy model at the same time and this starting from the minimization of four optimization criteria is used. A modification on both last optimization criterion is considered. Then an optimization method using the Particle Swarm Optimization method (PSO) is presented in this paper. Particle Swarm Optimization algorithm combined with the proposed algorithm is also presented. Simulation results on a nonlinear system and on a level control system shows that the proposed algorithm combined with the PSO algorithm gives results more effective than the proposed algorithm only more particularly to the level convergence and time computing.

Key words: Fuzzy identification, fuzzy clustering, Particle Swarm Optimization (PSO), nonlinear system, nonlinear identification, optimization problem.

INTRODUCTION

The development of a mathematical model making it possible to represent ˝ as well as possible ˝ the dynamic behavior of a complex real process represents a very important problem in the real world. In recent years, and with the evolution of technology, a significant effort has been given to modeling, identification and control of such systems. The T-S fuzzy model (Takagi and Sugeno, 1985; Grisales, 2007) is one of the best approaches to the representation of such a process. Indeed, the T-S fuzzy model can approximate highly nonlinear system into several locally linear subsystems. The identification problem in T-S fuzzy model can be summarized in two steps: structure identification and parameter estimation. On the other, T-S fuzzy model is composed of a premise-part and a consequent-part. The premise parameter identification consists determined the fuzzy partition matrix and the number of clusters (rules) needed to approximate the nonlinear system. While the consequent part identification, consists of estimated the parameters intervening in the conclusion of the rules of the T-S fuzzy model. Several techniques exist in the literature to identify the parameters involved in the T-S fuzzy model namely, the neuro-fuzzy technique (Babuska, 1998) and the clustering technique (Ahmed et al., 2008; Ahmed et al., 2011; Chen et al., 1998; Jang et al., 2007; Pingli et al., 2006; Xu et al., 2009; Zahid et al., 2003). In this work the clustering technique is used. In this context, several clustering algorithms have been proposed in the literature having to estimate the parameters of the T-S fuzzy model, we can quote as an example the Fuzzy C-Mean algorithm (FCM) (Dunn, 1974), the Gustafson Kessel algorithm (GK) (Gustafson et al., 1979) and the (GG) algorithm (Gath and Geva, 1989). Moreover, these algorithms are sensitive to noises or outliers. To overcome these disadvantages Krishnapuram and Keller have proposed the Possibilistic C-Means algorithm (PCM) (Krishnapuram and Keller, 1996) by abandoning the constraint of FCM and constructing a novel objective

*Corresponding author. E-mail: troudi.ahmed@yahoo.fr.

function. The PCM can deal with noisy data better than FCM, GK and GG. However, FCM, GK, and PCM are all only allowed the identification of the premise parameters while the consequent parameters are estimated using the least squares methods. In 1998 J.Q. Chen proposed another clustering algorithm allowing the identification of the premise and consequent parameters at the same time and this by using an iterative optimization method and starting from the minimization of four optimization criterion. This algorithm has many drawbacks such as convergence to local optima, sensitivity to noise and the aberrant point, also the computation time is very slow. In this paper we propose a new clustering algorithm to overcome this problem. This algorithm consists to introduce some modification to the optimization criteria and more particularly the last two criteria. Inspired by krishnapuram and Keller algorithm, we introduce two new objective functions into J.Q. Chen algorithm to replace the last two objective functions J_3 and J_4 in it. The algorithm which we have proposed overcomes the problems of sensitivity to noise and aberrant point better than FCM, GK, PCM and J.Q. Chen algorithms. However, FCM, GK, PCM and J.Q. Chen algorithm are all based on Euclidian distance in their objective function. In real world, the Euclidian distance is not complex enough to deal with more sophisticated problem. In order to introduce more robustness to the algorithm, Wu and Yang (2002) have proposed a non-Euclidean distance to replace the Euclidean distance in FCM algorithm. Inspired by Wu and Yang's algorithm, we introduce the new distance into J3 objective function to replace the Euclidean distance in it when calculating the fuzzy partition matrix. The new fuzzy clustering algorithm proposed becomes more robust; however, this algorithm does not solve the problems of convergence and the computation time. To overcome this problem, several solutions have been proposed in the literature. The idea of these techniques is to combine the clustering algorithms with other optimization techniques such as genetic algorithm (Goldberg, 1989) and particle swarm optimization (Adonyi et al., 2002; Biswal et al., 2009; Coelho and Herrera, 2007). Furthermore, we are presenting in this paper another approach for the identification of highly nonlinear systems and operating in a stochastic environment. This approach makes it possible to combine the algorithm which one proposed with the particle swarm optimization (PSO) algorithm. Indeed the particle swarm optimization is a global optimization technique. Thus the incorporation of local research capacity of clustering algorithms and the global optimization ability of PSO algorithm can give very good results. The effectiveness of this algorithm is tested on a nonlinear system and on a level control system. This paper is organized as follows: Next section gives a brief overview of the T-S fuzzy model. The criteria for fuzzy identification are presented in section 3. The proposed algorithm is introduced in section 4. The PSO for optimization of the T-S fuzzy model is presented in section 5. The simulations results are introduced in section 6. The validation model is presented in section 7. The application of the proposed algorithm to a level control system is described in section 8. And finally section 9 concludes the paper.

TAKAGI-SUGENO FUZZY MODEL

The implementation of a mathematical model for a complex real process operating in a stochastic environment draw the attention of many researchers in various disciplines of science and technology (Favier, 1982). In this context the use of traditional methods of modeling and identification in order to estimate the parameters of such a type of process can not satisfy the desired performance indices (speed, accuracy and stability). However, other techniques such as fuzzy logic and more particularly the T-S fuzzy fuzzy model showed a very good result for the identification of these types of processes. The T-S fuzzy model is based on a set of rules in which the consequent use of numeric variable rather than linguistic variables such as the Mamdani model. The consequent can be expressed as a constant, a polynomial or differential equation depending on the antecedent variables. The T-S fuzzy model makes it possible to approximate the nonlinear system into several locally linear subsystems. The identification of a T-S fuzzy model is made in two stages: adjustment of the parameters and structure optimization. The procedure of adjustment of the parameters is devoted to the estimation of a feasible set of parameters for a given structure. The optimization procedure aims at finding the optimal structure of all local models, the relevant premise variables and the suitable partition of the data space. The T-S fuzzy model consists of several fuzzy if-then rules that can be represented as follows:

$$R_i : if\ x_k \text{ is } A_i \text{ then } y_i = a_i^T x_k + b_i \qquad (1)$$

The ″ if ″ rule function define the premise part, while the ″ then ″ rule function constitute the consequent part of the T-S fuzzy model. $A_i \in R^n$ is a multidimensional antecedent fuzzy set, defined by its membership function $\mu_{A_i}(x_k)$. $x_k = \left[x_{k_1}, x_{k_2}, ..., x_{k_n}\right] \in R^n$, is the input vector of the fuzzy model; $a_i \in R^n, b_i \in R$: are the polynomial coefficients that form the consequent parameters of the ith rules, and $i=1,...,c$ (c : denote the numbers of rules in the rule base). $y_i \in R$: Is the rule output variable. Each antecedent fuzzy set A_i is associated with a membership function $\mu_{A_i}(x_k)$ described by the following equation:

$$\mu_{A_i}(x_k) = \exp\left[-\frac{1}{2}\frac{(x_k - m_{ki})^2}{\sigma_{ki}^2}\right]$$

Where m_{ki} and σ_{ki} are the center and the spread of the membership function, respectively. The output of the general nonlinear system is calculated as the average of output corresponding to the rules multiplied by the degree of fulfillment of the antecedent γ_i (Babuska and Verbruggen, 2003) of the form:

$$y = \frac{\sum_{i=1}^{c}\gamma_i(x_k)y_i}{\sum_{i=1}^{c}\gamma_i(x_k)} \quad (2)$$

With:

$$\gamma_i = \mu_{i1}(x_{k_1}).\mu_{i2}(x_{k_2}).\ \dots\ .\mu_{in}(x_{k_n}) \quad (3)$$

Introduce λ_i : the degree of achievement standard described by the following expression:

$$\lambda_i = \frac{\gamma_i(x_k)}{\sum_{i=1}^{c}\gamma_i(x_k)} \quad (4)$$

The estimated output of the Takagi-Sugeno fuzzy model can be expressed by:

$$y = \sum_{i=1}^{c}\lambda_i(x_k)\left[a_i^T x_k + b_i\right] \quad (5)$$

CRITERIAS FOR FUZZY IDENTIFICATION

Unlike to the other clustering algorithms which have been proposed in the literature, named the Fuzzy C-Means algorithm (FCM) (Dunn, 1974), the Gustafson and Kessel algorithm (GK) (Gustafson et al., 1979) and the Possibilistic C-Means algorithm (PCM) (Krishnapuram and Keller, 1996), which only allow that the premise Parameters identification intervening in the T-S fuzzy model. Chen et al. (1998) proposes another algorithm that allows the identification of premise and consequent parameters simultaneously. It is composed of fuzzy c-linear functions and Fuzzy C-Means clustering algorithm. Its obtaining requires the minimization of four optimization criteria. However, this algorithm has some disadvantages including: convergence to local optima, sensitivity with respect to noise these are due to the normalization constraint ($\sum_{i=1}^{c}\mu_{ik} = 1$) as well as the arbitrary choice of the third and fourth optimization criterion. To address this problem and in order to improve more robustness of the algorithm we tried to introduce modification to the optimization criteria which has been proposed, and more particularly on the last two criterions J_3 and J_4. Moreover we replaced the Euclidean distance by another non-Euclidean distance. This modification makes it possible to guarantee the robustness of the algorithm with respect to the noise and the aberrant points.

