Fuzzy Systems

Fuzzy Systems

Edited by
Ahmad Taher Azar

Intech

Published by Intech

Intech
Olajnica 19/2, 32000 Vukovar, Croatia

Free online edition of this book you can find under www.sciyo.com
Additional copies can be obtained from:
publication@sciyo.com

First published February 2010
Printed in India

Technical Editor: Teodora Smiljanic
Cover designed by Dino Smrekar

Fuzzy Systems, Edited by Ahmad Taher Azar
p. cm.
ISBN 978-953-7619-92-3

Preface

Since the idea of the fuzzy set was proposed in 1965, many developments have occurred in this area. Applications have been made in such diverse areas as medicine, engineering, management, behavioral science, just to mention some. The application of the fuzzy sets involves different technologies, such as fuzzy clustering on image processing, classification, identification and fault detection, fuzzy controllers to map expert knowledge into control systems, fuzzy modeling combining expert knowledge, fuzzy optimization to solve design problems. Fuzzy systems are used in the area of artificial intelligence as a way to represent knowledge. This representation belongs to the paradigm of behavioral representation in opposition to the structural representation (neural networks). The foundation of this paradigm is that intelligent behavior can be obtained by the use of structures that not necessarily resemble the human brain. A very interesting characteristic of the fuzzy systems is their capability to handle in the same framework numeric and linguistic information. This characteristic made these systems very useful to handle expert control tasks.

While several books are available today that address the mathematical and philosophical foundations of fuzzy logic, none, unfortunately, provides the practicing knowledge engineer, system analyst, and project manager with specific, practical information about fuzzy system modeling. Those few books that include applications and case studies concentrate almost exclusively on engineering problems: pendulum balancing, truck backeruppers, cement kilns, antilock braking systems, image pattern recognition, and digital signal processing. Yet the application of fuzzy logic to engineering problems represents only a fraction of its real potential. As a method of encoding and using human knowledge in a form that is very close to the way experts think about difficult, complex problems, fuzzy systems provide the facilities necessary to break through the computational bottlenecks associated with traditional decision support and expert systems. Additionally, fuzzy systems provide a rich and robust method of building systems that include multiple conflicting, cooperating, and collaborating experts (a capability that generally eludes not only symbolic expert system users but analysts that have turned to such related technologies as neural networks and genetic algorithms). Yet the application of fuzzy logic in the areas of decision support, medical systems, database analysis and mining has been largely ignored by both the commercial vendors of decision support products and the knowledge engineers that use them.

This book is intended to present fuzzy logic systems and useful applications with a self-contained, simple, readable approach. It is intended for the intelligent reader with an alert mind. The approach, the organization, and the presentation of this book are also hoped to enhance the accessibility to existing knowledge beyond its contents. The book is divided into twelve chapters. In Chapter 1 Gaussian membership functions (MFs) are proposed as an alternative to the traditional triangular MFs in order to improve the reliability and robustness of the system. Gaussian MFs provide smooth transition between levels and provides a way to fire the maximum number of rules in the rule base and hence a more accurate representation of the input-output relationship is achieved. Chapter 2 describes robust H_∞ control problems for uncertain Takagi-Sugeno (T-S) fuzzy systems with immeasurable premise variables. A continuous-time Takagi-Sugeno fuzzy system is first considered. The same control problems for discrete-time counterpart are also considered. Chapter 3 deals with the control of T-S fuzzy systems using fuzzy weighting-dependent lyapunov function. In Chapter 4, the digital fuzzy control system considering a time delay is developed and its stability analysis and design method are proposed. The discrete Takagi-Sugeno(TS) fuzzy model and parallel distributed compensation(PDC) conception for the controller are used. The proposed control system can be designed using the conventional methods for stabilizing the discrete time fuzzy systems and the feedback gains of the controller can be obtained using the concept of the linear matrix inequality (LMI) feasibility problem.

Chapter 5 presents an overview of adaptive neuro-fuzzy systems developed by exploiting the similarities between fuzzy systems and certain forms of neural networks, which fall in the class of generalized local methods. The chapter starts by making a classification of the different types of neuro-fuzzy systems and then explains the modeling methodology of neuro-fuzzy systems. Finally, the chapter is completed by a practical case-study. Chapter 6 describes a hybrid fuzzy system to develop a new technique, an integrated classifier, for real-time condition monitoring in, especially, gear transmission systems. In this novel classifier, the monitoring reliability is enhanced by integrating the information of the object's future states forecast by a multiple-step predictor; furthermore, the diagnostic scheme is adaptively trained by a novel recursive hybrid algorithm to improve its convergence and adaptive capability. Chapter 7 presents the mathematical theory of fuzzy filtering and its applications in life science. Chapter 8 is devoted to develop applications to enable information extraction under uncertainty, particularly on the conception and design of autonomous systems for natural language processing applications specifically on question and answering systems and textual entailment mechanism.

Chapter 9 deals with the algorithms of the body signature identification. The developed systems can be used to detect body position on the bed as well as the type of body movement. Using the body movement, the time period between two successive movement and the sensor amplitude, one can identify the sleep type (normal, agitate, abnormal, convulsive, etc). The system can be adapted to the person and does not depend on their weight, size or position. Chapter 10 is devoted to the probelm of students' evaluation method. This chapter proposes the use of the fuzzy set technique that will be applied in the evaluation process of the industrial automation systems learning area, aiming to lessen the evaluation complexity and ambiguity in this area. It is also important to emphasize that this fuzzy learning evaluation methodology may be used when training industrial plant operators and engineers who have already been working in the area but must be trained in new, emerging technologies. Chapter 11 studies the combination of particle swarm optimization (PSO) and ant colony optimization (ACO) for the design of fuzzy systems. One problem of PSO in FS design is that its performance is affected by initial particle positions, which are usually randomly generated in a continuous search space. A poor initialization usually results in poor performance. Searching in the discrete-space domain by ACO helps to find good solutions. However, the search constraint in a discrete-space domain restricts learning accuracy. The motivation on the combination of ACO and PSO is to compensate the aforementioned weakness of each method in FS design problems. Finally, Chapter 12 presents a directed formation control problem of heterogeneous multi-agent systems. Fuzzy logical controller for multi-agent systems with leader-following is presented, which can not only accomplish the desired triangle formation but also ensure that the followers' speeds converge to the leader's velocity without collision during the motion. The proposed Fuzzy logical controller is interesting for the design of optimization algorithms that can ensure the triangle formation that multi-agent systems are required maintaining a nominated distance.

The book is written at a level suitable for use in a graduate course on applications of fuzzy systems in decision support, nonlinear modeling and control. The book discusses novel ideas and provides a new insight into the studied topics. For this reason, the book is a valuable source for researchers in the areas of artificial intelligence, data mining, modeling and control. The realistic examples also provide a good opportunity to people in industry to evaluate these new technologies, which have been applied with success.

This text is addressed to engineering lecturers, researchers extending the frontiers of knowledge, professional engineers and designers and also students. A hallmark of fuzzy logic methods is that the cultural gap between researchers and practitioners is not apparent, the linguistic formulation of problems and conclusions is equally coherent to both.

I would like to acknowledge the invaluable help given by Aleksandar Lazinica in the final stages of compiling the text. Any flaws that remain are mine. Comments on any aspects of the text would be welcome.

Editor

Ahmad Taher Azar, PhD

Electrical Communication & Electronics Systems Engineering department,
Modern Science and Arts University (MSA),
6th of October City,
Egypt

Contents

1

Fuzzy Systems in Education: A More Reliable System for Student Evaluation

Ibrahim A. Hameed and Claus G. Sorensen
Aarhus University, Research Centre Foulum
Blichers Allé 20, DK-8830, Tjele,
Denmark

1. Introduction

Student evaluation is the process of determining the performance levels of individual students in relation to educational learning objectives. A high quality evaluation system certifies, supports, and improves individual achievement and ensures that all students receive a fair evaluation in order not to constrain students' present and future prospects. Thus, the system should regularly be reviewed and improved to ensure that it is suitable, fair, impartial and beneficial to all students. It is also desirable that the system is transparent and automation measures should be embedded in the evaluation. Fuzzy reasoning has proven beneficial to infer scores of students (e.g. Saleh & Kim, 2009). However, in order to improve the reliability and robustness of the system, Gaussian membership functions (MFs) are proposed as an alternative to the traditional triangular MFs.

Since its introduction in 1965 by Lotfi Zadeh (1965), the fuzzy set theory has been widely used in solving problems in various fields, and recently in educational evaluation. Biswas (1995) presented two methods for evaluating students' answer scripts using fuzzy sets and a matching function; a fuzzy evaluation method and a generalized fuzzy evaluation method. Chen and Lee (1999) presented two methods for applying fuzzy sets to overcome the problem of rewarding two different fuzzy marks the same total score which could result from Biswas' method (1995). Echauz and Vachtsevanos (1995) proposed a fuzzy logic system for translating traditional scores into letter-grades. Law (1996) built a fuzzy structure model for an educational grading system with its algorithm aimed at aggregating different test scores in order to produce a single score for an individual student. He also proposed a method to build the MFs of several linguistic values with different weights. Wilson, Karr and Freeman (1998) presented an automatic grading system based on fuzzy rules and genetic algorithms. Ma and Zhou (2000) proposed a fuzzy set approach to assess the outcomes of Student-centered learning using the evaluation of their peers and lecturer. Wang and Chen (2008) presented a method for evaluating students' answer scripts using fuzzy numbers associated with degrees of confidence of the evaluator. From the previous studies, it can be found that fuzzy numbers, fuzzy sets, fuzzy rules, and fuzzy logic systems are and have been used for various educational grading systems.

Evaluation strategies based on fuzzy sets require a careful consideration of the factors included in the evaluation. Weon and Kim (2001) pointed out that the system for students' achievement evaluation should consider three important factors of the questions which the students answer: the difficulty, the importance, and the complexity. Singleton functions were used to describe the factors of each question reflecting the effect of the three factors individually, but not collectively. Bai and Chen (2008b) stressed that the difficulty factor is a very subjective parameter and may cause an argument about fairness in the evaluation.

The automatic construction of the grade MFs of fuzzy rules for evaluating student's learning achievement has been attempted (Bai & Chen, 2008a). Also, Bai and Chen (2008b) proposed a method for applying fuzzy MFs and fuzzy rules for the same purpose. To solve the subjectivity of the difficulty factor embedded in the method of Weon and Kim (2001), Bai and Chen (2008b) acquired the difficulty parameter as a function of accuracy of the student's answer script and time used for each question. However, their method still has the subjectivity problem, since the resulting scores and rankings are heavily dependent on the values of several weights which are assessed by the subjective knowledge of domain experts.

Saleh and Kim (2009) proposed a three node fuzzy logic approach based on Mamdani's fuzzy inference engine and the centre-of-gravity (COG) defuzzification technique as an alternative to Bai and Chen's method (2008b). The transparency and objective nature of the fuzzy system makes their method easy to understand and enables teachers to explain the results of the evaluation to sceptic students. The method involved conventional triangular MFs of fixed parameters which could result in different results when changed. In this chapter, the Gaussian MFs are proposed as an alternative and a sensitivity study is conducted to get the appropriate values of their parameters for a more robust evaluation system.

The chapter will be organized as follows: In Section 2, a review of the three nodes fuzzy evaluation method based on triangular MFs is introduced. In Section 3, Gaussian MFs are proposed for a more robust evaluation system. A comparison of the two methods is presented in Section 4. Conclusions are drawn in Section 5.

2. A review of the three nodes fuzzy evaluation system

The method proposed by Bai and Chen (2008b) has several empirical weights which are determined subjectively by the domain expert. Quite different ranks can be obtained depending on these weight values. By using this method, the examiners could not easily verify how new ranks are acquired and could not persuade sceptical students. Naturally, there is no method to determine the optimum values of these weights. Also, the weighted arithmetic mean formula used to compute the outputs do not satisfy the concept of fuzzy set. To reduce the degree of subjectivity in this method and provide a method based on the theory of fuzzy set, Saleh and Kim (2009) proposed a system applying the most commonly used Mamdani's fuzzy inference mechanism (Mamdani, 1974) and center of gravity (COG) defuzzification technique. In this way, the system is represented as a block diagram of fuzzy logic nodes as shown in Fig. 1. The model of Bai and Chen (2008b) can be considered as a simple specific case of the block diagram by replacing each node with a weighted arithmetic mean formula. The system consists of three nodes; the difficulty node, the cost node and the adjustment node. Each node of the system behaves like a fuzzy logic controller (FLC in Fig. 1) with two scalable inputs and one output, as in Fig. 2. Each FLC maps a two-to-one fuzzy relation by inference through a given rule base.

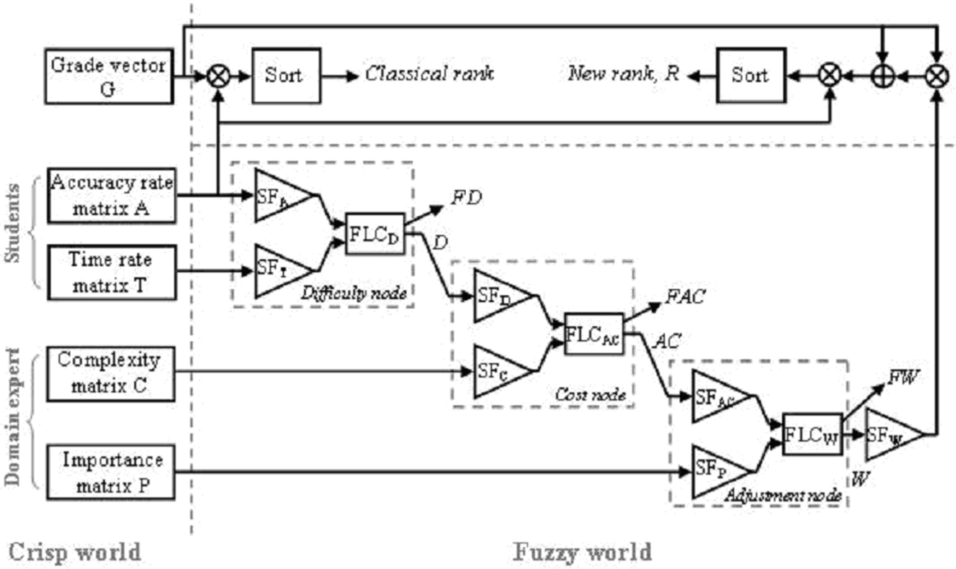


Fig. 1. Block diagram of the three nodes fuzzy evaluation system.

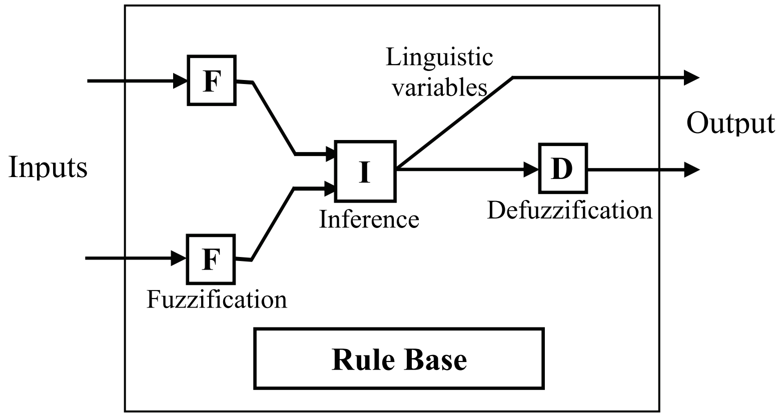


Fig. 2. Node representation as a fuzzy logic controller.