Optimization criteria

The minimization of the criterion J_1 allows the determination of the consequent parameters:

$$J_1 = \sum_{k=1}^{N}(y(k) - \sum_{i=1}^{c}\mu_{ik}y_i(k))^2 \quad (6)$$

The determination of the cluster centers v_i is obtained by minimizing the J_2 criterion:

$$J_2 = \sum_{k=1}^{N}(x_k - \sum_{i=1}^{c}\mu_{ik}v_i)^2 \quad (7)$$

The degree of membership μ_{ik} can be obtained by minimizing the following criterion:

$$J_3 = \sum_{i=1}^{c}(\mu_{ik})^{m_1}d_{ik}^2 + \sum_{i=1}^{c}\eta_i\sum_{k=1}^{N}(1-\mu_{ik})^{m_1} \quad : \quad (8)$$

Where N is the total number of observations, $d_{ik} = \sqrt{1-\exp(-\rho\left\|x_k - v_i\right\|^2)}$ is a non Euclidian distance from simple point x_k to the cluster center v_i , c is the number of clusters, $U = \left[\mu_{ik}\right]$ is a $c\times N$ matrix , denoted a fuzzy partition matrix. m_1 is a weighting exponent, $m_1 > 1$ and η_i is a suitable positive number. It should be noted that the first term demands that the distance between x_k to v_i be as low as possible, however the second term force μ_{ik} to be as large as possible. The concluding truth degree f_{ik} can be obtained by minimizing the J_4 criterion

$$J_4 = \sum_{i=1}^{c}(f_{ik})^{m_2}(\left|y(k) - \bar{x}k\theta_i\right|)^2 + \sum_{i=1}^{c}\beta_i\sum_{k=1}^{N}(1-f_{ik})^{m_2} \quad (9)$$
$$1 < m_2; i = 1,\dots,c; k = 1,\dots,N$$

Where $\bar{x}k = \left[x_k 1\right]$ and $\theta_i = \left[a_1^i \; a_2^i \; ... \; a_n^i \; b_i\right]^T$ are coefficients of consequence linear matrix, m_2 : is a weighting exponent for f_{ik} and β_i is a suitable positive numbers.

It should be noted that the unit X is partitioned in subsets S_i corresponds to c prototype v_i. Consequently, when x_k is an element of S_i, then μ_{ik} and f_{ik} are close to 1. In the contrary case, these two terms tend towards zero:

$$\begin{array}{l} \text{if } x_k \in S_i \text{, then } \mu_{ik} \text{ and } f_{ik} \text{ all neat to 1} \\ \text{if } x_k \notin S_i \text{, then } \mu_{ik} \text{ and } f_{ik} \text{ all near to 0} \end{array} \quad (10)$$

Thus J_1 and J_2 can be, respectively rewritten as follows:

$$J_1 = \sum_{k=1}^{N} (y(k) - \sum_{i=1}^{c} \mu_{ik} f_{ik} y_i(k))^2 \quad (11)$$

$$J_2 = \sum_{k=1}^{N} (x_k - \sum_{i=1}^{c} \mu_{ik} f_{ik} v_i) \quad (12)$$

Where (10) used on several occasions, equation (11) can be replaced by:

$$\begin{aligned} J_1 &= \sum_{k \in s_i} (y(k) - \mu_{ik} . f_{ik} . y_i(k))^2 \\ &\approx \sum_{k \in s_i} (\mu_{ik})^2 (f_{ik})^2 (y(k) - y_i(k))^2 \end{aligned} \quad (13)$$

Thus the vector of the consequent parameters $\theta_i (i = 1,...,c)$ can be obtained by minimization of the following criterion:

$$\begin{aligned} J_1 &= \sum_{k \in s_i} (\mu_{ik})^2 (f_{ik})^2 (y(k) - \bar{x}k . \theta_i)^2 \quad ; \\ &= \sum_{k=1}^{N} (\mu_{ik})^2 (f_{ik})^2 (y(k) - \bar{x}k \theta_i)^2 \end{aligned} \quad (14)$$

μ_{ik} and f_{ik} are calculated by minimizing J_3 and J_4 criterion respectively

Similarly J_2 can be expressed by the following form:

$$J_2 = \sum_{k=1}^{N} (\mu_{ik})^2 (f_{ik})^2 (x_k - v_i)^2 \quad (15)$$

Identification algorithm for premise and consequence parameters

The parameters identification intervening in the T-S fuzzy model is obtained by minimization of four optimization criteria J_1, J_2, J_3 and J_4 expressed by Equations (14), (15), (8) and (9) respectively.

Their minimization is complete in an iterative way and by using the following theorems:

Theorem 1: assume that v_i are fixed, then the minimization of the J_3 criteria gives for $i = 1,...,c$ and $k = 1,...,N$:

$$\mu_{ik} = \frac{1}{1 + \left(\frac{d_{ik}^2}{\eta_i}\right)^{\frac{1}{m_1 - 1}}} \quad (16)$$

Theorem 2: Assume that $\theta_i \; (i = 1,...,c)$ are fixed. The minimization of the J_4 criteria gives:

$$f_{ik} = \frac{1}{1 + \left(\frac{\left|y(k) - \vec{x}k\theta_i\right|}{\beta_i}\right)^{\frac{1}{m_2 - 1}}} \quad (17)$$

$i = 1,...,c$ and $k = 1,...,N$:

Theorem 3: Assume that μ_{ik} and f_{ik} are fixed, then coefficients θ_i are obtained from the minimization of the J_1 criteria:

$$\theta_i = (\bar{X}_k^T U_i^2 F_i^2 \bar{X} k)^{-1} (\bar{X}_k^T U_i^2 F_i^2 Y_k); \quad (i = 1,...,c) \quad (18)$$

With

$$\begin{aligned} &U_i = diag(\mu_{i1}, \mu_{i2}, ..., \mu_{iN}) \\ &and \\ &F_i = diag(f_{i1}, f_{i2}, ..., f_{iN}) \end{aligned}$$

Theorem 4: assume that μ_{ik} and f_{ik} are fixed, then the cluster centers of the prototypes v_i are obtained from the minimization of the J_2 criterion:

$$v_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^2 (f_{ik})^2 x_k}{\sum_{k=1}^{N} (\mu_{ik})^2 (f_{ik})^2}; \qquad i = 1,...,c \quad (19)$$

Based on the optimization conditions Equations (16) (17), (18) and (19), the identification algorithm of the premise and consequent parameters is obtained from an iterative optimization algorithm described later.

PROPOSED ALGORITHM

Given a data set $\left(x_k, y_k\right)$, the new clustering algorithm is given by the following steps:

Step 1: Initialization
• Choose the number of clusters
•Choose the weighting exponent m_1 and m_2
•Let err (0) be a large number

Step 2: initialize consequence parameter θ_i at random

• compute the fuzzy partition matrix U and the cluster centers v_i by the FCM algorithm

Step 3: compute η_i : $$\eta_i = \frac{\sum_{k=1}^{N} \mu_{ik}^{m_1} D_{ik}^2}{\sum_{k=1}^{N} \mu_{ik}^{m_1}}$$

Where $D_{ik}^2 = \|x_k - v_i\|$

Compute β_i : $$\beta_i = \frac{\sum_{k=1}^{N} f_{ik}^{m_2}(y(k) - \vec{x}k\theta_i)}{\sum_{k=1}^{N} f_{ik}^{m_2}}$$

Step 4: compute the new clustering distance d_{ik} :

$$d_{ik} = \sqrt{1 - \exp(-\rho\|x_k - v_i\|^2)}$$

Step 5: Update the fuzzy partition matrix:

$$\mu_{ik} = \left[1 + \left(\frac{1 - \exp\left(-\rho\|x_k - v_i\|^2\right)}{\eta_i}\right)^{\frac{1}{m_1 - 1}}\right]^{-1}$$

Step 6: Update the matrix f_{ik} by:

$$f_{ik} = \frac{1}{1 + \left(\frac{\left|y(k) - \vec{x}k\theta_i\right|}{\beta_i}\right)^{\frac{1}{m_2 - 1}}}$$

Step 7: Compute the cluster centers v_i :

$$v_i = \frac{\sum_{k=1}^{N} (\mu_{ik})^2 (f_{ik})^2 x_k}{\sum_{k=1}^{N} (\mu_{ik})^2 (f_{ik})^2}; \qquad i = 1, ..., c$$

Compute θ_i by:

$$\theta_i = (\overline{X}_k^T U_i^2 F_i^2 \overline{X}_k)^{-1} (\overline{X}_k^T U_i^2 F_i^2 Y_k); \quad (i = 1, ..., c)$$

Step 8: compute the err = $\|\theta - \theta(0)\|$

Where $\theta = \left[\theta_1, \theta_2, ..., \theta_c\right]$

Step 9: if err < ε_1 then turn to step10; else $\theta(0) = \theta$, turn to step 6.

Step 10: compute the estimated output and the sum of squared errors between the estimated output and the actual output by:

$$y(k) = \sum_{i=1}^{c} \mu_{ik} y_i(k)$$

$$error = \left(\frac{\sum_{k=1}^{N} (y(k) - y_i(k))^2}{N}\right)^{\frac{1}{2}}$$

$$rate = \left|\frac{error - error(0)}{error}\right|$$

Step 11: if $error < \varepsilon_2$ or $rate < \varepsilon_3$ then stop; else $c = c + 1$, $error(0) = error$, and turn to step 2

PSO FOR OPTIMIZATION OF THE T-S FUZZY MODEL

Fundamentals of the PSO approach

The Particle Swarm Optimization (PSO) is a stochastic optimization technique, it was originally developed by Kennedy and Eberhart (1995), it uses a population of candidates solution to develop an optimal solution of the problem. The degree of optimality is measured by a fitness function (Eberhart and Kennedy, 1995). Similar to genetic algorithm (Goldberg, 1989) the Particle Swarm Optimization (PSO) is an optimization technique based on a population where each member of population is considered as a particle, and each particle represents a

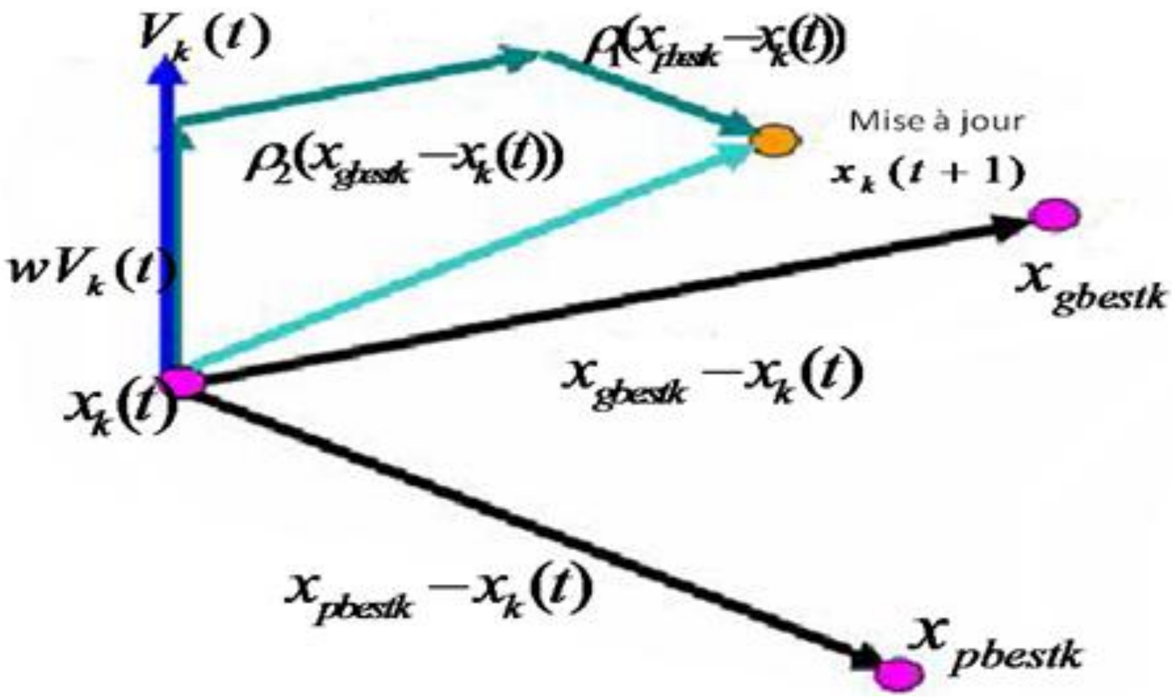

Figure 1. Geometric view for PSO algorithm.

solution of the current problem (Araujo and Coelho, 2008; Pan et al., 2006; Jang et al., 2007; Xu et al., 2009). Each particle in the algorithm is associated to a randomized velocity which enables it to move in the research space. The PSO algorithm does not have operators, such as crossover and mutation as in the genetic algorithm, in fact the PSO algorithm does not implement the survival of the suited individual, but it implements the simulation of social behavior individuals. From the algorithm, a swarm is randomly distributed in the search space, each particle also having a position and a random velocity (Figure1). Then, at each time instant, each particle is able to evaluate the quality of its position and to keep in memory its best performance. That is to say the best position it has achieved so far. Each particle in the PSO is able to query with a number of these neighbors. And get each of them its own best solution noted *pbest,* and then chose the best of the best performances in its possession noted *gbest.* The optimization procedure of PSO consists of each time instant to change the velocity of each particle flying the values of *pbest* and *gbest.* Acceleration is weighted by random terms, with separate random numbers being generated by acceleration toward of *pbest* and *gbest* locations, respectively. The implementation procedure of the PSO algorithm is summarized by the following steps (Araujo and Coelho, 2008; Jang et al., 2007).

Step 1: Initialize a population of particles with random positions and velocities using a uniform probability distribution.

Step 2: Compute the fitness value of each particle.

Step 3: Compare the fitness of each particle's with *pbest,* if the current value is better than *pbest,* then set the *pbest* value equal to the current value.

Step 4: Compare the fitness of each particle's with *gbest,* if the current value is better than *gbest,* then set the *gbest* value equal to the current value.

Step 5: update the position and velocity of the particle according to the Equations (21) and (20)

$$w\, v_d(k) + \rho_1\left(p_d(k) - x_d(k)\right) + \rho_2\left(p_g - x_d(k)\right) \quad (20)$$

$$x_d(k+1) = x_d(k) + v_d(k+1) \quad (21)$$

Where k is the current iteration number, $x_d = \left[x_{d_1}, x_{d_2}, x_{d_3}, \ldots, x_{d_k}, \ldots, x_{d_N}\right]^T$ represents the position of the ith particle, $v_d = \left[v_{d_1}, v_{d_2}, v_{d_3}, \ldots, v_{d_k}, \ldots, v_{d_N}\right]^T$ represents the velocity of the ith particle and $p_d = \left[p_{d_1}, p_{d_2}, \ldots, p_{d_N}\right]^T$, represents the best previous position (the position of which can give the best fitness value) of the i[th] particle. Index *g*: represents the index of the best particle in the population who can provide the best solution to the problem. ρ_1 and ρ_2 : represents two random variables defined as follows:

$$\begin{cases} \rho_1 = r_1 c_1 \\ \rho_2 = r_2 c_2 \end{cases}$$

r_1 and r_2 are two random variables between 0 and 1, c_1 and c_2 are two positive constants satisfying the following relationship: $c_2 + c_1 \le 4$

w : represents the factor of inertia proposed by Shi and Eberhart. This factor sets the ability to explore each particle which aims at improving the convergence of the method. Note that the size of this factor directly influences the size of the search space. Shi and Eberhart have shown that for $w \in [0.8, 1.2]$, can have a better convergence of the problem. The chosen of this factor also depends on the type of the intended application and the desired performance.

Step 6: until reaching the stopping criterion of the problem. It should be noted that the convergence of the algorithm towards the global optimal solution is not always guaranteed. For this reason it is necessary to define a stopping criterion for the algorithm. The stopping criteria used in most literature is the following:

i. The maximum number of iterations $nbIter_{\max}$ is reached.
ii. The change of speed is very low.
iii. The value of the fitness function is reached.

The position of particle, and its initial velocity must be chosen randomly following the uniform law, but to avoid the rapid movement of particle from one region to another in the search space, we fix a maximum speed $v_{\max}$ and we assume that the velocity of particle p_d at time k is equal $v_d(k)$, so that these two velocities satisfying the following conditions:

$$\begin{cases} v_d(k) = v_{\max} & si\ \ v_d(k) > v_{\max} \\ v_d(k) = -v_{\max} & si\ \ v_d(k) < -v_{\max} \end{cases} \tag{22}$$

In a subsequent a combination between the particle swarm optimization algorithm (PSO) and the modified clustering algorithm is used to build another approach called modified_algorithm-PSO to identify the premise and consequence parameters involved in the Takagi - Sugeno fuzzy model.

PSO combined with modified clustering algorithm

The optimal position is measured with said fitness function which defines the following optimization problem. This according to the following fitness function:

$$J_3 = \sum_{i=1}^{c} (\mu_{ik})^{m_1} d_{ik}^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{N} (1-\mu_{ik})^{m_1} \tag{3}$$

M : is a positive constant.

$J_{3(U,V)} = \sum_{k=1}^{N} \sum_{i=1}^{c} (\mu_{ik})^{m_1} d_{ik}^2 + \sum_{i=1}^{c} \eta_i \sum_{k=1}^{N} (1-\mu_{ik})^{m_1}$: is the objective function of the modified algorithm.

Where $V = [v_1, v_2, \cdots, v_i, \cdots v_c]$: represents the clusters vector, $U = [\mu_{ik}]$: represents the Fuzzy partition matrix.