The inputs to the system in the left hand side of Fig. 1 are given by examination results and domain experts. Assume that there are n students to answer m questions. From examination results, we get an accuracy rate matrix, A, of dimension m x n, which is the student's scores in each question divided by the maximum score assigned to this question

$$A = [a_{ij}], \quad m \times n,$$

where $a_{ij} \in [0, 1]$ denotes the accuracy rate of student j on question i. We also get a time rate matrix, T, of dimension m x n, which is the time used by a student to solve a question divided by the maximum time allowed to solve this question

$$T = [t_{ij}], \quad m \times n,$$

where $t_{ij} \in [0, 1]$ denotes the time rate of student j on question i. We are also given a grade vector, G, of dimension m x 1

$$G = [g_i], \quad m \times 1,$$

where $g_i \in [1, 100]$ denotes the assigned maximum score of question i satisfying

$$\sum_{i=1}^{m} g_i = 100$$

Based on the accuracy rate matrix, A, and the grade vector G, we obtain the total score vector of dimension n x 1,

$$S = A^T G = [s_j], \quad n \times 1, \tag{1}$$

where $s_j \in [0, 100]$ is the total score of student j. The "classical" ranks of students are then obtained by sorting the element values of S in descending order.

Example. Assume that 10 students laid to an exam of 5 questions and the accuracy rate matrix, the time rate matrix, and the grade vector are given as follows (Bai & Chen, 2008b; Saleh & Kim, 2009):

$$A = \begin{bmatrix} 0.59 & 0.35 & 1 & 0.66 & 0.11 & 0.08 & 0.84 & 0.23 & 0.04 & 0.24 \\ 0.01 & 0.27 & 0.14 & 0.04 & 0.88 & 0.16 & 0.04 & 0.22 & 0.81 & 0.53 \\ 0.77 & 0.69 & 0.97 & 0.71 & 0.17 & 0.86 & 0.87 & 0.42 & 0.91 & 0.74 \\ 0.73 & 0.72 & 0.18 & 0.16 & 0.5 & 0.02 & 0.32 & 0.92 & 0.9 & 0.25 \\ 0.93 & 0.49 & 0.08 & 0.81 & 0.65 & 0.93 & 0.39 & 0.51 & 0.97 & 0.61 \end{bmatrix},$$

$$T = \begin{bmatrix} 0.7 & 0.4 & 0.1 & 1 & 0.7 & 0.2 & 0.7 & 0.6 & 0.4 & 0.9 \\ 1 & 0 & 0.9 & 0.3 & 1 & 0.3 & 0.2 & 0.8 & 0 & 0.3 \\ 0 & 0.1 & 0 & 0.1 & 0.9 & 1 & 0.2 & 0.3 & 0.1 & 0.4 \\ 0.2 & 0.1 & 0 & 1 & 1 & 0.3 & 0.4 & 0.8 & 0.7 & 0.5 \\ 0 & 0.1 & 1 & 1 & 0.6 & 1 & 0.8 & 0.2 & 0.8 & 0.2 \end{bmatrix},$$

$$G^T = \begin{bmatrix} 10 & 15 & 20 & 25 & 30 \end{bmatrix}$$

Here, G^T denotes the transpose of G. Total score for each individual student is then obtained by formula (1) as

$$\begin{matrix} & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 & s_9 & s_{10} \\ S^T = [& 67.60 & 54.05 & 38.40 & 49.70 & 49.70 & 48.80 & 46.10 & 52.30 & 85.95 & 49.70 \;] \end{matrix}$$

and thus the "classical" ranks of students is become

$$S_9 > S_1 > S_2 > S_8 > S_4 = S_5 = S_{10} > S_6 > S_7 > S_3,$$

where $S_a > S_b$ means that the score of student *a* is higher than score of student *b* and $S_a = S_b$ means that their scores are equal.

□

From the domain expert, we get the importance matrix, *P*, of dimension *m* x *l* which describes the degree of importance of each question in the fuzzy domain.

$$P = [p_{ik}], \quad m \times l,$$

where $p_{ik} \in [0, 1]$ denotes the degree of importance of question *i* belonging to the importance level *k*. Here, five levels of importance (*l* = 5) are represented by five fuzzy sets; *k* = 1 representing the linguistic term "low", *k* = 2 representing "more or less low", *k* = 3 representing "medium", *k* = 4 representing "more or less high" and k = 5 representing "high". The MFs are shown in Fig. 3. Once crisp values are given as a vector for the importance of questions by a domain expert, the values of fuzzy importance matrix or p_{ik}'s are obtained by the fuzzification.

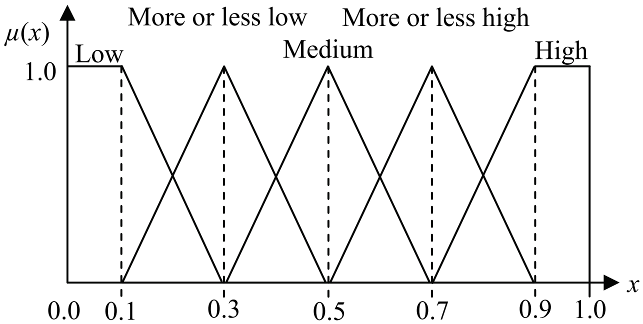


Fig. 3. Triangular membership functions of the five levels.

It is noted that the same five fuzzy sets, shown in Fig. 3, are applied to represent the accuracy, the time rate, the difficulty, the complexity, and the adjustment of questions in the fuzzy domain.

We are also given the complexity matrix of dimension *m* x *l*, which is an important factor indicating the ability of students to give correct answers of complex questions

$$Co = [co_{ik}], \quad m \times 1,$$

where $co_{ik} \in [0, 1]$ denotes the degree of complexity of question *i* belonging to the complexity level *k*.

Example. For the above example we get the following, in fuzzy domain, by the domain expert:

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0.33 & 0.67 & 0 & 0 \\ 0 & 0 & 0 & 0.15 & 0.85 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0.07 & 0.93 & 0 & 0 \end{bmatrix}$$

$$Co = \begin{bmatrix} 0 & 0.85 & 0.15 & 0 & 0 \\ 0 & 0 & 0.33 & 0.67 & 0 \\ 0 & 0 & 0 & 0.69 & 0.31 \\ 0.56 & 0.44 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0.3 & 0 \end{bmatrix}$$

□

As a primary step, the inputs from the examination results are averaged and fuzzified based on the defined levels (fuzzy sets) in Fig. 3. The average accuracy rate vector of dimension m x 1 is obtained as

$$\bar{A} = [\, a_{i\bullet} \,], \quad m \times 1,$$

where $a_{i\bullet}$ denotes the average accuracy rate of question i which is obtained by

$$a_{i\bullet} = \sum\nolimits_{j=1}^{n} a_{ij} \Big/ n, \tag{2}$$

and the average time rate vector of the same dimension m x 1 is obtained as

$$\bar{T} = [\, t_{i\bullet} \,], \quad m \times 1,$$

where $t_{i\bullet}$ denotes the average time rate of question i which is obtained by

$$t_{i\bullet} = \sum\nolimits_{j=1}^{n} t_{ij} \Big/ n. \tag{3}$$

Next, by fuzzification, we obtain the fuzzy accuracy rate matrix of dimension m x l,

$$FA = [fa_{ik}], \quad m \times l,$$

where $fa_{ik} \in [0, 1]$ denotes the membership value of the average accuracy rate of question i belonging to level k, and the fuzzy time rate matrix of dimension m x l,

$$FT = [ft_{ik}], \quad m \times l,$$

where $ft_{ik} \in [0, 1]$ denotes the membership value of the average time rate of question i belonging to level k, respectively.

Example. For the above by formula (2) and formula (3), respectively, we get

$$\bar{A}^T = \begin{bmatrix} 0.45 & 0.31 & 0.711 & 0.47 & 0.637 \end{bmatrix},$$

$$\bar{T}^T = \begin{bmatrix} 0.57 & 0.48 & 0.31 & 0.50 & 0.57 \end{bmatrix}.$$

Based on the fuzzy MFs in Fig. 3 we obtain the fuzzy accuracy rate matrix and the fuzzy time rate matrix as:

$$FA = \begin{bmatrix} 0 & 0.25 & 0.75 & 0 & 0 \\ 0 & 0.95 & 0.05 & 0 & 0 \\ 0 & 0 & 0 & 0.945 & 0.055 \\ 0 & 0.15 & 0.85 & 0 & 0 \\ 0 & 0 & 0.315 & 0.685 & 0 \end{bmatrix},$$

$$FT = \begin{bmatrix} 0 & 0 & 0.65 & 0.35 & 0 \\ 0 & 0.1 & 0.9 & 0 & 0 \\ 0 & 0.95 & 0.05 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0.65 & 0.35 & 0 \end{bmatrix}.$$

□

In the first node, both inputs are given by examination results, whereas in the later nodes, one input is the output of its previous node and the other is given by a domain expert. The output of each node can be in the form of a crisp value (defuzzified) or in the form of linguistic variables (MFs). Each node has two scale factors (SFs in Fig. 1). We can adjust the effects of inputs by varying the values of scale factors. Here, we let both scaling factors have the same value of unity assuming equal influence of each input on the output. Each fuzzy node proceeds in three steps:

Step 1 (Fuzzification). In this step, if given in crisp sets, inputs are converted into membership values in the fuzzy sets shown in Fig. 3. Triangular MF is the commonly used due to its simplicity and easy computation.

Step 2 (Inference). Here, inference is performed based on the given rule base, in the form of IF–THEN rules. We use Mamdani's max-min inference mechanism which is the most commonly used inference mechanism to produce fuzzy sets for defuzzification (1974). In Mamdani's max-min mechanism, implication is modelled by means of minimum operator, and the resulting output MFs are combined using maximum operator.

The inference mechanism can be written into the form:

$$\alpha_{ik} = \max_{\{(l_1, l_2)|k=\Re(l_1, l_2)\}} \left\{ \min\left(fa_{i,l_1}, ft_{i,l_2} \right) \right\}, \quad (4)$$

where α_{ik} is the output of inference (i.e. the fire-strength) of question i in fuzzy set k. A matrix of dimension m x l is then obtained as

$$\alpha = [\alpha_{ik}], \quad m \times l,$$

Step 3 (Defuzzification). In this step, fuzzy output values are converted into a single crisp value or final decision. Here, the COG method is applied. The crisp value of question i is obtained by

$$y_i = \int_x x.\mu(x)dx \Big/ \int_x \mu(x)dx, \quad (5)$$

where integrals are taken over the entire range of the output where $\mu(x)$ is obtained from step 2. By using the COG method, a computable and reliable crisp value can be obtained.

The proposed system has been implemented using the Fuzzy Logic Toolbox™ 2.2.7 by MathWorks (http://www.mathworks.com/products/fuzzylogic).
In order to obtain the adjustment vector, each of the three nodes follows the above scheme. The difficulty node has two inputs comprising the accuracy rate and the time rate, and one output displaying the difficulty. The cost node has two inputs comprising difficulty and complexity, and one output displaying the cost. Likewise, the adjustment node has two inputs comprising cost and importance, and one output displaying the resulting adjustment. The adjustment vector, *W*, is then used to obtain the adjusted grade vector of dimension m x 1,

$$\tilde{G} = [\tilde{g}_i], \quad m \times 1,$$

where $\tilde{g}_i$ is the adjusted grade of question i,

$$\tilde{g}_i = g_i \cdot (1 + w_i). \tag{6}$$

Then, the obtained value is scaled to its total grade (i.e. 100) by using the formula

$$\tilde{g}_i = \tilde{g}_i \cdot \sum_j^m g_j \Big/ \sum_j^m \tilde{g}_j \tag{7}$$

Finally, we obtain the adjusted total scores of students by

$$\tilde{S} = A^T \tilde{G}. \tag{8}$$

New and modified ranks of students are then obtained by sorting element values of $\tilde{S}$ in descending order.

Example. By referring to Fig. 1, the average values of *A* and *T* in the difficulty node are computed using formula (2) and formula (3) and then fuzzified, in step 1, to obtain the fuzzy matrices *FA* and *FT*, respectively. Next, as an example for step 2, the output for question 1 in level 4 (fuzzy set "more or less high") is computed based on the rule base given in Table 1a. The computation uses the Mamdani's fuzzy interference mechanism and is obtained by formula (7) as the following:

$$\begin{aligned}
\alpha_{14} &= \max_{\{(l_1, l_2) | \Re(l_1, l_2) = 4\}} \left\{ \min\left(fa_{1,l_1}, ft_{1,l_2} \right) \right\} \\
&= \max_{\{(1,2),(1,3),(2,3),(2,4),(3,4),(3,5),(4,5)\}} \left\{ \min\left(fa_{1,l_1}, ft_{1,l_2} \right) \right\} \\
&= \max\{\min(0,0), \min(0,0.65), \min(0.25,0.65), \\
&\quad \min(0.25,0.35), \min(0.75,0.35), \min(0.75,0), \min(0,0)\} \\
&= \max\{0,0,0.25,0.25,0.35,0,0\} \\
&= 0.35
\end{aligned}$$

By the same procedure, we obtain the inference output, the fire-strength of the difficulty matrix as:

$$\alpha_D = \begin{bmatrix} 0 & 0 & 0.65 & 0.35 & 0 \\ 0 & 0.05 & 0.1 & 0.9 & 0 \\ 0.055 & 0.945 & 0 & 0 & 0 \\ 0 & 0 & 0.85 & 0.15 & 0 \\ 0 & 0.65 & 0.35 & 0.315 & 0 \end{bmatrix}$$

In Step 3, we use COG to compute the crisp value of the difficulty of question 1 by formula (8) as 0.576, as illustrated in Fig. 4. Likewise, we compute the crisp values of other questions as

$$D^T = [0.576 \quad 0.653 \quad 0.299 \quad 0.538 \quad 0.456].$$

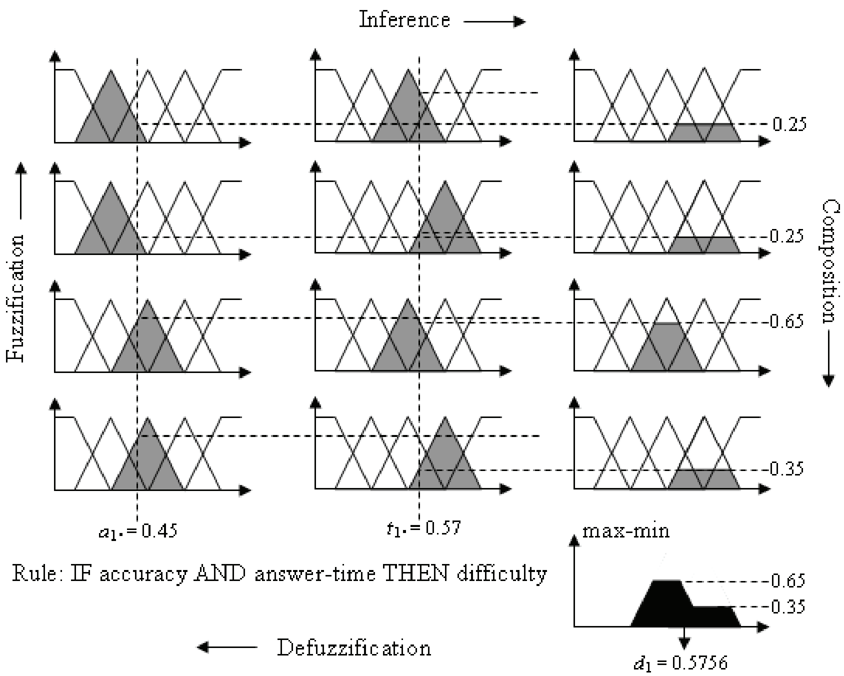


Fig. 4. Fuzzification, Mamdani's max-min inference, and COG to compute the difficulty of question 1.