Modified algorithm-PSO

Given a data set (x_k, y_k), the new clustering algorithm

is given by the following steps:

Step 1: Initialization
• Choose the number of clusters
• Choose the weighting exponent m_1 and m_2
• Let err (0) be a large number
• Give ρ_1 and ρ_2
• Set the weight of inertia: ω
• Set the size of the search space: *D*
• Initialize the 1^{st} particle generation.
• Initialize the position and velocity of each particle.
• Initialize the fitness function $f(x_k)$

Step 2: initialize consequence parameter θ_i at random

• compute the fuzzy partition matrix U and the cluster centers v_i by the FCM algorithm

Step 3: compute η_i :

$$\eta_i = \frac{\sum_{k=1}^{N} \mu_{ik}^{m_1} D_{ik}^2}{\sum_{k=1}^{N} \mu_{ik}^{m_1}}$$

Where $D_{ik}^2 = \left\| x_k - v_i \right\|$

Compute β_i :

$$\beta_i = \frac{\sum_{k=1}^{N} f_{ik}^{m_2} (y(k) - \vec{x}_k \theta_i)}{\sum_{k=1}^{N} f_{ik}^{m_2}}$$

Step 4: compute the new clustering distance d_{ik} :

$$d_{ik} = \sqrt{1-\exp(-\rho\left\|x_k - v_i\right\|^2)}$$

Step 5: Update the fuzzy partition matrix:

$$\mu_{ik} = \left[1+\left(\frac{1-\exp\left(-\rho\left\|x_k - v_i\right\|^2\right)}{\eta_i}\right)^{\frac{1}{m_1-1}}\right]^{-1}$$

Step 6: Compute the new value of the fitness function for each particle.

$$f(x_k) = \frac{M}{J_3(U,V)}$$

Step 7: Update the velocity and the position of each particle with.

$$v_d(k+1) = w\, v_d(k) + \rho_1\left(p_d(k) - x_d(k)\right) + \rho_2\left(p_g - x_d(k)\right)$$
$$x_d(k+1) = x_d(k) + v_d(k+1)$$

Step 8: Update the matrix f_{ik} by:

$$f_{ik} = \frac{1}{1+\left(\frac{\left|y(k)-\vec{x}k\theta_i\right|}{\beta_i}\right)^{\frac{1}{m_2-1}}}$$

Step 9: compute the cluster centers v_i :

$$v_i = \frac{\sum_{k=1}^{N}(\mu_{ik})^2(f_{ik})^2 x_k}{\sum_{k=1}^{N}(\mu_{ik})^2(f_{ik})^2}; \qquad i = 1,\ldots,c$$

Compute θ_i by:

$$\theta_i = (\overline{X}_k^T U_i^2 F_i^2 \overline{X}_k)^{-1}(\overline{X}_k^T U_i^2 F_i^2 Y_k); \quad (i = 1, \ldots, c)$$

Step 10: compute the err = $\left\|\theta - \theta(0)\right\|$

Where $\theta = \left[\theta_1, \theta_2, \ldots, \theta_c\right]$

Step 11: if err $< \varepsilon_1$ then turn to step10; else $\theta(0) = \theta$, turn to step 6.

Step 12: compute the estimated output and the sum of squared errors between the estimated output and the actual output by:

$$y(k) = \sum_{i=1}^{c} \mu_{ik} y_i(k)$$

$$error = \left(\frac{\sum_{k=1}^{N}(y(k) - y_i(k))^2}{N}\right)^{\frac{1}{2}}$$

$$rate = \left|\frac{error - error(0)}{error}\right|$$

Step 13: if $error < \varepsilon_2$ or $rate < \varepsilon_3$ then stop; else $c = c+1$, $error(0) = error$, and turn to step 2.

SIMULATION RESULTS

In this section, we will present two examples. All examples are nonlinear and difficult to be described by the ordinary method, so the fuzzy model presented in this paper is adopted.

Example 1: in the literature, there are several types of nonlinear systems described by different equations. In this paper, we are going to study a nonlinear system described by the following equation (Sastry et al., 1994):

$$y(k+1) = \frac{y(k)\,y(k-1)\,y(k-2)\,u(k-1)\left(y(k-2)-1\right)+u(k)}{1+y^2(k-1)+y^2(k-2)} + e(k) \qquad (24)$$

Where $y(k)$, $u(k)$ are the output and the input of the system respectively.
$e(k)$ is a linear noise given by the recurrent equation.

$$e_1(k+1) = \cos(\beta)e_1(k) + \sin(\beta)e_2(k)$$
$$e_2(k+1) = -\sin(\beta)e_1(k) + \cos(\beta)e_2(k)$$
$$e(k) = 0.5e_1(k) \qquad (25)$$
$$\beta = \frac{\pi}{6}$$

This system was proposed by Narendra and Parthasarathy (1990) in the context of neural networks modeling. Boukhirs (1998) and more recently Verdult (2002) used this same system to show the capabilities of approximation of a Takagi-Sugeno multiple models.

In this case, we present the simulation results concerning

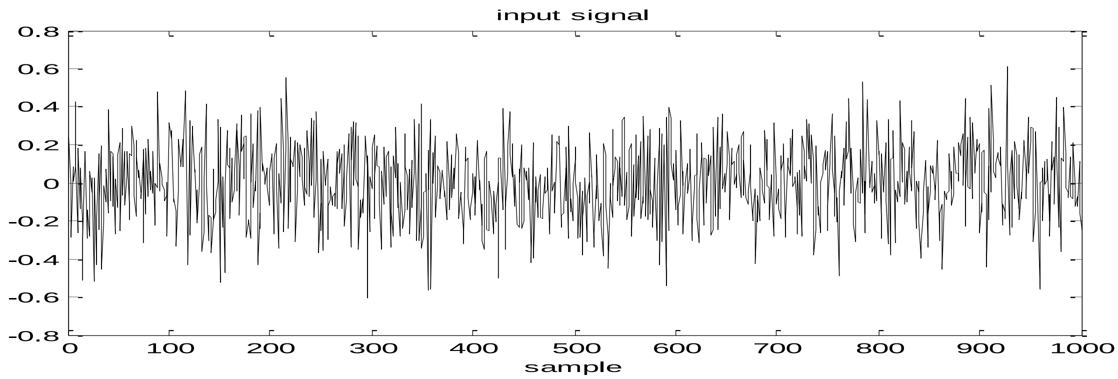

Figure 2. Input sequences.

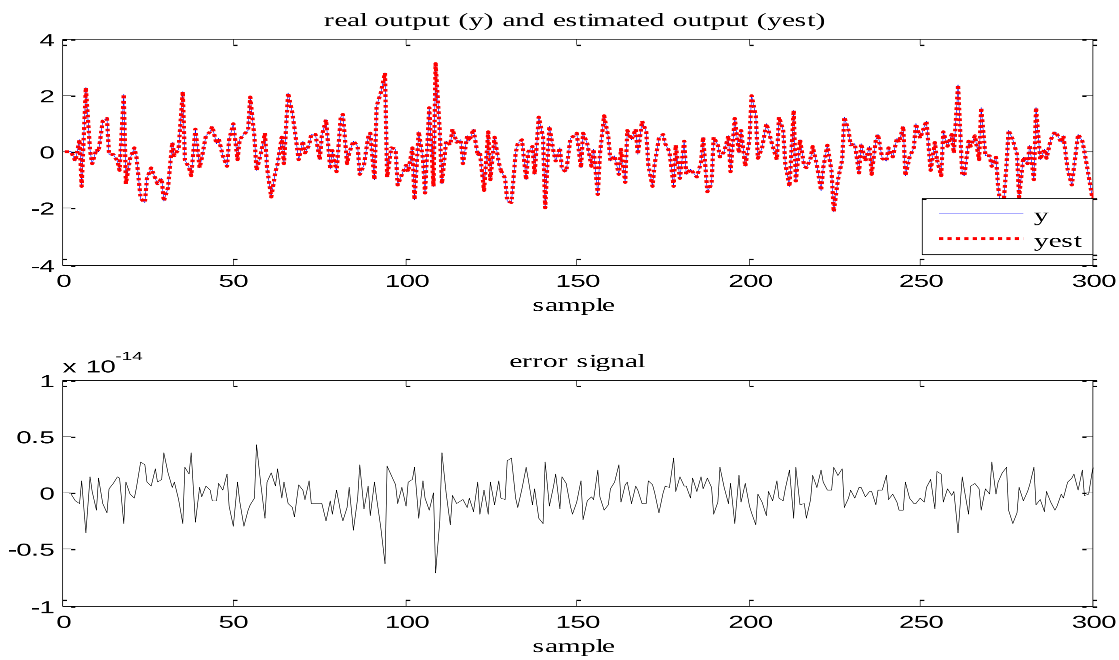

Figure 3. Identification result for the proposed algorithm.

the identification of the algorithms we have introduced previously.

i. There exists the system by a random binary signal given in Figure 2.
ii. For another input**,** the simulation results given by the proposed algorithm is given in Figures 3 and 4.

Examples 2: this system is described by the following equation:

$$y(k+1)=\frac{y(k)\ (y(k-1)+2)\ (y(k)+2.5)}{8.5+y^2(k)+\ y^2(k-1)}+u(k)+e(k)$$

The simulation results concerning the identification of the algorithms we have introduced previously have been presented as follows;

i. There exit the system by a random binary signal shown in Figure 5. proposed algorithm is given in Figures 6 and 7.

VALIDATION MODEL

Therefore, to ensure that the model obtained from the estimation it is compatible with other forms of inputs to correctly represent the system operating to identify it. It we present, in this paragraph, statistical tests to validate

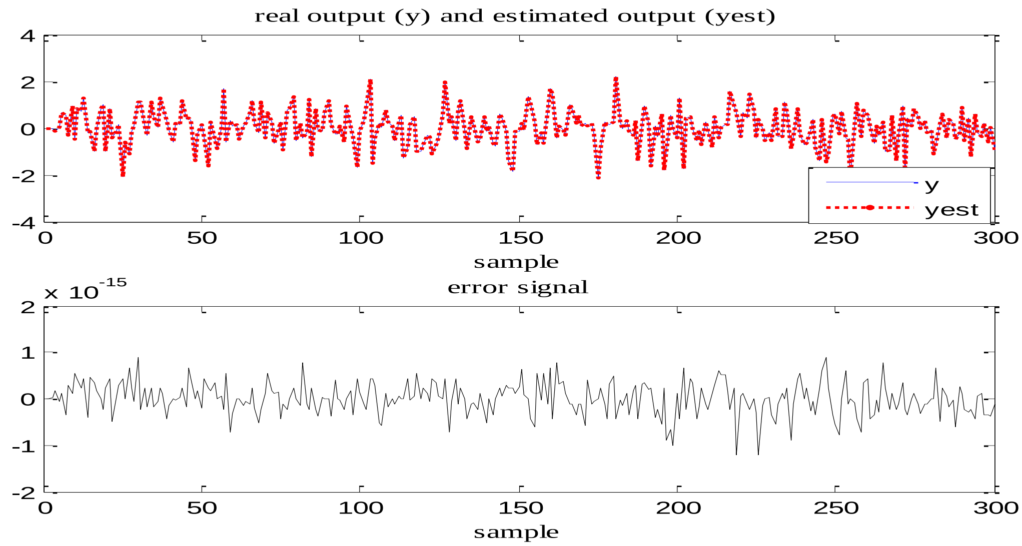

Figure 4. Identification result for the proposed algorithm-PSO.

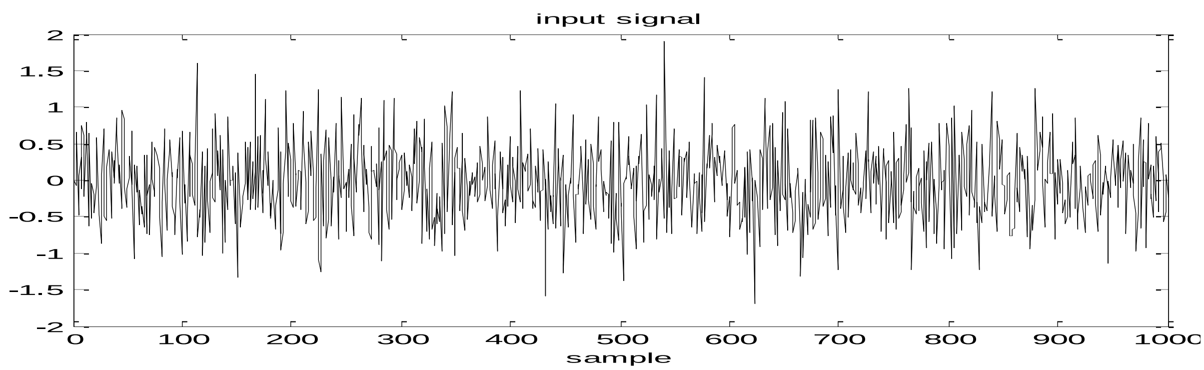

Figure 5. Input sequences.

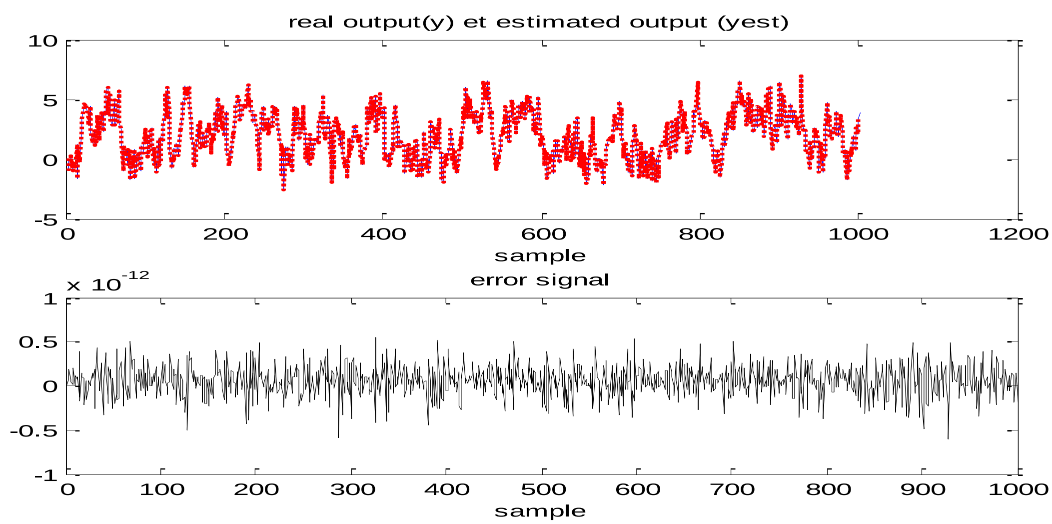

Figure 6. Identification result for the proposed algorithm.

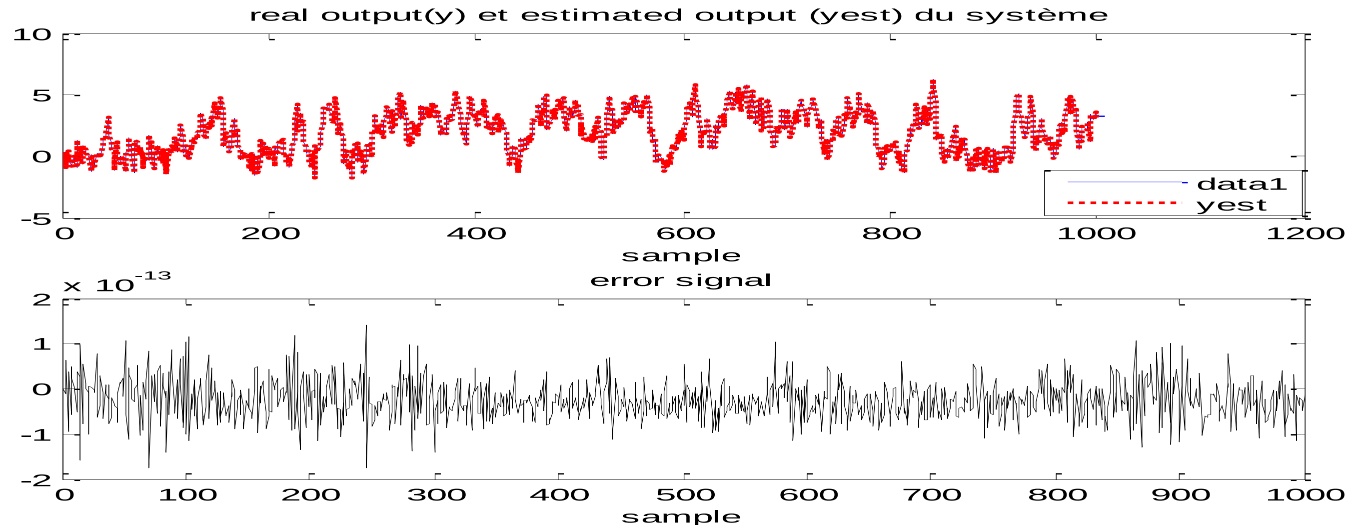

Figure 7. Identification result for the proposed algorithm-PSO.

a prediction model based on the residues autocorrelation function and on the cross-correlation between residues and other inputs in the system. Moreover one will present other validation tests named the RMSE test and the VAF test.

A) RMSE (Root Mean Square Error)

This test calculates the mean squared error between the measured output and model output.

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y(k)-y(k)\right)^2} \tag{23}$$

When the model output and actual output are very near, the test tends to zero.

B) VAF (Variance Accounting For)

Introduced by Babuska (1998), this criterion makes it possible to evaluate expressed as a percentage, quality of a model by measuring the standardized variation of the variance between two signals. Its optimal value is 100% when the two signals are equal, more they are different, plus its value becomes weak. Criteria VAF is given by the expression (24)

$$VAF = 100\%\left[1-\frac{\mathrm{var}\left(y(k)-y(k)\right)}{\mathrm{var}(y(k))}\right] \tag{24}$$

Example 1

C) Residue auto-correlation function:

$$\hat{r}_{\varepsilon\varepsilon}(\tau)=\frac{\sum_{k=1}^{N-\tau}\left(\varepsilon(k,\hat{\theta})-\bar{\varepsilon}\right)\left(\varepsilon(k-\tau,\hat{\theta})-\bar{\varepsilon}\right)}{\sum_{k=1}^{N}\left(\varepsilon(k,\hat{\theta})-\bar{\varepsilon}\right)^2} \tag{25}$$

(D) Cross-correlation function between residues and previous input

$$\hat{r}_{u\varepsilon}(\tau)=\frac{\sum_{k=1}^{N-\tau}\left(u(k)-\bar{u}\right)\left(\varepsilon(k-\tau,\hat{\theta})-\bar{\varepsilon}\right)}{\sqrt{\sum_{k=1}^{N}\left(u(k)-\bar{u}\right)^2}\sqrt{\sum_{k=1}^{N}\left(\varepsilon(k,\hat{\theta})-\bar{\varepsilon}\right)^2}} \tag{26}$$

ε is the prediction error and u is the system input.