The surface view of the relation of the rule base in Table 1a is shown in Fig. 5.

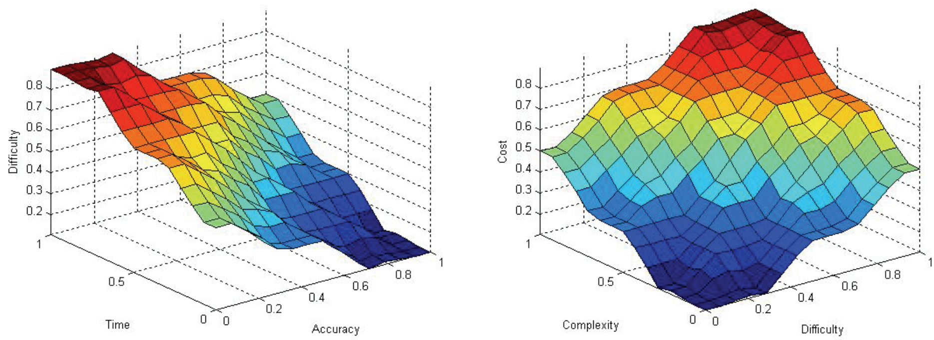


Fig. 5. Surface view of rule base in Table 1a for the difficulty (left) and rule base in Table 1b for the cost (right).

(a) Difficulty					
Accuracy	Time rate				
	1	2	3	4	5
1	3	4	4	5	5
2	2	3	4	4	5
3	2	2	3	4	4
4	1	2	2	3	4
5	1	1	2	2	3

(b) Cost					
Difficulty	Complexity				
	1	2	3	4	5
1	1	1	2	2	3
2	1	2	2	3	4
3	2	2	3	4	4
4	2	3	4	4	5
5	3	4	4	5	5

(c) Adjustment					
Cost	Importance				
	1	2	3	4	5
1	1	1	2	2	3
2	1	2	2	3	4
3	2	2	3	4	4
4	2	3	4	4	5
5	3	4	4	5	5

1: "Low", 2: "more or less low", 3: "medium", 4: "more or less high" and 5: "high".

Table 1. A fuzzy rule base to infer the difficulty, cost, and adjustment

At the next cost node, in Step 1, the crisp values of D obtained at the previous node are fuzzified to obtain the fuzzy difficulty matrix as:

$$FD = \begin{bmatrix} 0 & 0 & 0.622 & 0.378 & 0 \\ 0 & 0 & 0.236 & 0.764 & 0 \\ 0.003 & 0.997 & 0 & 0 & 0 \\ 0 & 0 & 0.811 & 0.189 & 0 \\ 0 & 0.221 & 0.779 & 0 & 0 \end{bmatrix}.$$

In Step 2, based on the rule base given in Table 1b, we obtain the inference output (i.e. the fire-strength) of the cost matrix as a function of the fuzzy difficulty and fuzzy complexity matrices by formula (7) as:

$$\alpha_E = \begin{bmatrix} 0 & 0.622 & 0.378 & 0.15 & 0 \\ 0 & 0 & 0.236 & 0.67 & 0 \\ 0 & 0.003 & 0.994 & 0.31 & 0 \\ 0 & 0.56 & 0.189 & 0 & 0 \\ 0 & 0.221 & 0.7 & 0.3 & 0 \end{bmatrix}.$$

In Step 3, the crisp values of cost are obtained as

$$C^T = [0.424 \quad 0.642 \quad 0.568 \quad 0.354 \quad 0.514].$$

The surface view of the rule base in Table 1b is shown in Fig. 5.
At the final adjustment node, in Step 1, the crisp values of C obtained at the previous node are fuzzified to obtain the fuzzy cost matrix as

$$FC = \begin{bmatrix} 0 & 0.38 & 0.62 & 0 & 0 \\ 0 & 0 & 0.289 & 0.711 & 0 \\ 0 & 0 & 0.661 & 0.339 & 0 \\ 0 & 0.733 & 0.267 & 0 & 0 \\ 0 & 0 & 0.931 & 0.069 & 0 \end{bmatrix}.$$

In Step 2, based on the rule base given in Table 1c, we obtain the inference output (i.e. the fire-strength) of the adjustment matrix as a function of the fuzzy cost and fuzzy importance matrices by formula (7) as:

$$\alpha_W = \begin{bmatrix} 0 & 0 & 0 & 0.62 & 0 \\ 0 & 0.289 & 0.33 & 0.67 & 0 \\ 0 & 0 & 0 & 0.661 & 0.339 \\ 0.733 & 0.267 & 0 & 0 & 0 \\ 0 & 0.07 & 0.93 & 0.069 & 0 \end{bmatrix}.$$

The surface view of the rule base in Table 1c for the adjustment is typically that of effort shown in Fig. 5. In Step 3, the crisp values of adjustment are obtained as:

$$W^T = [0.7 \quad 0.552 \quad 0.749 \quad 0.177 \quad 0.5].$$

Finally, we get the fuzzified adjustment matrix as:

$$FW = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.742 & 0.258 & 0 \\ 0 & 0 & 0 & 0.754 & 0.246 \\ 0.617 & 0.383 & 0 & 0 & 0 \\ 0 & 0.002 & 0.998 & 0 & 0 \end{bmatrix}$$

Now, the adjusted grades are obtained using formula (9) as

$$\tilde{G}^T = [17 \quad 23.272 \quad 34.829 \quad 29.415 \quad 44.990].$$

After scaling to the total score of the manuscript (i.e., 100) using formula (10), we have:

$$\tilde{G}^T = [11.371 \quad 15.566 \quad 23.296 \quad 19.675 \quad 30.092].$$

The new total scores of the students are then obtained using formula (11) as:

$$\begin{array}{ccccccccccc} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 & S_8 & S_9 & S_{10} \\ \tilde{S}^T = [& 67.15 & 53.17 & 42.10 & 52.19 & 48.31 & 51.81 & 48.47 & 49.27 & 85.23 & 51.49], \end{array}$$

With the new ranks being

$$S_9 > S_1 > S_2 > S_4 > S_6 > S_{10} > S_8 > S_7 > S_5 > S_3.$$

□

3. Three nodes system based on Gaussian membership functions

The three nodes fuzzy evaluation system described in Section 2 is based on triangular MF's which are the simplest MF's formed using straight lines. Triangular MFs are defined by three parameters and there is no way to acquire its optimum values. It was noted that when theses parameters are changed slightly, different ranking orders are obtained which could impair the system's reliability. In order to avoid losing reliability and having a robust system, the system should be able to give the same ranking orders without changing students' scores and for various values of these parameters. As an alternative approach, Gaussian MF's are proposed. Gaussian MF's are suitable for problems which require continuously differentiable curves and therefore smooth transitions, whereas the triangular do not posses these abilities. Gaussian MF's are defined by two parameters which is one parameter less than that of the triangular MF's. The tuning of a reduced number of parameters will result in a reduced Degree Of Freedom (DOF) and hence a more robust system. Gaussian MF's is defined as

$$\mu_{A_i}(x) = e^{-\frac{1}{2}(x - c_i/\sigma_i)^2}, \tag{9}$$

Where c_i is the center (i.e., mean) of the i^{th} fuzzy set and σ_i is the width (i.e., standard deviation) of the i^{th} fuzzy set, which have by nature, infinite support (i.e., every control point influences the whole calculations of the output) (Lohani et al., 2007). Therefore, for Gaussian MF's with wide widths it is possible to obtain a membership degree to each fuzzy set greater than 0 and hence every rule in the rule base fires. Consequently, the relationship between input and output can be described accurate enough. Here, the centres of the five Gaussian MF's are chosen to be the same as that of the triangular MFs shown in Fig. 3 (i.e. [0.1 0.3 0.5 0.7 0.9]). Gaussian MF's of the five levels for $\sigma = 0.1$ are shown in Fig. 6. From Fig. 6, it is obvious that Gaussian MF's provide more continuous transition from one interval to another and hence provides smoother control surface from the fuzzy rules. The surface view of the rule base in Table 1a and b for Gaussian MFs of $\sigma = 0.1$ are shown in Fig. 7.

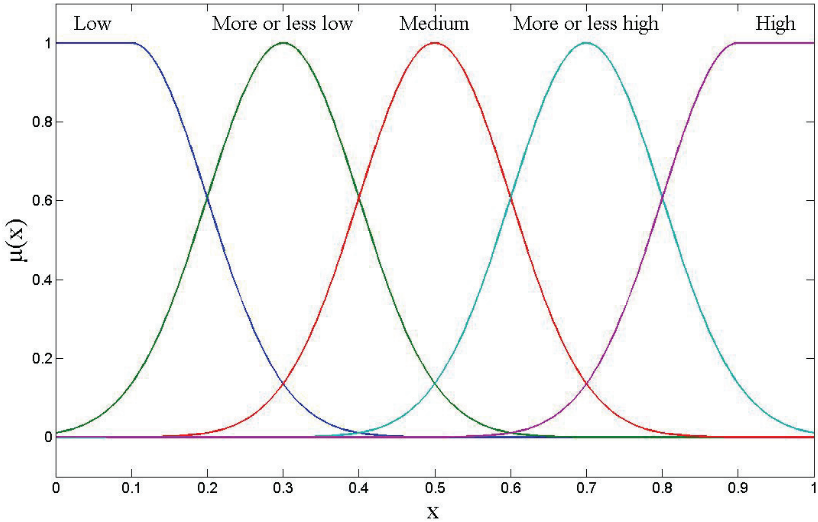


Fig. 6. Gaussian membership functions of the five levels for σ = 0.1.

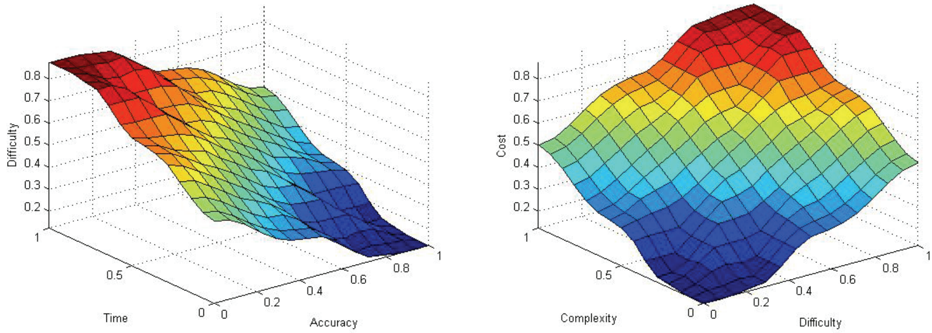


Fig. 7. Surface view of rule base in Table 1a (left) and in Table 1b (right) for Gaussian MFs of σ = 0.1.

Example. The width of the Gaussian MF's, σ, is varied between 0.05 and 12 in steps of 0.05 and the three nodes fuzzy system based on Gaussian MF's is applied to the same example introduced in Section 2. The new total scores of the students are then obtained using formula (11) as shown in Table 2.

The mean of the new scores of the 10 students is computed for each value of the membership width, σ, and is shown in Fig. 8. From the figure, it is obvious that the mean of the new scores is equal to the mean of the classical scores obtained using formula (1) for membership width of 4.0 and higher. Although the new scores, rounded to two digits, are equal, the system is still able to give the correct ranking order of the students with equal total scores.

σ	1>	2>	3>	4>	5>	6>	>7	>8	>9	10
0.05	9	1	2	10	4	6	5	7	8	3
	84.59	65.00	52.58	51.43	50.80	50.22	48.31	48.30	48.16	43.22
0.10	9	1	2	4	6	10	8	5	7	3
	84.36	64.27	53.24	52.12	51.78	51.43	49.49	48.44	48.31	41.77
0.15	9	1	2	4	6	10	8	5	7	3
	84.54	64.55	53.59	51.49	50.93	50.89	50.44	48.65	47.80	40.91
0.20	9	1	2	8	4	10	6	5	7	3
	84.66	64.61	53.78	51.12	50.87	50.44	50.13	48.96	47.26	40.09
0.25	9	1	2	8	4	10	6	5	7	3
	84.74	64.61	53.87	51.52	50.49	50.18	49.67	49.17	46.90	39.57
0.30	9	1	2	8	4	10	6	5	7	3
	84.79	64.61	53.93	51.74	50.26	50.03	49.40	49.31	46.68	39.25
0.35	9	1	2	8	4	10	5	6	7	3
	84.83	64.61	53.96	51.89	50.12	49.94	49.41	49.24	46.54	39.04
4.0~12.0	9	1	2	8	4	10	5	6	7	3
	84.95	64.60	54.05	52.30	49.70	49.70	49.70	48.80	46.10	38.40

Table 2. New total scores and new ranking orders of 10 students using the three fuzzy nodes system based on Gaussian MF's

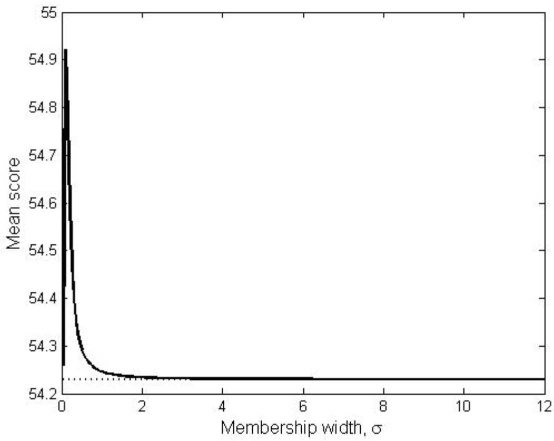


Fig. 8. Mean scores of the 10 students ranked using classical method (dotted line) and the three nodes fuzzy system based on Gaussian MF's (solid line).