The validity of model requires the following results:

$$\hat{r}_{\varepsilon\varepsilon}(\tau)=\begin{cases}1 & si \quad \tau=0\\ 0 & si \quad \tau\neq 0\end{cases} \quad \text{and} \quad \hat{r}_{u\varepsilon}(\tau)=0,\ \forall\tau$$

In general, the correlation functions $\hat{r}$ are zero when τ is the interval $[-20,20]$ with a confidence interval of 95%,

i.e: $\frac{-1.96}{\sqrt{N}}<\hat{r}<\frac{1.96}{\sqrt{N}}$

The simulation results show that the proposed algorithm-PSO can effectively solve the problem of the other algorithm (FCM, GK and PCM). The validation tests used (Table 1) have shown good performance of these algorithms. However, their RMSE and VAF show a better behavior of the proposed algorithm- PSO compared to the FCM-PSO algorithm, the GK-PSO algorithm and the PCM-PSO algorithm (Figures 8 and 9).

Table 1. Validation results.

	FCM	GK	PCM	Proposed algorithm
RMSE	$2.3527\ 10^{-5}$	$2.3713\ 10^{-5}$	$2.3341\ 10^{-5}$	1.24×10^{-10}
VAF (%)	99.9987	99.9986	99.9992	99.78
Computation time	16.21	18.54	21.68	32.05

	FCM-PSO	GK-PSO	PCM-PSO	Proposed algorithm-PSO
RMSE (10-5)	$2.1283\ 10^{-5}$	$1.4087\ 10^{-5}$	1.2201	3.4772^{-16}
VAF (%)	99.9987	99.9995	99.9997	99.9999
Computation time	7.86	10.21	17.56	24,17

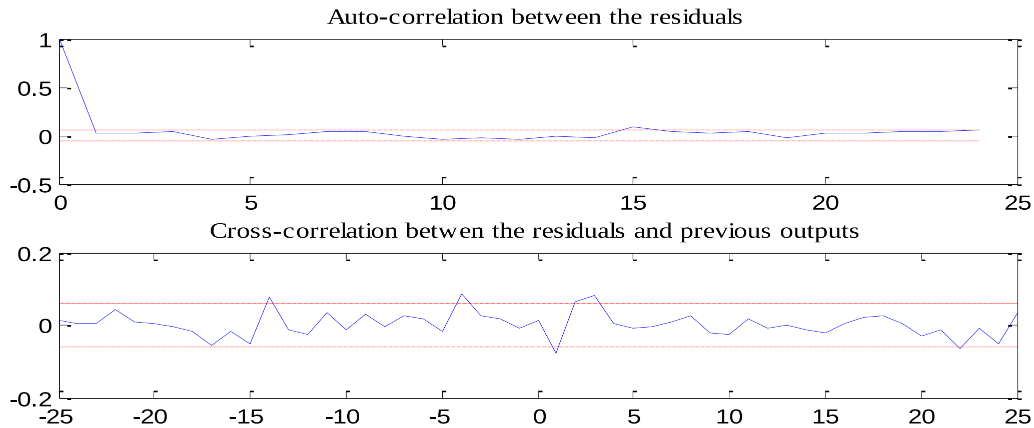

Figure 8. Validation result for the proposed algorithm.

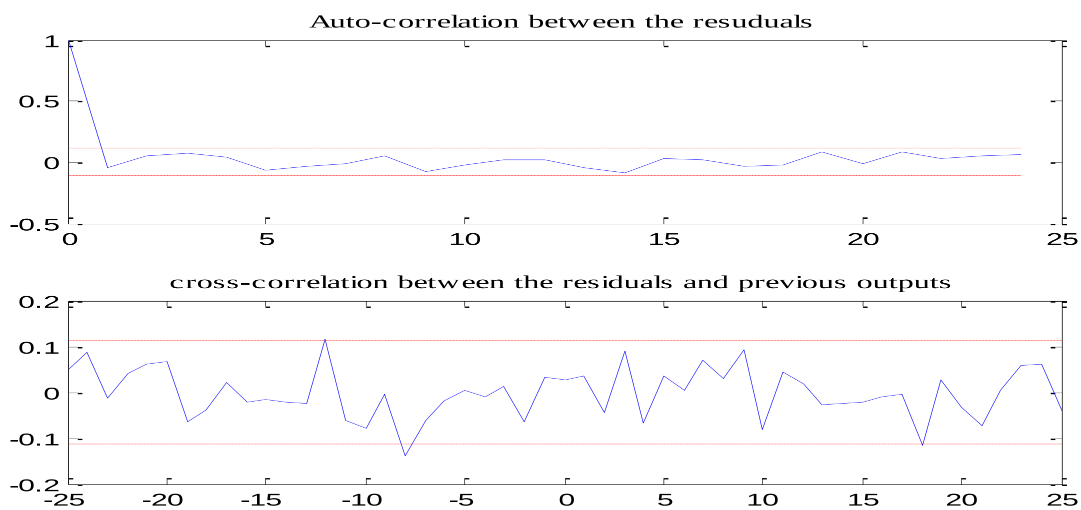

Figure 9. Validation result for the proposed algorithm-PSO.

Example 2

The simulation results (Table 2) show that the proposed algorithm-PSO can effectively solve the problem of the other algorithm (FCM, GK and PCM). The validation tests used have shown good performance of these algorithms

Table 2. Validation results.

	FCM	GK	PCM	Proposed algorithm
RMSE	$2.9080\ 10^{-5}$	5.837010^{-5}	2.119110^{-5}	1.831410^{-13}
VAF (%)	100	100	100	100
Computation time	9.76	5,99	15.48	25.37
	FCM-PSO	**GK-PSO**	**PCM-PSO**	**Proposed algorithm-PSO**
RMSE (10-5)	$4.0209\ 10^{-7}$	$5.4891\ 10^{-7}$	$1.1357\ 10^{-6}$	5.422310^{-14}
VAF (%)	100	100	100	100
Computation time	1.71	2.05	7.22	19.45

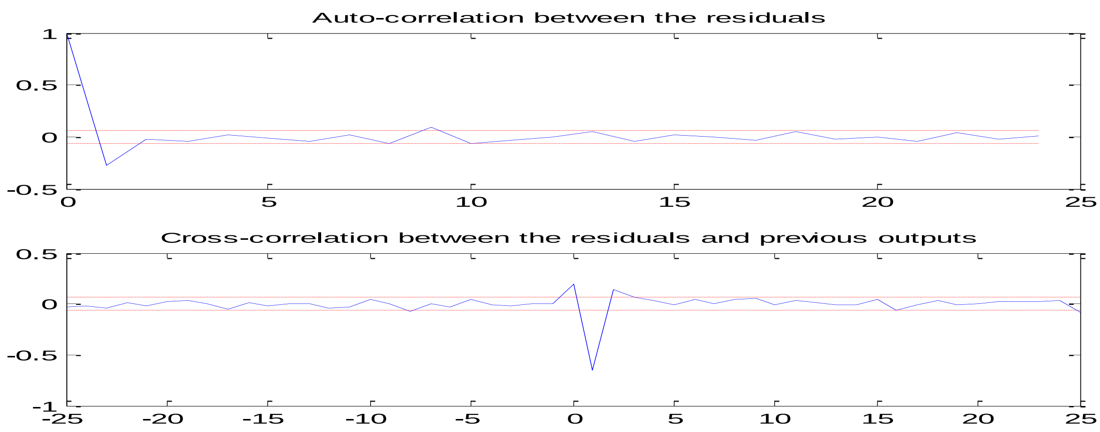

Figure 10. Validation result for the proposed algorithm.

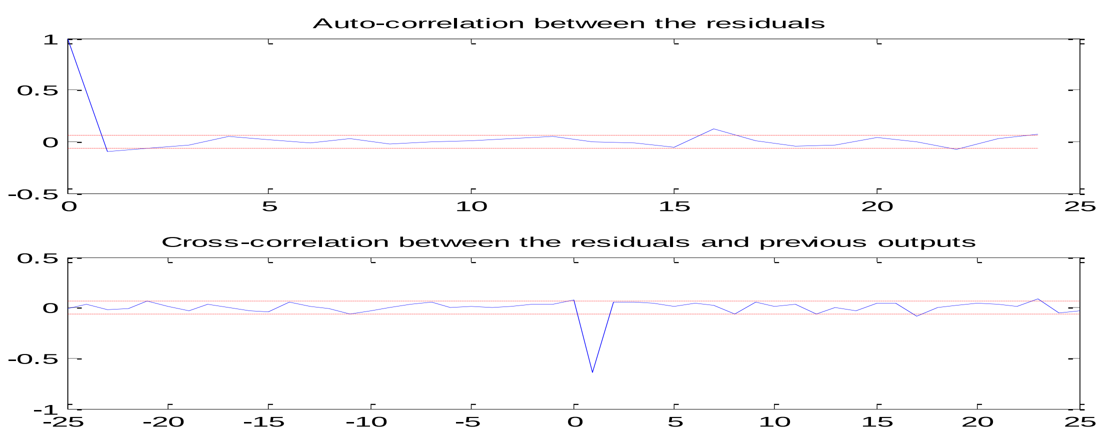

Figure 11. Validation result for the proposed algorithm-PSO.