□

4. Comparison of the methods

The ranking order has been obtained when the three nodes fuzzy system based on triangular MF's are applied to students with equal total scores. When the three nodes fuzzy system based on triangular membership functions is applied to all students, new scores and hence new rankings are obtained. When the three nodes system based on Gaussian membership of width of 4.0 is applied to all students, the resultant new total scores of students rounded to two digits are equal to that of the classical scores but with new ranking orders. Ranking orders and scores of students are shown in Table 3.

Ranking method	1>	2>	3>	4>	5>	6>	>7	>8	>9	10
Classical	9	1	2	8	4=	5=	10=	6	7	3
	84.95	64.60	54.05	52.30	49.70	49.70	49.70	48.80	46.10	38.40
3 nodes based triangular (3)	9	1	2	8	4	10	5	6	7	3
	84.95	64.60	54.05	52.30	52.19	51.49	48.31	48.80	46.10	38.40
3 nodes based triangular (10)	9	1	2	4	6	10	8	7	5	3
	85.23	67.15	53.17	52.19	51.81	51.49	49.27	48.47	48.31	42.10
3 nodes based Gaussian (10)	9	1	2	8	4	10	5	6	7	3
	84.95	64.60	54.05	52.30	49.70	49.70	49.70	48.80	46.10	38.40

(3): Applied to 3 students with equal score, (10): applied to 10 (i.e. all) students

Table 3. Ranking orders and new scores obtained by the three methods.

From Table 3 it is seen that the same ranking order has been obtained when the three nodes fuzzy system based on triangular MF's is applied to only students with total scores and the three nodes fuzzy system based on Gaussian MF's applied to all students. The varying of the parameters of the triangular MF's results in different scores and different ranking orders while the same scores and the same ranking orders are obtained for Gaussian MF's of various widths.

5. Conclusions

In this chapter, we proposed Gaussian MF's to represent the fuzzy sets (i.e., levels) representing the importance, the complexity and the difficulty of the questions given to students. Results show that using Gaussian MFs with a width value (i.e., standard deviation) of 0.4 and higher provide a more reliable evaluation system which is able to provide new ranking orders without changing students' total scores. Gaussian MFs provide smooth transition between levels and provides a way to fire the maximum number of rules in the rule base and hence a more accurate representation of the input-output relationship is achieved. Gaussian MF's also provides a system with less degree of freedom and hence more robustness. The proposed three nodes fuzzy evaluation system based on Gaussian MF's provides a new ranking order while keeping the scores of students unchanged. The system is implemented by using the Fuzzy Logic Toolbox™ 2.2.7 by MathWorks®.

6. References

Bai, S.-M. & Chen, S.-M. (2008a). Automatically constructing grade membership functions of fuzzy rules for students' evaluation. *Expert Systems with Applications*, 35, 3, pp. 1408-1414.

Bai, S.-M. & Chen, S.-M. (2008b). Evaluating students' learning achievement using fuzzy membership functions and fuzzy rules. *Expert Systems with Applications*, 34, pp. 399-410.

Biswas, R. (1995). An application of fuzzy sets in students' evaluation. *Fuzzy Sets and Systems*, 74, 2, pp. 187-194.

Chen, S.M. & Lee, C.H. (1999). New methods for students' evaluating using fuzzy sets. *Fuzzy Sets and Systems*, 104, 2, pp. 209–218.

Echauz, J.R., & Vachtsevanos, G. J. (1995). Fuzzy grading system. IEEE Transactions on Education, 38(2), 158-165.

Fuzzy Logic Toolbox™ 2.2.7 from http://www.mathworks.com/products/fuzzylogic/.

Larsen, P.M. (1980). Industrial applications of fuzzy logic control. *International Journal of Man-Machine Studies*, 12, 1, pp. 3–10.

Law, C.K. (1996). Using fuzzy numbers in education grading system. *Fuzzy Sets and Systems*, 83, 3, pp. 311-323.

Lohani A.K. ; Goel, N.K. & Bhatia, K.K.S. (2007). Reply to comments provided by Z. Şen on "Takagi–Sugeno fuzzy system for modeling stage-discharge relationship" by A.K. Lohani, N.K. Goel and K.K.S. Bhatia. *Journal of Hydrology*, 337, 1-2, pp. 244-247.

Ma, J. & Zhou, D. (2000). Fuzzy set approach to the assessment of student-centered learning. *IEEE Transactions on Education*, 43, 2, pp. 237-241.

Mamdani, E.M. & Assilian, S. (1975). An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7, pp. 1–13.

Mamdani, E.M. (1974). Applications of fuzzy algorithms for simple dynamic plants. *Proceedings of the IEEE*, 21, 2, pp. 1585–1588.

Saleh, I. ; Kim, S.-I. (2009). A fuzzy system for evaluating students' learning achievement. *Expert Systems with Applications*, 36, 3, pp. 6236-6243.

Wang, H.Y. & Chen, S.M. (2008). Evaluating students' answerscripts using fuzzy numbers associated with degrees of confidence. *IEEE Transactions on Fuzzy Systems*, 16, 2, pp. 403-415.

Weon, S. & Kim, J. (2001). Learning achievement evaluation strategy using fuzzy membership function, *Proceedings of the 31st ASEE/IEEE Frontiers in Education Conference*, Reno: NV.

Wilson, E. ; Karr, C.L. & Freeman, L.M. (1998). Flexible, adaptive, automatic fuzzy-based grade assigning system, *Proceedings of the North American Fuzzy Information Processing Society Conference*, pp. 334-338.

Zadeh, L.A. (1965). Fuzzy sets. *Information and Control*, 8, pp. 338-3

2

Control Design of Fuzzy Systems with Immeasurable Premise Variables

Jun Yoneyama and Tomoaki Ishihara
Aoyama Gakuin University
Japan

1. Introduction

The Takagi-Sugeno fuzzy model is described by fuzzy if-then rules which represent local linear systems of the underlining nonlinear systems (Takagi & Sugeno, 1985; Tanaka et al., 1996; Tanaka & Sugeno, 1992) and thus it can describe a wide class of nonlinear systems. In the last decade, nonlinear control design methods based on Takagi-Sugeno fuzzy system have been explored. Since the stability analysis and state feedback stabilization first made in (Tanaka & Sugeno, 1992), system theory and various control schemes for fuzzy systems have been developed. Parallel to state feedback control design, observer problems were also considered in (Tanaka & Sano, 1994; Tanaka & Wang, 1997; Yoneyama et al., 2000; Yoneyama et al., 2001a). When the observer is available, it is known in (Yoneyama et al., 2000) that for fuzzy systems, the separation property of designing state feedback controller and observer is established. Thus, an output feedback controller for fuzzy systems was proposed in (Yoneyama et al., 2001a). Theory was extended to H_∞ control (Cao et al., 1996; Chen et al., 2005; Feng et al., 1996; Hong & Langari, 1998; Katayama & Ichikawa, 2002; Yoneyama et al., 2001b). In spite of these developments in fuzzy system control theory, the separation property holds only for a limited class of fuzzy systems where the premise variable is measurable.
When we consider a fuzzy system, the selection of the premise variables plays an important role in system representation. The premise variable is usually given, and hence the output is natually selected. In this case, however, a class of fuzzy systems is limited. If the premise variable is the state of the system, a fuzzy system can represent the widest class of nonlinear systems. In this case, output feedback controller design is difficult because the state variable is immeasurable and it is not available for the premise variable of an output feedback controller. For this class of fuzzy systems, output feedback control design schemes based on parallel distributed compensator (PDC) have been considerd in (Ma et al., 1998; Tanaka & Wang, 2001; Yoneyama et al., 2001a) where the premise variable of the controller was replaced by its estimate. Furthermore, Linear Matrix Inequality (Boyd et al, 1994) approach was introduced in (Guerra et al., 2006). Uncertain system approach was taken for stabilizatin and H_∞ control of both continuous-time and discrete-time fuzzy systems in (Assawinchaichote, 2006; Tanaka et al., 1998; Yoneyama, 2008a; Yoneyama, 2008b). However, controller design conditions given in these approaches are still conservative.
This chapter is concerned with robust H_∞ output feecback controller design for a class of uncertain continuous-time Takagi-Sugeno fuzzy systems where the premise variable is the immeasurable state variable. This class of fuzzy systems covers a general nonlinear system

and its output feedback control problem is of practical importance. First, it is shown that Takagi-Sugeno fuzzy system with immeasurable premise variables can be written as an uncertain linear system, and robust stability with H_∞ disturbance attenuation via output feedback control for a fuzzy system is converted into the same problem for an uncertain linear system. Then, an original robust control problem is shown to be equivalent to stailization with H_∞ disturbance attenuation problem for nominal system with no uncertainty. The discrete-time counterpart of the same robust control problems is also considered. Numerical examples of both continuous-time and discrete-time fuzzy systems are shown to illustrate our design methods. Finally, an extension to Takagi-Sugeno fuzzy time-delay systems is given. The same technique is used to write a fuzzy time-delay system as an uncertain linear time-delay system.

2. Robust output feedback control for uncertain continuous-time fuzzy systems with immeasurable premise variables

We consider robust stability with H_∞ disturbance attenuation problems via output feedback control for continuous-time Takagi-Sugeno fuzzy systems with immeasurable premise variables. We first show that such a fuzzy system can be written as an uncertain linear system. Then, robust stability with H_∞ disturbance attenuation problem for an uncertain system is converted into stability with H_∞ disturbance attenuation problem for a nominal system. Based on such a relationship, a solution to a robust stability with H_∞ disturbance attenuation via output feedback control for a fuzzy system with immeasurable premise variables is given. Finally, a numerical example illustrates our theory.

2.1 Fuzzy systems and problem formulation

In this section, we introduce continuous-time Takagi-Sugeno fuzzy systems with immeasurable premise variables. Consider the Takagi-Sugeno fuzzy model, described by the following IF-THEN rule:

$$
\begin{aligned}
&\textbf{IF} && \xi_1 \text{ is } M_{i1} \text{ and } \dots \text{ and } \xi_p \text{ is } M_{ip},\\
&\textbf{THEN} && \dot{x}(t) = (A_i + \Delta A_i)x(t) + (B_{1i} + \Delta B_{1i})w(t) + (B_{2i} + \Delta B_{2i})u(t),\\
& && z(t) = (C_{1i} + \Delta C_{1i})x(t) + (D_{11i} + \Delta D_{11i})w(t) + (D_{12i} + \Delta D_{12i})u(t),\\
& && y(t) = (C_{2i} + \Delta C_{2i})x(t) + (D_{21i} + \Delta D_{21i})w(t) + (D_{22i} + \Delta D_{22i})u(t), \quad i = 1,\cdots,r
\end{aligned}
$$

where $x(t) \in \Re^n$ is the state, $w(t) \in \Re^{m_1}$ is the disturbance, $u(t) \in \Re^{m_2}$ is the control input, $z(t) \in \Re^{q_1}$ is the controlled output, $y(t) \in \Re^{q_2}$ is the measurement output. r is the number of IF-THEN rules. M_{ij} is a fuzzy set and $\xi_1, \cdots, \xi_p$ are premise variables. We set $\xi = [\xi_1 \ \cdots \ \xi_p]^T$. We assume that the premise variables do not to depend on $u(t)$. A_i, B_{1i}, B_{2i}, C_{1i}, C_{2i}, D_{11i}, D_{12i}, D_{21i} and D_{22i} are constant matrices of appropriate dimensions. The uncertain matrices are of the form

$$
\begin{bmatrix} \Delta A_i & \Delta B_{1i} & \Delta B_{2i} \\ \Delta C_{1i} & \Delta D_{11i} & \Delta D_{12i} \\ \Delta C_{2i} & \Delta D_{21i} & \Delta D_{22i} \end{bmatrix} = \begin{bmatrix} H_{1i} \\ H_{2i} \\ H_{3i} \end{bmatrix} F_i(t) \begin{bmatrix} E_{1i} & E_{2i} & E_{3i} \end{bmatrix}, \ i = 1,\cdots,r \tag{1}
$$

where each $F_i(t) \in \Re^{l \times j}$ is an uncertain matrix satisfying $F_i^T(t)F_i(t) \le I$, and H_{1i}, H_{2i}, H_{3i}, E_{1i}, E_{2i} and E_{3i} are known constant matrices of appropriate dimensions.

Assumption 2.1 The system (A_r, B_{1r}, B_{2r}, C_{1r}, C_{2r}, D_{11r}, D_{12r}, D_{21r}, D_{22r}) represents a nominal system that can be chosen as a subsystem including the equilibrium point of the original system.

Throughout the chapter, we assume $\xi(t)$ is a function of the immeasurable state $x(t)$. Then, the state equation, the controlled output and the output equation are defined as follows:

$$\begin{aligned} \dot{x}(t) &= \sum_{i=1}^{r} \lambda_i(x(t))\{(A_i + \Delta A_i)x(t) + (B_{1i} + \Delta B_{1i})w(t) + (B_{2i} + \Delta B_{2i})u(t)\}, \\ z(t) &= \sum_{i=1}^{r} \lambda_i(x(t))\{(C_{1i} + \Delta C_{1i})x(t) + (D_{11i} + \Delta D_{11i})w(t) + (D_{12i} + \Delta D_{12i})u(t)\}, \\ y(t) &= \sum_{i=1}^{r} \lambda_i(x(t))\{(C_{2i} + \Delta C_{2i})x(t) + (D_{21i} + \Delta D_{21i})w(t) + (D_{22i} + \Delta D_{22i})u(t)\} \end{aligned} \tag{2}$$

where

$$\lambda_i(x) = \frac{\beta_i(x)}{\sum_{i=1}^{r} \beta_i(x)}, \quad \beta_i(x) = \prod_{j=1}^{p} M_{ij}(x_j) \tag{3}$$

and $M_{ij}(\cdot)$ is the grade of the membership function of M_{ij}. We assume

$$\beta_i(x(t)) \ge 0, \quad i = 1, \cdots, r, \quad \sum_{i=1}^{r} \beta_i(x(t)) > 0$$

for any $x(t)$. Hence $\lambda_i(x(t))$ satisfies

$$\lambda_i(x(t)) \ge 0, \quad i = 1, \cdots, r, \quad \sum_{i=1}^{r} \lambda_i(x(t)) = 1 \tag{4}$$

for any $x(t)$. Our problem is to find a control $u(\cdot)$ for the system (2) given the output measurements $y(\cdot)$ such that the controlled output $z(\cdot)$ satisfies

$$\int_0^{\infty} z^T(t)z(t)dt < \gamma^2 \int_0^{\infty} w^T(t)w(t)dt \tag{5}$$

for a prescribed scalar $\gamma > 0$. If such a controller exists, we call it a robust H_∞ output feedback controller. Here, we consider the robust stabilization and the robust H_∞ disturbance attenuation of uncertain fuzzy system (2) with the immeasurable premise variable.

2.2 Robust stability of equivalent uncertain systems

When we consider the H_∞ output feedback control problem for fuzzy systems, the selection of the premise variables plays an important role. The premise variable is usually given, and so the output is selected as the premise variable. In this case, however, the system covers a very limited class of nonlinear systems. When the premise variable is the state of the system, a fuzzy system describes a widest class of nonlinear systems. However, the controller design based on PDC concept is infeasible due to the immeasurable premise variable imposed on the control law (Tanaka & Wang, 2001; Yoneyama et al., 2001a; Yoneyama et al., 2001b). To avoid a difficulty due to the parallel distributed compensation (PDC) concept, we rewrite a fuzzy system, and consider a controller design for an equivalent uncertain system. We rewrite (2) as an uncertain system. Since $\lambda_i(x(t))$ satisfies (4), we have

$$\lambda_r(x(t)) = 1 - [\lambda_1(x(t)) + \lambda_2(x(t)) + \cdots + \lambda_{r-1}(x(t))].$$

It follows from this relationship that

$$\begin{aligned}\sum_{i=1}^{r}\lambda_i(x(t))(A_i + \Delta A_i) &= A_r + H_{1r}F(t)E_{1r} + \lambda_1(x(t))[A_{1r} + \overline{H}_{1r}\overline{F}_{1r}(t)\overline{E}_{1r}] \\ &\quad + \cdots + \lambda_{r-1}(x(t))[A_{r-1,r} + \overline{H}_{r-1,r}\overline{F}_{r-1,r}(t)\overline{E}_{r-1,r}] \\ &= A_r + H_1\widetilde{F}(t)\widetilde{A}\end{aligned}$$

where

$$\begin{aligned} A_{ir} &= A_i - A_r, \quad i = 1,\cdots,r-1 \\ H_1 &= [I \quad I \quad \cdots \quad I \quad H_{11} \quad H_{12} \quad \cdots \quad H_{1r}], \\ \tilde{F}(t) &= \mathrm{diag}[\lambda_1(x(t))I \quad \cdots \quad \lambda_{r-1}(x(t))I \quad \lambda_1(x(t))F_1(t) \quad \cdots \quad \lambda_r(x(t))F_r(t)], \\ \widetilde{A} &= [A_{1r}^T \quad A_{2r}^T \quad \cdots \quad A_{r-1,r}^T \quad E_{11}^T \quad E_{12}^T \quad \cdots \quad E_{1r}^T]^T. \end{aligned}$$

Similarly, we rewrite other matrices and have an equivalent description for (1):

$$\begin{aligned} \dot{x}(t) &= (A_r + \Delta A)x(t) + (B_{1r} + \Delta B_1)w(t) + (B_{2r} + \Delta B_2)u(t) \\ &\triangleq A_\Delta x(t) + B_{1\Delta}w(t) + B_{2\Delta}u(t), \\ z(t) &= (C_{1r} + \Delta C_1)x(t) + (D_{11r} + \Delta D_{11})w(t) + (D_{12r} + \Delta D_{12})u(t) \\ &\triangleq C_{1\Delta}x(t) + D_{11\Delta}w(t) + D_{12\Delta}u(t), \\ y(t) &= (C_{2r} + \Delta C_2)x(t) + (D_{21r} + \Delta D_{21})w(t) + (D_{22r} + \Delta D_{22})u(t) \\ &\triangleq C_{2\Delta}x(t) + D_{21\Delta}w(t) + D_{22\Delta}u(t) \end{aligned} \qquad (6)$$

where

$$\begin{bmatrix} \Delta A & \Delta B_1 & \Delta B_2 \\ \Delta C_1 & \Delta D_{11} & \Delta D_{12} \\ \Delta C_2 & \Delta D_{21} & \Delta D_{22} \end{bmatrix} = \begin{bmatrix} H_1 & 0 & 0 \\ 0 & H_2 & 0 \\ 0 & 0 & H_3 \end{bmatrix} \begin{bmatrix} \widetilde{F}(t) & 0 & 0 \\ 0 & \widetilde{F}(t) & 0 \\ 0 & 0 & \widetilde{F}(t) \end{bmatrix} \begin{bmatrix} \widetilde{A} & \widetilde{B}_1 & \widetilde{B}_2 \\ \widetilde{C}_1 & \widetilde{D}_{11} & \widetilde{D}_{12} \\ \widetilde{C}_2 & \widetilde{D}_{21} & \widetilde{D}_{22} \end{bmatrix}, \qquad (7)$$

$$\begin{aligned} H_2 &= [I \quad I \quad \cdots \quad I \quad H_{21} \quad H_{22} \quad \cdots \quad H_{2r}], \\ H_3 &= [I \quad I \quad \cdots \quad I \quad H_{31} \quad H_{32} \quad \cdots \quad H_{3r}], \\ B_{1ir} &= B_{1i} - B_{1r}, \quad B_{2ir} = B_{2i} - B_{2r}, \quad C_{1ir} = C_{1i} - C_{1r}, \\ C_{2ir} &= C_{2i} - C_{2r}, \quad D_{11ir} = D_{11i} - D_{11r}, \quad D_{12ir} = D_{12i} - D_{12r}, \\ D_{21ir} &= D_{21i} - D_{21r}, \quad D_{22ir} = D_{22i} - D_{22r}, \quad i = 1,\cdots,r-1, \\ \widetilde{B}_1 &= [B_{11r}^T \quad B_{12r}^T \quad \cdots \quad B_{1r-1,r}^T \quad E_{21}^T \quad E_{22}^T \quad \cdots \quad E_{2r}^T]^T, \end{aligned}$$

$$\widetilde{B}_2 = [B_{21r}^T \quad B_{22r}^T \quad \cdots \quad B_{2r-1,r}^T \quad E_{31}^T \quad E_{32}^T \quad \cdots \quad E_{3r}^T]^T,$$
$$\widetilde{C}_1 = [C_{11r}^T \quad C_{12r}^T \quad \cdots \quad C_{1r-1,r}^T \quad E_{11}^T \quad E_{12}^T \quad \cdots \quad E_{1r}^T]^T,$$
$$\widetilde{C}_2 = [C_{21r}^T \quad C_{22r}^T \quad \cdots \quad C_{2r-1,r}^T \quad E_{11}^T \quad E_{12}^T \quad \cdots \quad E_{1r}^T]^T,$$
$$\widetilde{D}_{11} = [D_{111r}^T \quad D_{112r}^T \quad \cdots \quad D_{11r-1,r}^T \quad E_{21}^T \quad E_{22}^T \quad \cdots \quad E_{2r}^T]^T,$$
$$\widetilde{D}_{12} = [D_{121r}^T \quad D_{122r}^T \quad \cdots \quad D_{12r-1,r}^T \quad E_{31}^T \quad E_{32}^T \quad \cdots \quad E_{3r}^T]^T,$$
$$\widetilde{D}_{21} = [D_{211r}^T \quad D_{212r}^T \quad \cdots \quad D_{21r-1,r}^T \quad E_{21}^T \quad E_{22}^T \quad \cdots \quad E_{2r}^T]^T,$$
$$\widetilde{D}_{22} = [D_{221r}^T \quad D_{222r}^T \quad \cdots \quad D_{22r-1,r}^T \quad E_{31}^T \quad E_{32}^T \quad \cdots \quad E_{3r}^T]^T.$$

We note that since the state $x(t)$ is not measurable, $\lambda_i(x(t))$ is unknown. This implies that $\widetilde{F}(t)$ is a time varying unknown function. However, it is easy to see that $\widetilde{F}(t)$ satisfies $\widetilde{F}^T(t)\widetilde{F}(t) \le I$, because $0 \le \lambda_i(x(t)) \le 1$, $F_i^T(t)F_i(t) \le I$, $i = 1,\cdots,r$. Hence, we can see (6) as a linear system with time varying uncertainties.

Remark 2.1 Representation (6) has less uncertain matrices than that of (Yoneyama, 2008a; Yoneyama, 2008b), which leads to less conservative results. H_1 and $\widetilde{A}$ in (7) are not unique and can be chosen such that $H_1\widetilde{F}(t)\widetilde{A} = H_{1r}F(t)E_{1r} + \lambda_1(x(t))[A_{1r} + \overline{H}_{1r}\overline{F}_{1r}(t)\overline{E}_{1r}] + \cdots + \lambda_{r-1}(x(t))[A_{r-1,r} + \overline{H}_{r-1,r}\overline{F}_{r-1,r}(t)\overline{E}_{r-1,r}]$. This is true for other matrices H_i, $i = 2,3$ and $\widetilde{B}_1$, $\widetilde{B}_2$, $\widetilde{C}_1$, $\widetilde{C}_2$, $\widetilde{D}_{11}$, $\widetilde{D}_{12}$, $\widetilde{D}_{21}$, $\widetilde{D}_{22}$.

Now, our problem of finding a robust stabilizing output feedback controller with H∞ disturbance attenuation for the system (2) is to find a controller of the form (8) for the uncertain system (6).

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}y(t), \\ u(t) &= \hat{C}\hat{x}(t). \end{aligned} \tag{8}$$

Definition 2.1 (i) Consider the unforced system (6) with $w(t) = 0$, $u(t) = 0$. The uncertain system (6) is said to be robustly stable if there exists a matrix $X > 0$ such that

$$A_\Delta^T X + XA_\Delta < 0$$

for all admissible uncertainties.

(ii) The uncertain system (6) is said to be robustly stabilizable via output feedback controller if there exists an output feedback controller of the form (4) such that the resulting closed-loop system (6) with (4) is robustly stable.

Definition 2.2 (i) Given a scalar $\gamma > 0$, the system (6) is said to be robustly stable with H∞ disturbance attenuation γ if there exists a matrix $X > 0$ such that

$$\begin{bmatrix} A_\Delta^T X + XA_\Delta & XB_{1\Delta} & C_{1\Delta}^T \\ B_{1\Delta}^T X & -\gamma^2 I & D_{11\Delta}^T \\ C_{1\Delta} & D_{11\Delta} & -I \end{bmatrix} < 0.$$

(ii) Given a scalar $\gamma > 0$, the uncertain system (6) is said to be robustly stabilizable with H∞ disturbance attenuation γ via output feedback controller if there exists an output feedback controller of the form (8) such that the resulting closed-loop system (6) with (8) is robustly stable with H∞ disturbance attenuation γ.

Definition 2.3 Given a scalar $\gamma > 0$, the system

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bw(t),\\ z(t) &= Cx(t) + Dw(t)\end{aligned}$$

is said to be stable with H∞ disturbance attenuation γ if it is exponentially stable and input-output stable with (5).

The following lemma is well known, and we need it to prove our main results.

Lemma 2.1 (Xie, 1996) Given constant matrices $Q = Q^T$, H, E of appropriate dimensions. Suppose a time varying matrix $F(t)$ satisfies $F^T(t)F(t) \le I$. Then, the following holds

$$Q + E^T F^T(t) H^T + HF(t)E \le Q + \frac{1}{\varepsilon} HH^T + \varepsilon E^T E$$

for some $\varepsilon > 0$.

Now, we state our key results that show the relationship between robust stability with H∞ disturbance attenuation and stability with H∞ disturbance attenuation.

Theorem 2.1 The system (6) with $w(t) = 0$ is robustly stable with H∞ disturbance attenuation γ if and only if for $\varepsilon > 0$ that

$$\begin{aligned}\dot{x}(t) &= A_r x(t) + [\gamma^{-1}B_{1r} \quad \varepsilon^{-1}H_1 \quad 0]\widetilde{w}(t),\\ \widetilde{z}(t) &= \begin{bmatrix} C_{1r} \\ \varepsilon A \\ \varepsilon\widetilde{C}_1 \end{bmatrix} x(t) + \begin{bmatrix} \gamma^{-1}D_{11r} & 0 & \varepsilon^{-1}H_2 \\ \varepsilon\gamma^{-1}\widetilde{B}_1 & 0 & 0 \\ \varepsilon\gamma^{-1}\widetilde{D}_{11r} & 0 & 0 \end{bmatrix}\widetilde{w}(t)\end{aligned}$$

is stable with unitary H∞ disturbance attenuation $\gamma = 1$.

Proof: The system (6) is robustly stable with H∞ disturbance attenuation γ if and only if there exists a matrix $X > 0$ such that

$$\begin{bmatrix} (A_r + H_1\widetilde{F}_1(t)\widetilde{A})^T X + X(A_r + H_1\widetilde{F}_1(t)\widetilde{A}) & X(B_{1r} + H_1\widetilde{F}_1(t)\widetilde{B}_1) & (C_{1r} + H_2\widetilde{F}_2(t)\widetilde{C}_1)^T \\ (B_{1r} + H_1\widetilde{F}_1(t)\widetilde{B}_1)^T X & -\gamma^2 I & (D_{11r} + H_2\widetilde{F}_2(t)\widetilde{D}_{11})^T \\ (C_{1r} + H_2\widetilde{F}_2(t)\widetilde{C}_1) & D_{11r} + H_2\widetilde{F}_2(t)\widetilde{D}_{11} & -I \end{bmatrix} < 0,$$

which can be written as

$$\hat{A} + \hat{H}\hat{F}(t)\hat{E} + \hat{E}^T\hat{F}^T(t)\hat{H}^T < 0 \tag{9}$$

where

$$\hat{A} = \begin{bmatrix} A_r^T X + XA_r & XB_{1r} & C_{1r}^T \\ B_{1r}^T X & -\gamma^2 I & D_{11r}^T \\ C_{1r} & D_{11r} & -I \end{bmatrix}, \ \hat{H} = \begin{bmatrix} XH_1 & 0 \\ 0 & 0 \\ 0 & H_2 \end{bmatrix}, \ \hat{F}(t) = \begin{bmatrix} \widetilde{F}_1(t) & 0 \\ 0 & \widetilde{F}_2(t) \end{bmatrix}, \ \hat{E} = \begin{bmatrix} \widetilde{A} & \widetilde{B}_1 & 0 \\ \widetilde{C}_1 & \widetilde{D}_{11} & 0 \end{bmatrix}.$$

It can be shown by Lemma 2.1 that there exists $X > 0$ such that (9) holds if and only if there exist $X > 0$ and a scalar $\varepsilon > 0$ such that

$$\hat{A} + \frac{1}{\varepsilon^2}\hat{H}\hat{H}^T + \varepsilon^2 \hat{E}^T \hat{E} < 0,$$

which can be written as

$$\begin{bmatrix} \hat{A} & \varepsilon^{-1}\hat{H} & \varepsilon\hat{E}^T \\ \varepsilon^{-1}\hat{H}^T & -I & 0 \\ \varepsilon\hat{E} & 0 & -I \end{bmatrix} < 0.$$

Pre-multiplying and post-multiplying the above LMI by

$$\begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma^{-1}I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix},$$

and its transpose, respectively, we have

$$\begin{bmatrix} A_r^T X + XA_r & X\widetilde{B} & \widetilde{C}^T \\ \widetilde{B}^T X & -I & \widetilde{D}^T \\ \widetilde{C} & \widetilde{D} & -I \end{bmatrix} < 0.$$

The result follows from Definition 2.1.

Theorem 2.2 The system (6) with $w(t) = 0,\ u(t) = 0$ is robustly stable if and only if for $\varepsilon > 0$ the system

$$\begin{aligned} \dot{x}(t) &= A_r x(t) + \varepsilon^{-1} H_1 \widetilde{w}(t), \\ \widetilde{z}(t) &= \varepsilon \widetilde{A} x(t) \end{aligned}$$

is stable with unitary H_∞ disturbance attenuation $\gamma = 1$.