(Figures 10 and 11).

APPLICATION TO AN ELECTRO-HYDRAULIC SYSTEM

The effectiveness of the identification algorithm we proposed in this paper is tested on an electro-hydraulic system described by the schematic diagram in Figure 12. Filling the tank 1 is achieved through a centrifugal pump unidirectional. The latter is driven by a DC motor controlled by a variable speed operating in a single quadrant. The tank 1 is located at an elevation difference ″ha″ compared to the container 2. The direction of flow of

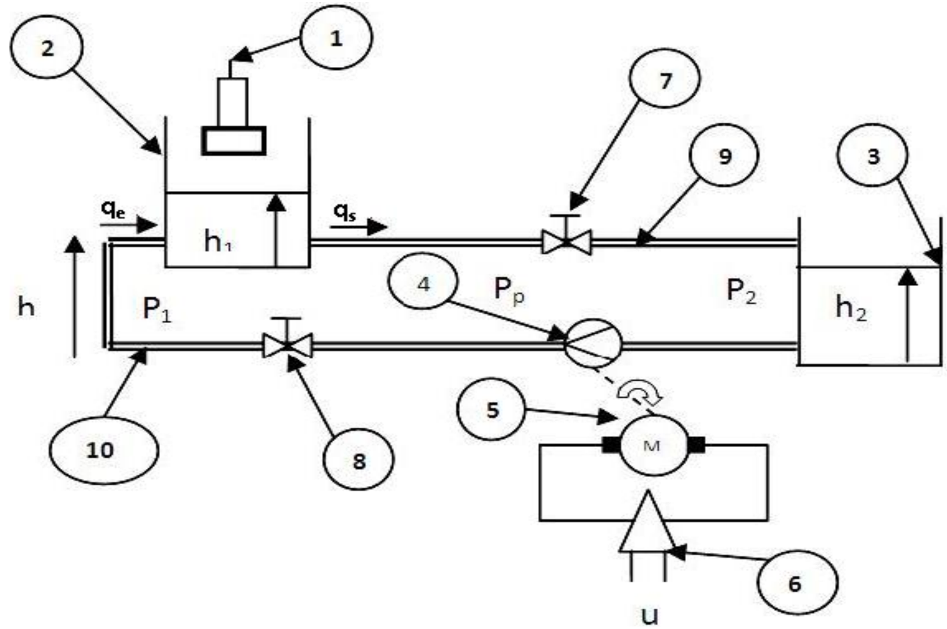

Figure 12. bloc diagram. **1**: Ultrasonic level sensor; **2**: Tank 1; **3**: Tank 2; **4**: Centrifugal pump; **5**: DC motor; **6**: Variable speed; **7**: Manual valve v1; **8**: manual valve v2; **9**: Pipe 1; **10**: Pipe 2; **qe**: felling flow; **qs**: outgoing flow of tank 1; **P1**: pressure at the button of tank; **P2**: pressure at the button of tank 2; **Pp**: exit pressure of centrifugal pump; **h1**: water level in tank 1; **h2**: water level in tank 2; **ha**: difference in altitude between the sites of the two tanks; **u**: supply voltage of the engine.

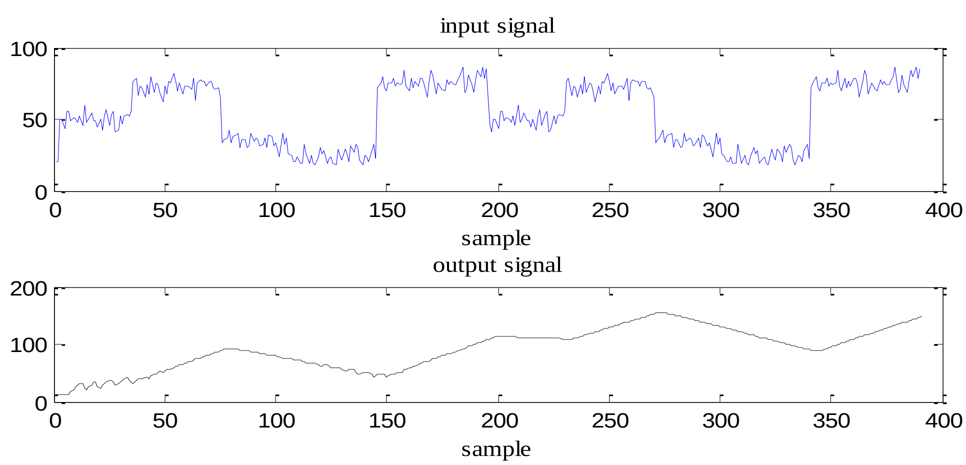

Figure 13. Sequence of input-output.

liquid depends mainly on the pressure ″Pp″ in the output of the pump pressure at the bottom of two tanks ″ P1″ and ″P2″ and the pressure due to the difference in the elevation between the two reservoirs. The manual valve ″v2″ is always kept open. In contrast, the valve ″v1″ is used as body perturbation dump tank 1. The fluid level in the reservoir 1 is measured using an analog ultrasonic sensor.

Identification of system parameters

To identify the parameters of this system, we applied a proposed clustering algorithm. The set of observations we have taken is illustrated in Figure 13.

For another sequence of input-output**,** the simulation results given by the proposed algorithm is given in Figures 14 and 15.

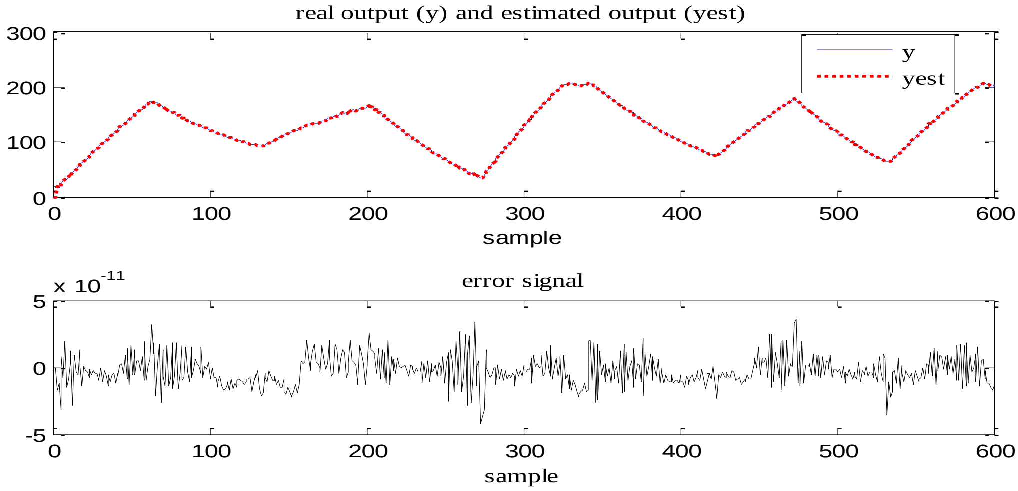

Figure 14. Identification result for the proposed algorithm.

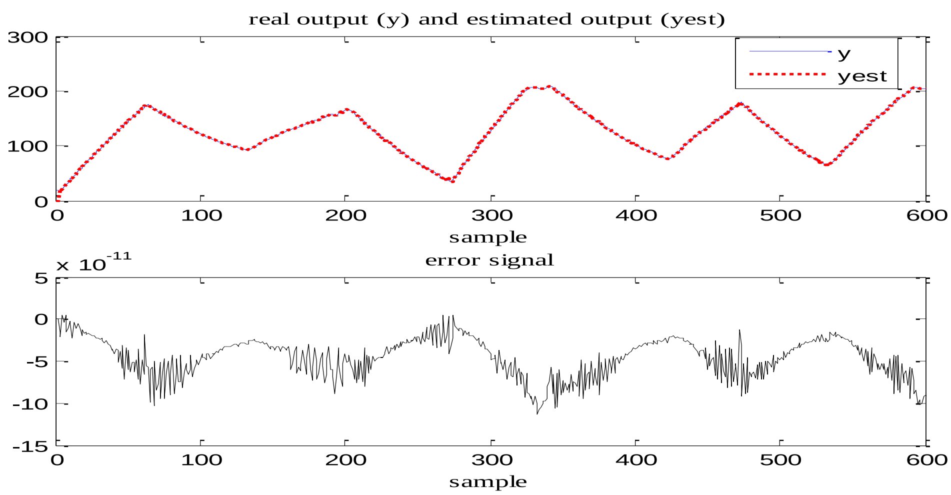

Figure 15. Identification result for the proposed algorithm-PSO.

Table 3. Validation results.