Proof: The system (6) is robustly stable if and only if there exists a matrix $X > 0$ such that

$$(A_r + H_1\widetilde{F}(t)\widetilde{A})^T X + X(A_r + H_1\widetilde{F}(t)\widetilde{A}) < 0.$$

It can be shown by Lemma 2.1 that there exists $X > 0$ such that the above inequality holds if and only if there exist $X > 0$ and a scalar $\varepsilon > 0$ such that

$$A_r^T X + XA_r + \frac{1}{\varepsilon^2} XH_1 H_1^T X + \varepsilon^2 \widetilde{A}^T \widetilde{A} < 0,$$

which can be written as

$$\begin{bmatrix} A_r^T X + XA_r & \varepsilon^{-1} XH_1 & \varepsilon \widetilde{A}^T \\ \varepsilon^{-1} H_1^T X & -I & 0 \\ \varepsilon \widetilde{A} & 0 & -I \end{bmatrix} < 0.$$

The result follows from Definition 2.1.

2.3 Robust controller design

Based on the results in the previous section, we consider the design of robust H_∞ output feedback controller for the system (2). Such a controller can be designed for a nominal linear system with no uncertainty. For the following two auxiliary systems:

$$\begin{aligned} \dot{x}(t) &= A_r x(t) + [\gamma^{-1} B_{1r} \quad \varepsilon^{-1} H_1 \quad 0 \quad 0] \widetilde{w}(t) + B_{2r} u(t), \\ \widetilde{z}(t) &= \begin{bmatrix} C_{1r} \\ \varepsilon \widetilde{A} \\ \varepsilon \widetilde{C}_2 \\ \varepsilon \widetilde{C}_1 \end{bmatrix} x(t) + \begin{bmatrix} \gamma^{-1} D_{11r} & 0 & 0 & \varepsilon^{-1} H_2 \\ \varepsilon \gamma^{-1} \widetilde{B}_1 & 0 & 0 & 0 \\ \varepsilon \gamma^{-1} \widetilde{D}_{21} & 0 & 0 & 0 \\ \varepsilon \gamma^{-1} \widetilde{D}_{11} & 0 & 0 & 0 \end{bmatrix} \widetilde{w}(t) + \begin{bmatrix} D_{12r} \\ \varepsilon \widetilde{B}_2 \\ \varepsilon \widetilde{D}_{22} \\ \varepsilon \widetilde{D}_{12} \end{bmatrix} u(t), \\ y(t) &= C_{2r} x(t) + [\gamma^{-1} D_{21r} \quad 0 \quad \varepsilon^{-1} H_3 \quad 0] \widetilde{w}(t) + D_{22r} u(t), \end{aligned} \tag{10}$$

and

$$\begin{aligned} \dot{x}(t) &= A_r x(t) + [\varepsilon^{-1} H_1 \quad 0] \overline{w}(t) + B_{2r} u(t), \\ \overline{z}(t) &= \begin{bmatrix} \varepsilon \widetilde{A} \\ \varepsilon \widetilde{C}_2 \end{bmatrix} x(t) + \begin{bmatrix} \varepsilon \widetilde{B}_2 \\ \varepsilon \widetilde{D}_{22} \end{bmatrix} u(t), \\ y(t) &= C_{2r} x(t) + [0 \quad \varepsilon^{-1} H_3] \overline{w}(t) + D_{22r} u(t) \end{aligned} \tag{11}$$

where $\varepsilon > 0$ is a scaling parameter, we can show that the following theorems hold.

Theorem 2.3 The system (6) is robustly stabilizable with H_∞ disturbance attenuation with γ via the output feedback controller (8) if the closed-loop system corresponding to (10) and (8) is stable with unitary H_∞ disturbance attenuation.

Proof: The closed-loop system (6) with (8) is given by

$$\begin{aligned} \dot{x}_c(t) &= (A_c + H_{1c} \widetilde{F}_{1c}(t) E_{1c}) x_c(t) + (B_c + H_{1c} \widetilde{F}_{1c}(t) E_{2c}) w(t), \\ z(t) &= (C_c + H_2 \widetilde{F}_{2c}(t) E_{3c}) x_c(t) + (D_{11r} + H_2 \widetilde{F}_{2c}(t) \widetilde{D}_{11}) w(t) \end{aligned}$$

where $x_c = [x^T \quad \hat{x}^T]^T$ and

$$A_c = \begin{bmatrix} A_r & B_{2r}\hat{C} \\ \hat{B}C_{2r} & \hat{A} + \hat{B}D_{22r}\hat{C} \end{bmatrix}, \quad B_c = \begin{bmatrix} B_{1r} \\ \hat{B}D_{21r} \end{bmatrix}, \quad C_c = [C_{1r} \quad D_{12r}\hat{C}], \quad H_{1c} = \begin{bmatrix} H_1 & 0 \\ 0 & \hat{B}H_3 \end{bmatrix},$$

$$\widetilde{F}_{1c}(t) = \begin{bmatrix} \widetilde{F}_1(t) & 0 \\ 0 & \widetilde{F}_3(t) \end{bmatrix}, \quad E_{1c} = \begin{bmatrix} \widetilde{A} & \widetilde{B}_2\hat{C} \\ \widetilde{C}_2 & \widetilde{D}_{22}\hat{C} \end{bmatrix}, \quad E_{2c} = \begin{bmatrix} \widetilde{B}_1 \\ \widetilde{D}_{21} \end{bmatrix}, \quad E_{3c} = [\widetilde{C}_1 \quad \widetilde{D}_{12}\hat{C}].$$

On the other hand, the closed-loop system (10) with (8) is given by

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + [\gamma^{-1}B_c \quad \varepsilon^{-1}H_{1c} \quad 0]\widetilde{w}(t), \\ \widetilde{z}(t) &= \begin{bmatrix} C_c \\ \varepsilon E_{1c} \\ \varepsilon E_{3c} \end{bmatrix} x_c(t) + \begin{bmatrix} \gamma^{-1}D_{11r} & 0 & \varepsilon^{-1}H_2 \\ \varepsilon\gamma^{-1}E_{2c} & 0 & 0 \\ \varepsilon\gamma^{-1}\widetilde{D}_{11r} & 0 & 0 \end{bmatrix} \widetilde{w}(t). \end{aligned}$$

The result follows from Theorem 2.1.

Similar to Theorem 2.3, a robust stabilization is obtained from Theorem 2.2 as follows:

Theorem 2.4 The system (6) with $w(t) = 0$ is robustly stabilizable via the output feedback controller (8) if the closed-loop system corresponding to (11) and (8) is stable with unitary H_∞ disturbance attenuation.

Remark 2.2 Theorem 2.3 shows that a controller that achieves a unitary H_∞ disturbance attenuation for the nominal system (10) can robustly stabilize the uncertain fuzzy system (6). This implies that the same controller can be used to robustly stabilize the fuzzy system (6). Similarly, Theorem 2.4 indicates that a controller that achieves a unitary H_∞ disturbance attenuation γ. This leads to the fact that the same controller can be used to achieve the stability and the prescribed H_∞ disturbance attenuation level of the fuzzy system (6). We also note that research on an H_∞ output feedback controller design has been extensively investigated, and a design method of H_∞ controllers has already been given. Therefore, the existing results on stability with H_∞ disturbance attenuation can be applied to solve the robust H_∞ output feedback stabilization problem for fuzzy systems with the immeasurable premise variables. In addition, if there exist solutions to Theorems 2.3 and 2.4, then, controllers keep certain robustness.

2.4 Numerical examples

We give an illustrative example of designing robust H_∞ output feedback controller for a continuous-time Takagi-Sugeno fuzzy system with immeasurable premise variables. Let us consider the following continuous-time nonlinear system with uncertain parameters.

$$\begin{aligned} \dot{x}_1(t) &= (-3.2 - \alpha)x_1(t) - 0.6x_2(t) + (0.8 + \beta)x_1(t)x_2(t) + 0.3w_1(t), \\ \dot{x}_2(t) &= 0.4x_1(t) - 1.1x_1(t)x_2(t) + 0.2w_1(t) + 0.7u(t), \\ z(t) &= \begin{bmatrix} 0.3x_1(t) + 0.2x_2(t) \\ 0.1u(t) \end{bmatrix}, \\ y(t) &= 0.3x_1(t) - 0.2x_2(t) + w_2(t) \end{aligned}$$

where a and β are uncertain scalars which satisfy $|a| \le 0.1$ and $|\beta| \le 0.3$, respectively. Defining $x(t) = [x_1(t) \quad x_2(t)]$, $w(t) = [w_1(t) \quad w_2(t)]$ and assuming $x_2(t) \in [-1,\ 1]$, we have an equivalent fuzzy system description

$$\begin{aligned}
\dot{x}(t) &= \sum_{i=1}^{2} \lambda_i(x_2(t))\left\{(A_i + H_{1i}F_i(t)E_{1i})x(t) + \begin{bmatrix} 0.3 & 0 \\ 0.2 & 0 \end{bmatrix} w(t) + \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} u(t)\right\}, \\
z(t) &= \begin{bmatrix} 0.3 & 0.2 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(t), \\
y(t) &= [0.3 \quad -0.2]x(t) + [0 \quad 1]w(t)
\end{aligned}$$

where $\lambda_1(x_2(t)) = (-x_2(t)+1)/2$, $\lambda_2(x_2(t)) = (x_2(t)+1)/2$ and

$$A_1 = \begin{bmatrix} -2.4 & -0.6 \\ -1.1 & 0.4 \end{bmatrix}, \quad A_2 = \begin{bmatrix} -4 & -0.6 \\ 1.1 & 0.4 \end{bmatrix}, \quad H_{11} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad H_{12} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$F_1(t) = \frac{a}{|a|}, \quad F_2(t) = \frac{\beta}{|\beta|}, \quad E_{11} = [0.1 \quad 0], \quad E_{12} = [0.1 \quad 0]$$

which can be written as

$$\begin{aligned}
\dot{x}(t) &= (A_2 + H_1\widetilde{F}_1(t)\widetilde{A})x(t) + \begin{bmatrix} 0.3 & 0 \\ 0.2 & 0 \end{bmatrix} w(t) + \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix} u(t), \\
z(t) &= \begin{bmatrix} 0.3 & 0.2 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0.1 \end{bmatrix} u(t), \\
y(t) &= [0.3 \quad -0.2]x(t) + [0 \quad 1]w(t)
\end{aligned}$$

where

$$H_1 = \begin{bmatrix} 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \widetilde{F}_1(t) = \mathrm{diag}[\lambda_1(t) \quad \lambda_1(t) \quad F_1(t) \quad F_2(t)], \quad \widetilde{A} = \begin{bmatrix} 1.6 & 0 \\ -2.2 & 0 \\ 0.1 & 0 \\ 0.1 & 0 \end{bmatrix},$$

It is easy to see that the original system with $u(t) = 0$ is unstable. Theorem 2.3 allows us to design a robust stabilizing controller with H_∞ disturbance attenuation $\gamma = 10$:

$$\begin{aligned}
\dot{\hat{x}}(t) &= \begin{bmatrix} -24.6351 & -4.8920 \\ 8.5631 & -13.3744 \end{bmatrix} \hat{x}(t) + \begin{bmatrix} 48.7026 \\ -32.8950 \end{bmatrix} y(t), \\
u(t) &= [-37.4520 \quad -90.1612]\hat{x}(t)
\end{aligned}$$

This controller is applied to the system. A simulation result with the initial conditions $x(0) = [-0.5 \quad 0.6]^T$, $\hat{x}(0) = [0 \quad 0]^T$, the noises $w_1(t) = w_2(t) = 0.1e^{-0.2t}\sin(0.1t)$ and the assumption $F_1(t) = F_2(t) = \sin(10t)$ is depicted in Figures 1 and 2, which show the trajectories of the state and control, respectively. It is easy to see that the obtained controller stabilizes the nonlinear system.

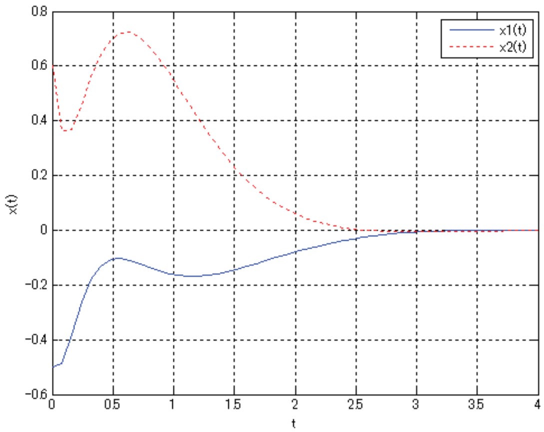


Fig. 1. The state trajectories

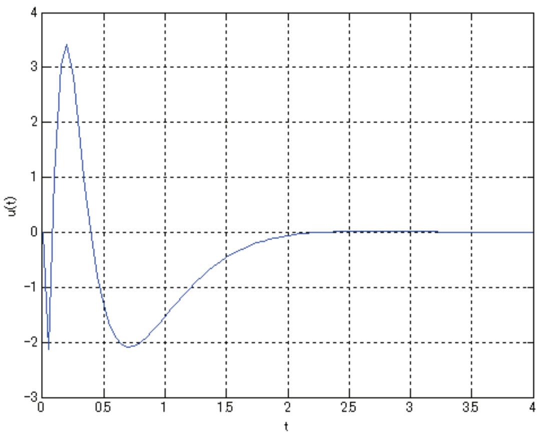


Fig. 2. The control trajectory

3. Robust output feedback control for uncertain discrete-time fuzzy systems with immeasurable premise variables

We now consider the discrete-time counterpart of the previous section. We first show that a discrete-time Takagi-Sugeno fuzzy system with immeasurable premise variables can be written as an uncertain discrete-time linear system. Then, robust stability with H_∞ disturbance attenuation problem for such an uncertain system is converted into stability with H_∞ disturbance attenuation for a nominal system. Based on such a relationship, a solution to a robust stability with H_∞ disturbance attenuation problem via output feedback controller for a discrete-time fuzzy system with immeasurable premise variables is given. Finally, a numerical example illustrates our theory.

3.1 Fuzzy system and problem formulation

In this section, we consider discrete-time Takagi-Sugeno fuzzy systems with immeasurable premise variables. The Takagi-Sugeno fuzzy model is described by the following IF-THEN rules:

IF ξ_1 is M_{i1} and … and ξ_p is M_{ip},

THEN
$$\begin{aligned}
x(k+1) &= (A_i + \Delta A_i)x(k) + (B_{1i} + \Delta B_{1i})w(k) + (B_{2i} + \Delta B_{2i})u(k),\\
z(k) &= (C_{1i} + \Delta C_{1i})x(k) + (D_{11i} + \Delta D_{11i})w(k) + (D_{12i} + \Delta D_{12i})u(k),\\
y(k) &= (C_{2i} + \Delta C_{2i})x(k) + (D_{21i} + \Delta D_{21i})w(k) + (D_{22i} + \Delta D_{22i})u(k), \quad i = 1,\cdots,r
\end{aligned}$$

where $x(k) \in \Re^n$ is the state, $w(k) \in \Re^{m_1}$ is the disturbance, $u(k) \in \Re^{m_2}$ is the control input, $z(k) \in \Re^{q_1}$ is the controlled output, $y(k) \in \Re^{q_2}$ is the measurement output. r is the number of IF-THEN rules. M_{ij} are fuzzy sets and ξ_1, $\cdots$, ξ_p are premise variables. We set $\xi = [\xi_1 \ \cdots \ \xi_p]^T$ and $\xi(k)$ is assumed to be a function of $x(k)$, which is not measurable. We assume that the premise variables do not to depend on $u(k)$, and that A_i, B_{1i}, B_{2i}, C_{1i}, C_{2i}, D_{11i}, D_{12i}, D_{21i} and D_{22i} are constant matrices of appropriate dimensions. The uncertain matrices are of the form

$$\begin{bmatrix} \Delta A_i & \Delta B_{1i} & \Delta B_{2i} \\ \Delta C_{1i} & \Delta D_{11i} & \Delta D_{12i} \\ \Delta C_{2i} & \Delta D_{21i} & \Delta D_{22i} \end{bmatrix} = \begin{bmatrix} H_{1i} \\ H_{2i} \\ H_{3i} \end{bmatrix} F_i(k) \begin{bmatrix} E_{1i} & E_{2i} & E_{3i} \end{bmatrix}, \quad i = 1,\cdots,r$$

where $F_i(k) \in \Re^{l\times j}$ is an uncertain matrix satisfying $F_i^T(k)F_i(k) \le I$, and H_{1i}, H_{2i}, H_{3i}, E_{1i}, E_{2i} and E_{3i} are known constant matrices of appropriate dimensions.

Assumption 3.1 The system (A_r, B_{1r}, B_{2r}, C_{1r}, C_{2r}, D_{11r}, D_{12r}, D_{21r}, D_{22r}) represents a nominal system that can be chosen as a subsystem including the equilibrium point of the original system.

$\xi(k)$ is assumed to be a function of the immeasurable state $x(k)$. The state, the controlled output and the measurement output equations are defined as follows:

$$\begin{aligned}
x(k+1) &= \sum_{i=1}^{r}\lambda_i(\xi(k))\{(A_i + \Delta A_i)x(k) + (B_{1i} + \Delta B_{1i})w(k) + (B_{2i} + \Delta B_{2i})u(k)\},\\
z(k) &= \sum_{i=1}^{r}\lambda_i(\xi(k))\{(C_{1i} + \Delta C_{1i})x(k) + (D_{11i} + \Delta D_{11i})w(k) + (D_{12i} + \Delta D_{12i})u(k)\}, \quad (12)\\
y(k) &= \sum_{i=1}^{r}\lambda_i(\xi(k))\{(C_{2i} + \Delta C_{2i})x(k) + (D_{21i} + \Delta D_{21i})w(k) + (D_{22i} + \Delta D_{22i})u(k)\}
\end{aligned}$$

where

$$\lambda_i(\xi) = \frac{\beta_i(\xi)}{\sum_{i=1}^{r}\beta_i(\xi)}, \quad \beta_i(\xi) = \prod_{j=1}^{p} M_{ij}(\xi_j)$$

and $M_{ij}(\cdot)$ is the grade of the membership function of M_{ij}. We assume

$$\beta_i(\xi(k)) \ge 0, \quad i = 1,\cdots,r, \quad \sum_{i=1}^{r}\beta_i(\xi(k)) > 0$$

for any $\xi(k)$. Hence $\lambda_i(\xi(k))$ satisfies

$$\lambda_i(\xi(k)) \geq 0, \quad i = 1,\cdots,r, \quad \sum_{i=1}^{r} \lambda_i(\xi(k)) = 1 \tag{13}$$

for any $\xi(k)$. Our problem is to find a control $u(\cdot)$ for the system (1) given the output measurements $y(\cdot)$ such that the controlled output $z(\cdot)$ satisfies

$$\sum_{i=0}^{\infty} z^T(i)z(i) < \gamma^2 \sum_{i=1}^{\infty} w^T(i)w(i) \tag{14}$$

for a prescribed scalar $\gamma > 0$. If such a controller exists, we call it an H_∞ output feedback controller. Here, we consider the robust stabilization and the robust H_∞ disturbance attenuation of uncertain fuzzy system (9) with the immeasurable premise variable.

3.2 Robust stability of equivalent uncertain systems

Similar to continuous-time case, we rewrite (12) as an uncertain system. Then, the robust stability with H_∞ disturbance attenuation problem of fuzzy system is converted as the robust stability and robust stability with H_∞ disturbance attenuation problems of equivalent uncertain system. Since $\lambda_i(x(k))$ satisfies (13), we have

$$\lambda_r(x(k)) = 1 - [\lambda_1(x(k)) + \lambda_2(x(k)) + \cdots + \lambda_{r-1}(x(k))].$$

It follows from this relationship that

$$\begin{aligned} \sum_{i=1}^{r} \lambda_i(x(k))(A_i + \Delta A_i) &= A_r + H_{1r}F(k)E_{1r} + \lambda_1(x(k))[A_{1r} + \overline{H}_{1r}\overline{F}_{1r}(k)\overline{E}_{1r}] \\ &\quad + \cdots + \lambda_{r-1}(x(k))[A_{r-1,r} + \overline{H}_{r-1,r}\overline{F}_{r-1,r}(k)\overline{E}_{r-1,r}] \\ &= A_r + H_1\widetilde{F}(k)\widetilde{A} \end{aligned}$$

where

$$\begin{gathered} A_{ir} = A_i - A_r, \quad i = 1,\cdots,r-1 \\ H_1 = [I \quad I \quad \cdots \quad I \quad H_{11} \quad H_{12} \quad \cdots \quad H_{1r}], \\ \tilde{F}(k) = \text{diag } [\lambda_1(k)I \quad \cdots \quad \lambda_{r-1}(k)I \quad \lambda_1(t)F_1(k) \quad \cdots \quad \lambda_r(t)F_r(k)], \\ \widetilde{A} = [A_{1r}^T \quad A_{2r}^T \quad \cdots \quad A_{r-1,r}^T \quad E_{11}^T \quad E_{12}^T \quad \cdots \quad E_{1r}^T]^T. \end{gathered}$$

Similarly, we rewrite $\sum_{i=1}^{r} \lambda_i(x(k))(B_{1i} + \Delta B_{1i})$, $\sum_{i=1}^{r} \lambda_i(x(k))(B_{2i} + \Delta B_{2i})$, $\sum_{i=1}^{r} \lambda_i(x(k))(C_{1i} + \Delta C_{1i})$, $\sum_{i=1}^{r} \lambda_i(x(k))(C_{2i} + \Delta C_{2i})$, $\sum_{i=1}^{r} \lambda_i(x(k))(D_{11i} + \Delta D_{11i})$, $\sum_{i=1}^{r} \lambda_i(x(k))(D_{12i} + \Delta D_{12i})$, $\sum_{i=1}^{r} \lambda_i(x(k))(D_{21i} + \Delta D_{21i})$ and $\sum_{i=1}^{r} \lambda_i(x(k))(D_{22i} + \Delta D_{22i})$, and we have an equivalent description for (12):

$$\begin{aligned} x(k+1) &= (A_r+\Delta A)x(k)+(B_{1r}+\Delta B_1)w(k)+(B_{2r}+\Delta B_2)u(k) \\ &\triangleq A_\Delta x(k)+B_{1\Delta}w(k)+B_{2\Delta}u(k), \\ z(k) &= (C_{1r}+\Delta C_1)x(k)+(D_{11r}+\Delta D_{11})w(k)+(D_{12r}+\Delta D_{12})u(k) \\ &\triangleq C_{1\Delta}x(k)+D_{11\Delta}w(k)+D_{12\Delta}u(k), \\ y(k) &= (C_{2r}+\Delta C_2)x(k)+(D_{21r}+\Delta D_{21})w(k)+(D_{22r}+\Delta D_{22})u(k) \\ &\triangleq C_{2\Delta}x(k)+D_{21\Delta}w(k)+D_{22\Delta}u(k) \end{aligned} \tag{15}$$

where

$$\begin{bmatrix} \Delta A & \Delta B_1 & \Delta B_2 \\ \Delta C_1 & \Delta D_{11} & \Delta D_{12} \\ \Delta C_2 & \Delta D_{21} & \Delta D_{22} \end{bmatrix} = \begin{bmatrix} H_1 & 0 & 0 \\ 0 & H_2 & 0 \\ 0 & 0 & H_3 \end{bmatrix} \begin{bmatrix} \widetilde{F}(k) & 0 & 0 \\ 0 & \widetilde{F}(k) & 0 \\ 0 & 0 & \widetilde{F}(k) \end{bmatrix} \begin{bmatrix} \widetilde{A} & \widetilde{B}_1 & \widetilde{B}_2 \\ \widetilde{C}_1 & \widetilde{D}_{11} & \widetilde{D}_{12} \\ \widetilde{C}_2 & \widetilde{D}_{21} & \widetilde{D}_{22} \end{bmatrix},$$

$$H_2 = [I \quad I \quad \cdots \quad I \quad H_{21} \quad H_{22} \quad \cdots \quad H_{2r}],$$

$$H_3 = [I \quad I \quad \cdots \quad I \quad H_{31} \quad H_{32} \quad \cdots \quad H_{3r}],$$

$$B_{1ir} = B_{1i} - B_{1r}, \quad B_{2ir} = B_{2i} - B_{2r}, \quad C_{1ir} = C_{1i} - C_{1r},$$

$$C_{2ir} = C_{2i} - C_{2r}, \quad D_{11ir} = D_{11i} - D_{11r}, \quad D_{12ir} = D_{12i} - D_{12r},$$

$$D_{21ir} = D_{21i} - D_{21r}, \quad D_{22ir} = D_{22i} - D_{22r}, \quad i = 1,\cdots,r-1,$$

$$\widetilde{B}_1 = [B_{11r}^T \quad B_{12r}^T \quad \cdots \quad B_{1r-1,r}^T \quad E_{21}^T \quad E_{22}^T \quad \cdots \quad E_{2r}^T]^T,$$

$$\widetilde{B}_2 = [B_{21r}^T \quad B_{22r}^T \quad \cdots \quad B_{2r-1,r}^T \quad E_{31}^T \quad E_{32}^T \quad \cdots \quad E_{3r}^T]^T,$$

$$\widetilde{C}_1 = [C_{11r}^T \quad C_{12r}^T \quad \cdots \quad C_{1r-1,r}^T \quad E_{11}^T \quad E_{12}^T \quad \cdots \quad E_{1r}^T]^T,$$

$$\widetilde{C}_2 = [C_{21r}^T \quad C_{22r}^T \quad \cdots \quad C_{2r-1,r}^T \quad E_{11}^T \quad E_{12}^T \quad \cdots \quad E_{1r}^T]^T,$$

$$\widetilde{D}_{11} = [D_{111r}^T \quad D_{112r}^T \quad \cdots \quad D_{11r-1,r}^T \quad E_{21}^T \quad E_{22}^T \quad \cdots \quad E_{2r}^T]^T,$$

$$\widetilde{D}_{12} = [D_{121r}^T \quad D_{122r}^T \quad \cdots \quad D_{12r-1,r}^T \quad E_{31}^T \quad E_{32}^T \quad \cdots \quad E_{3r}^T]^T,$$

$$\widetilde{D}_{21} = [D_{211r}^T \quad D_{212r}^T \quad \cdots \quad D_{21r-1,r}^T \quad E_{21}^T \quad E_{22}^T \quad \cdots \quad E_{2r}^T]^T,$$

$$\widetilde{D}_{22} = [D_{221r}^T \quad D_{222r}^T \quad \cdots \quad D_{22r-1,r}^T \quad E_{31}^T \quad E_{32}^T \quad \cdots \quad E_{3r}^T]^T.$$

We note that since the state $x(k)$ is not measurable, $\lambda_i(x(k))$ are unknown. This implies that $\widetilde{F}(k)$ is a time varying unknown function. However, it is easy to see that $\widetilde{F}(k)$ satisfies $\widetilde{F}^T(k)\widetilde{F}(k) \le I$, because $0 \le \lambda_i(x(k)) \le 1$, $F_i^T(k)F_i(k) \le I$, $i = 1,2,3$. Hence, with Assumption 3.1, we can see (12) as a linear nominal system with time varying uncertainties. The importance on this description is that the system (15) is exactly the same as the system (12). Now, our problem of finding an H∞ controller for the system (12) is to find a robustly stabilizing output feedback controller with H∞ disturbance attenuation of the form (16) for the uncertain system (15).

$$\begin{aligned} \hat{x}(k+1) &= \hat{A}\hat{x}(k)+\hat{B}y(k), \\ u(k) &= \hat{C}\hat{x}(k). \end{aligned} \tag{16}$$

Definition 3.1 (i) Consider the unforced system (12) with $w(k)=0$, $u(k)=0$. The uncertain system (12) is said to be robustly stable if there exists a matrix $X>0$ such that

$$A_\Delta^T X A_\Delta - X < 0$$

for all admissible uncertainties.
(ii) The uncertain system (12) is said to be robustly stabilizable via output feedback controller if there exists an output feedback controller of the form (16) such that the resulting closed-loop system (12) with (16) is robustly stable.
Definition 3.2 (i) Given a scalar $\gamma>0$, the system (12) is said to be robustly stable with H∞ disturbance attenuation γ if there exists a matrix $X>0$ such that

$$\begin{bmatrix} -X & 0 & A_\Delta^T & C_{1\Delta}^T \\ 0 & -\gamma^2 I & B_{1\Delta}^T & D_{11\Delta}^T \\ A_\Delta & B_{1\Delta} & -X^{-1} & 0 \\ C_{1\Delta} & D_{11\Delta} & 0 & -I \end{bmatrix} < 0.$$

(ii) Given a scalar $\gamma>0$, the uncertain system (12) is said to be robustly stabilizable with H∞ disturbance attenuation γ via output feedback controller if there exists an output feedback controller of the form (16) such that the resulting closed-loop system (12) with (16) is robustly stable with H∞ disturbance attenuation γ.
The robust stability and the robust stability with H∞ disturbance attenuation are converted into the stability with H∞ disturbance attenuation.
Definition 3.3 Given a scalar $\gamma>0$, the system

$$\begin{aligned} x(k+1) &= Ax(k)+Bw(k), \\ z(k) &= Cx(k)+Dw(k) \end{aligned} \tag{17}$$

is said to be stable with H∞ disturbance attenuation γ if it is exponentially stable and input-output stable with (14).
Now, we state our key results that show the relationship between the robust stability and the robust stability with H∞ disturbance attenuation of an uncertain system, and stability with H∞ disturbance attenuation of a nominal system.
Theorem 3.1 The system (12) with $w(k)=0$ is robustly stable if and only if for $\varepsilon>0$ the system

$$\begin{aligned} x(k+1) &= A_r x(k)+\varepsilon^{-1}H_1\overline{w}(k), \\ \overline{z}(k) &= \varepsilon\tilde{A}x(k) \end{aligned}$$

where $\overline{w}$ and $\overline{z}$ are of appropriate dimensions, is stable with unitary H∞ disturbance attenuation $\gamma=1$.
Proof: The system (12) is robustly stable if and only if there exists a matrix $X>0$ such that

$$(A_r+H_1F_1(k)\tilde{A})^T X(A_r+H_1F_1(k)\tilde{A})-X<0,$$

which can be written as

$$Q + HF_1(k)E + E^T F_1^T(k)H^T < 0 \tag{18}$$

where

$$Q = \begin{bmatrix} -X & A_r^T \\ A_r & -X^{-1} \end{bmatrix}, \quad H = \begin{bmatrix} 0 \\ H_1 \end{bmatrix}, \quad E = [\tilde{A} \quad 0].$$

It follows from Lemma 2.1 that there exists $X > 0$ such that (18) holds if and only if there exist a matrix $X > 0$ and a scalar $\varepsilon > 0$ such that

$$Q + \frac{1}{\varepsilon^2} HH^T + \varepsilon^2 E^T E < 0,$$

which can be written as

$$Y_1 = \begin{bmatrix} Q & \varepsilon^{-1}H & \varepsilon E^T \\ \varepsilon^{-1}H^T & -I & 0 \\ \varepsilon E & 0 & -I \end{bmatrix} < 0.$$

Pre-multiplying and post-multiplying

$$S_1 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix},$$

we have

$$S_1 Y_1 S_1 = \begin{bmatrix} -X & 0 & A_r^T & \varepsilon\tilde{A}^T \\ 0 & -I & \varepsilon^{-1}H_1^T & 0 \\ A_r & \varepsilon^{-1}H_1 & -X^{-1} & 0 \\ \varepsilon A & 0 & 0 & -I \end{bmatrix} < 0.$$

The desired result follows from Definition 3.1.