	FCM	GK	PCM	Proposed algorithm
RMSE (10^{-5})	1.4766 10^{-5}	1.2074 10^{-5}	1.2096 10^{-5}	1.4780 10^{-8}
VAF(%)	99.9956	99.9983	99.9981	98.9999
Computation time	8.61	15.51	14.64	28.79
	FCM-PSO	**GK-PSO**	**PCM-PSO**	**Proposed algorithm-PSO**
RMSE (10-5)	1.2388 10^{-5}	1.1883 10^{-5}	1.1976 10^{-5}	5.0911 10^{-11}
VAF (%)	99.9979	99.9987	99.9974	99.98
Computation time	4.77	7.22	6.88	21.18

Validation results

The validations results as well as their RMSE and VAM tests (Table 3) show well the effectiveness of the proposed algorithm (modified algorithm and modified algorithm-PSO) compared to the other algorithms,

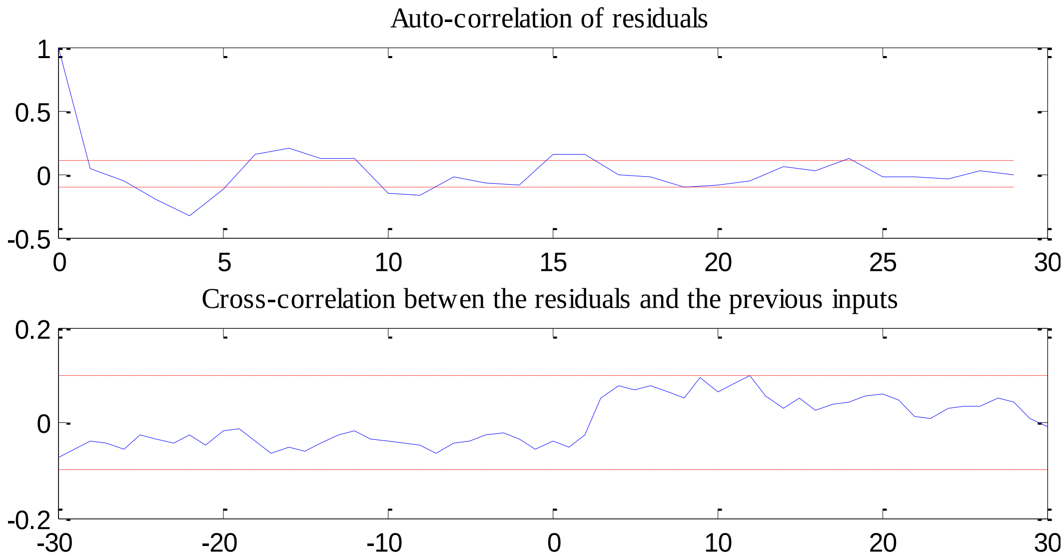

Figure 16. Validation result for the proposed algorithm.

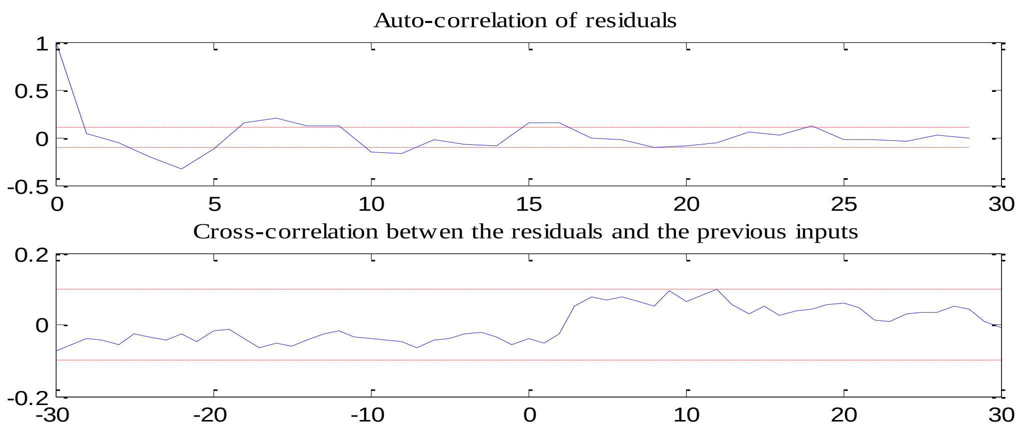

Figure 17. Validation result for the proposed algorithm-PSO.

however the modified algorithm combined with the PSO algorithm has the best result (Figures 16 and 17).

Conclusion

In this paper, another approach of the identification of nonlinear stochastic systems is used. Unlike to the other clustering algorithms which has been proposed, that only allows the identification of premise parameters, the proposed algorithm can estimate simultaneously the premise and consequence parameters by using an iterative optimization method. It is starting from the minimization of four optimization criterion. In fact this algorithm is an extension of the algorithm proposed by J. Q. Chen. In this paper we introduced some modifications in the last two criterion and we replaced the Euclidean distance by another non Euclidean distance. The proposed algorithm overcome the problems of sensitivity to noise and aberrant points, however, it cannot solve the problems of convergences and the time computing.

The particle swarm optimization method combined with the proposed algorithm can solve these problems. The experimental results on a nonlinear system as well as on a level control system showed that the proposed algorithm and the proposed algorithm combined with the PSO presented successful results.

But it is interesting to note that the proposed algorithm combined with the PSO algorithm present the best convergence results and computing time compared the proposed algorithm only.

REFERENCES

Adonyi J, Babuska R, Szeifert F (2002). Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models. IEEE Trans. Syst. Man. Cybern. Part B., 32(5): 612-621.

Ahmed T, Abdelkader C, Moncef G (2008). Algorithmes de clustering pour l'identification d'un modèle flou. JTEA, pp. 1085-1094.

Ahmed T, Lassad H, Abdelkader C (2011). New Extented Possibilistic C-Means algorithm for identification of an Electreo-hydraulic system. 12th. Inter. Conf. STA. ACS. 1695: 1-10.

Araujo E, Coelho LS (2008). Particle swarm approches using lozi map chaotic sequences to fuzzy modelling of an experimental thermal-vauum system. Appl. Soft Comput., 8: 1354-1364.

Babuska R (1998). Fuzzy Modeling for Control. Kluwer. Acad. Publ. Mass. USA.

Babuska R, Verbruggen H (2003). Neuro-fuzzy methods for nonlinear system identification. Annual. Rev. in cont., 27: 73-85.

Biswal B, Dash PK, Panigrahi BK (2009). Power Quality Disturbance Classification Using Fuzzy C-Means Algorithm and Adaptive Particle Swarm Optimization. IEEE Trans. Ind Elect., 56(1): 212-220.

Chen JQ, Xi YG, Zhang ZJ (1998). A clustering algorithm for fuzzy model identification. Fuzzy Sets Syst., 319-329.

Coelho LS, Herrera BM (2007). Fuzzy Identification Based on a Chaotic Particle Swarm Optimization Approach Applied to a Nonlinear Yo-yo Motion System. IEEE Trans. Ind. Elect., 54(6): 3234-3245.

Dunn JC (1974). A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybern., 3(3): 32-57.

Eberhart RC, Kennedy J (1995). A new optimizer using particle swarm theory. in Proc. 6th Int. Symp. Micro Mach. and Human Sci. Nag. Jap., pp. 39-43.

Favier G (1982). Filtrage, modélisation et identifications des systems linéaires stochastiques à temps discret. Edit. CNRS. Paris.

Gath I, Geva AB (1989). Unsupervised optimal fuzzy clustering. IEEE Trans. Patt. Anal. Mach. Intell., 7: 773-781.

Goldberg DE (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading. MA. Addison. Wesley.

Grisales VH (2007). Modélisation et commande floue de type Takagi-Sugeno appliquées à un Bioprocédé de traitement des eaux usées. Thèse de doctorat de l'Université Paul Sabaties- Toulouse III et l'Université de los Andes. Colombie.

Gustafson DE, Kessel WC (1979). Fuzzy Clustering with a fuzzy covariance matrix. In Proc. IEEE CDC. San Diego. CA. USA, pp.761-766.

Jang W, Kang H, Lee B, Kim K, Shin D, Kim S (2007). Optimized Fuzzy Clustering By Predator Prey Particle Swarm Optimization. IEEE Congresson Evol. Comput., pp. 3232-3238.

Kennedy J, Eberhart RC (1995). Particle swarm optimization. in Proc. IEEE Int. Conf. Neural Netw. IV. Perth. Aust, pp. 1942-1948.

Krishnapuram R, Keller J (1996). The Possibilistic c-Means Algorithm: Insights and Recommendations. IEEE Trans. Fuzzy Syst., 4(3): 385-393.

Pan H, Wang L, Liu B (2006). Particle swarm optimization for function optimization in noisy environment. Appl. Math. Comput., 181: 908-919.

Pingli L, Yang Y, Wenbo M (2006). Random sampling fuzzy c-means clustering and recursive least square based fuzzy identification. Proceedings of the 2006 American control conference.

Sastry PS, Santharam G, Unnikrishnan KP (1994). Memory neuron networks for identification and control of dynamical systems. IEEE Trans. Neural networks, NN-5: 306-319.

Takagi T, Sugeno M (1985). Fuzzy identification of systems and its application to modelling and control. IEEE Trans. Syst. Man Gyber., 15: 116-132.

Wu KL, Yang MS (2002). Alternative c-means clustering algorithms, Patt. Recog., 35: 2267-2278.

Xu YF, Zhang SL (2009). Fuzzy Particle Swarm Clustering of Infrared Images. Sec. Int. Conf. on Information and Computing Science.

Zahid N, Abouelala O, Limouri M, Essaid A (2003). Fuzzy clustering based on K-nearestn neighbors rule. Fuzzy Sets Syst., pp. 73-85.