Theorem 3.2 The system (12) with $u(k) = 0$ is robustly stable with H_∞ disturbance attenuation γ if and only if for $\varepsilon > 0$ the system

$$\begin{aligned} x(k+1) &= A_r x(k) + [\gamma^{-1}B_{1r} \quad \varepsilon^{-1}H_1 \quad 0]\tilde{w}(k), \\ \tilde{z}(k) &= \begin{bmatrix} C_{1r} \\ \varepsilon\tilde{A} \\ \varepsilon\tilde{C}_1 \end{bmatrix} x(k) + \begin{bmatrix} \gamma^{-1}D_{11r} & 0 & \varepsilon^{-1}H_2 \\ \varepsilon\gamma^{-1}\tilde{B}_1 & 0 & 0 \\ \varepsilon\gamma^{-1}\tilde{D}_{11r} & 0 & 0 \end{bmatrix} \tilde{w}(k) \end{aligned}$$

where $\tilde{w}$ and $\tilde{z}$ are of appropriate dimensions, is stable with unitary H_∞ disturbance attenuation $\gamma = 1$.

Proof: The system (12) is robustly stable with H∞ disturbance attenuation γ if and only if there exists a matrix $X>0$ such that

$$\begin{bmatrix} -X & 0 & (A_r+H_1F_1(k)\widetilde{A})^T & (C_{1r}+H_2F_2(k)\widetilde{C}_1)^T \\ 0 & -\gamma^2 I & (B_{1r}+H_1F_1(k)\widetilde{B}_1)^T & (D_{11r}+H_2F_2(k)\widetilde{D}_{11})^T \\ A_r+H_1F_1(k)\widetilde{A} & B_{1r}+H_1F_1(k)\widetilde{B}_1 & -X^{-1} & 0 \\ C_{1r}+H_2F_2(k)\widetilde{C}_1 & D_{11r}+H_2F_2(k)\widetilde{D}_{11} & 0 & -I \end{bmatrix} < 0,$$

which can be written as

$$\hat{Q}+\hat{H}\hat{F}(k)\hat{E}+\hat{E}^T\hat{F}^T(k)\hat{H}^T < 0 \tag{19}$$

where

$$\hat{Q}=\begin{bmatrix} -X & 0 & A_r^T & C_{1r}^T \\ 0 & -\gamma^2 I & B_{1r}^T & D_{11r}^T \\ A_r & B_{1r} & -X^{-1} & 0 \\ C_{1r} & D_{11r} & 0 & -I \end{bmatrix},\ \hat{H}=\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ H_1 & 0 \\ 0 & H_2 \end{bmatrix},\ \hat{F}(k)=\begin{bmatrix} F_1(k) & 0 \\ 0 & F_2(k) \end{bmatrix},\ \hat{E}=\begin{bmatrix} \widetilde{A} & \widetilde{B}_1 & 0 & 0 \\ \widetilde{C}_1 & \widetilde{D}_{11} & 0 & 0 \end{bmatrix}.$$

It can be shown from Lemma 2.1 that there exists $X>0$ such that (19) holds if and only if there exist $X>0$ and a scalar $\varepsilon>0$ such that

$$\hat{Q}+\frac{1}{\varepsilon^2}\hat{H}\hat{H}^T+\varepsilon^2\hat{E}^T\hat{E}<0,$$

which can be written as

$$Y_2=\begin{bmatrix} \hat{Q} & \varepsilon^{-1}\hat{H} & \varepsilon\hat{E}^T \\ \varepsilon^{-1}\hat{H}^T & -I & 0 \\ \varepsilon\hat{E} & 0 & -I \end{bmatrix}<0.$$

Pre-multiplying and post-multiplying the above LMI by

$$S_2=\begin{bmatrix} I & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma^{-1}I & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & I & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & I \end{bmatrix},$$

we have

$$S_2Y_2S_2 = \begin{bmatrix} -X & 0 & A_r^T & \widetilde{C}^T \\ 0 & -I & \widetilde{B}^T & \widetilde{D}^T \\ A_r & \widetilde{B} & -X^{-1} & 0 \\ \widetilde{C} & \widetilde{D} & 0 & -I \end{bmatrix} < 0.$$

The result follows from Definition 3.1.

3.3 Robust controller design

We are now at the position where we propose the control design of an H_∞ output feedback controller for the system (12). The controller design is based on the equivalent system (15). The design of a robustly stabilizing output feedback controller with H_∞ disturbance attenuation for the system (15) can be converted into that of a stabilizing controller with H_∞ disturbance attenuation controllers for a nominal system. For the following auxiliary systems, we can show that the following theorems hold. Consider the following systems:

$$\begin{aligned} x(k+1) &= A_r x(k) + [\gamma^{-1}B_{1r} \quad \varepsilon^{-1}H_1 \quad 0 \quad 0]\widetilde{w}(k) + B_{2r}u(k), \\ \widetilde{z}(k) &= \begin{bmatrix} C_{1r} \\ \varepsilon\widetilde{A} \\ \varepsilon\widetilde{C}_2 \\ \varepsilon\widetilde{C}_1 \end{bmatrix} x(k) + \begin{bmatrix} \gamma^{-1}D_{11r} & 0 & 0 & \varepsilon^{-1}H_2 \\ \varepsilon\gamma^{-1}\widetilde{B}_1 & 0 & 0 & 0 \\ \varepsilon\gamma^{-1}\widetilde{D}_{21} & 0 & 0 & 0 \\ \varepsilon\gamma^{-1}\widetilde{D}_{11} & 0 & 0 & 0 \end{bmatrix} \widetilde{w}(k) + \begin{bmatrix} D_{12r} \\ \varepsilon\widetilde{B}_2 \\ \varepsilon\widetilde{D}_{22} \\ \varepsilon\widetilde{D}_{12} \end{bmatrix} u(k), \\ y(k) &= C_{2r}x(k) + [\gamma^{-1}D_{21r} \quad 0 \quad \varepsilon^{-1}H_3 \quad 0]\widetilde{w}(k) + D_{22r}u(k) \end{aligned} \tag{20}$$

and

$$\begin{aligned} x(k+1) &= A_r x(k) + [\varepsilon^{-1}H_1 \quad 0]\overline{w}(k) + B_{2r}u(k), \\ \overline{z}(k) &= \begin{bmatrix} \varepsilon\widetilde{A} \\ \varepsilon\widetilde{C}_2 \end{bmatrix} x(k) + \begin{bmatrix} \varepsilon\widetilde{B}_2 \\ \varepsilon\widetilde{D}_{22} \end{bmatrix} u(k), \\ y(k) &= C_{2r}x(k) + [0 \quad \varepsilon^{-1}H_3]\overline{w}(k) + D_{22r}u(k), \end{aligned} \tag{21}$$

where $\varepsilon > 0$ is a scaling parameter.

Theorem 3.3 The system (12) is robustly stabilizable with H_∞ disturbance attenuation with γ via the output feedback controller (16) if the closed-loop system corresponding to (20) and (16) is stable with unitary H_∞ disturbance attenuation.

Proof: The closed-loop system (12) with (16) is given by

$$\begin{aligned} x_c(k+1) &= (A_c + H_{1c}F_{1c}(k)E_{1c})x_c(k) + (B_c + H_{1c}F_{1c}(k)E_{2c})w(k), \\ z(k) &= (C_c + H_2F_2(k)E_{3c})x(k) + (D_{11r} + H_2F_2(k)\widetilde{D}_{11})w(k). \end{aligned}$$

where $x_c = [x^T \quad \hat{x}^T]^T$ and

$$A_c = \begin{bmatrix} A_r & B_{2r}\hat{C} \\ \hat{B}C_{2r} & \hat{A} + \hat{B}D_{22r}\hat{C} \end{bmatrix}, \quad B_c = \begin{bmatrix} B_{1r} \\ \hat{B}D_{21r} \end{bmatrix}, \quad C_c = [C_{1r} \quad D_{12r}\hat{C}], \quad H_{1c} = \begin{bmatrix} H_1 & 0 \\ 0 & \hat{B}H_3 \end{bmatrix},$$

$$F_{1c}(k)=\begin{bmatrix}F_1(k) & 0\\ 0 & F_3(k)\end{bmatrix},\quad E_{1c}=\begin{bmatrix}\tilde{A} & \tilde{B}_2\hat{C}\\ \tilde{C}_2 & \tilde{D}_{22}\hat{C}\end{bmatrix},\quad E_{2c}=\begin{bmatrix}\tilde{B}_1\\ \tilde{D}_{21}\end{bmatrix},\quad E_{3c}=[\tilde{C}_1 \quad \tilde{D}_{12}\hat{C}].$$

On the other hand, the closed-loop system (20) with (17) is given by

$$\begin{aligned} x_c(k+1) &= A_c x_c(k)+[\gamma^{-1}B_c \quad \varepsilon^{-1}H_{1c} \quad 0]\tilde{w}(k),\\ \tilde{z}(k) &= \begin{bmatrix}C_c\\ \varepsilon E_{1c}\\ \varepsilon E_{3c}\end{bmatrix}x_c(k)+\begin{bmatrix}\gamma^{-1}D_{11r} & 0 & \varepsilon^{-1}H_2\\ \varepsilon\gamma^{-1}E_{2c} & 0 & 0\\ \varepsilon\gamma^{-1}\tilde{D}_{11r} & 0 & 0\end{bmatrix}\tilde{w}(k). \end{aligned}$$

The result follows from Theorem 3.2.
Similar to Theorem 3.3, a robust stabilization is obtained from Theorem 3.2 as follows:

Theorem 3.4 The system (12) with $w(k)=0$ is robustly stabilizable via the output feedback controller (16) if the closed-loop system corresponding to (21) and (16) is stable with unitary H_∞ disturbance attenuation.

Remark 3.1 Theorem 3.3 indicates that a controller that achieves a unitary H_∞ disturbance attenuation for the nominal system (20) can robustly stabilize the fuzzy system (12) with H_∞ disturbance attenuation γ. Similar argument can be made on robust stabilization of Theorem 3.4. Therefore, the existing results on stability with H_∞ disturbance attenuation can be applied to solve our main problems.

3.4 Numerical examples

Now, we illustrate a control design of a simple discrete-time Takagi-Sugeno fuzzy system with immeasurable premise variables. We consider the following nonlinear system with uncertain parameters.

$$\begin{aligned} x_1(k+1) &= (0.9+a)x_1(k)-0.2x_2(k)+0.2x_1(k)x_2^2(k)+0.3w_1(k),\\ x_2(k+1) &= 0.2x_1(k)-(0.4+\beta)x_2^3(k)+0.5w_1(k)+0.7u(k),\\ z(k) &= \begin{bmatrix}1.5x_1(k)+0.5x_2(k)\\ u(k)\end{bmatrix},\\ y(k) &= 0.3x_1(k)-0.1x_2(k)+w_2(k) \end{aligned}$$

where a and β are uncertain scalars which satisfy $|a|\le 0.1$ and $|\beta|\le 0.02$, respectively. Defining $x(k)=[x_1(k) \quad x_2(k)]$, $w(k)=[w_1(k) \quad w_2(k)]$ and assuming $x_2(k)\in[-1,\ 1]$, we have an equivalent fuzzy system description

$$\begin{aligned} x_1(k+1) &= \sum_{i=1}^{2}\lambda_i(x_2(k))\left\{(A_i+H_{1i}F_i(k)E_{1i})x(k)+\begin{bmatrix}0.3 & 0\\ 0.5 & 0\end{bmatrix}w(k)+\begin{bmatrix}0\\ 0.7\end{bmatrix}u(k)\right\},\\ z(k) &= \begin{bmatrix}1.5 & 0.5\\ 0 & 0\end{bmatrix}x(k)+\begin{bmatrix}0\\ 1\end{bmatrix}u(k),\\ y(k) &= [0.3 \quad -0.1]x(k)+[0 \quad 1]w(k) \end{aligned}$$

where $\lambda_1(x_2(k))=1-x_2^2(k)$, $\lambda_2(x_2(k))=x_2^2(k)$ and

$$A_1 = \begin{bmatrix} 0.9 & -0.2 \\ 0.2 & 0 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1.1 & -0.2 \\ 0.2 & -0.3 \end{bmatrix}, \quad H_{11} = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \quad H_{12} = \begin{bmatrix} 0 \\ 0.2 \end{bmatrix},$$

$$F_1(k) = \frac{a}{|a|}, \quad F_2(k) = \frac{\beta}{|\beta|}, \quad E_{11} = [0.2 \quad 0], \quad E_{12} = [0 \quad 0.1],$$

which can be written as

$$\begin{aligned} x_1(k+1) &= (A_2 + H_1\widetilde{F}(k)\widetilde{A})x(k) + \begin{bmatrix} 0.3 & 0 \\ 0.5 & 0 \end{bmatrix} w(k) + \begin{bmatrix} 0 \\ 0.7 \end{bmatrix} u(k), \\ z(k) &= \begin{bmatrix} 1.5 & 0.5 \\ 0 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k), \\ y(k) &= [0.3 \quad -0.1]x(k) + [0 \quad 1]w(k) \end{aligned}$$

where

$$H_1 = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.2 \end{bmatrix}, \quad \widetilde{F}(k) = \mathrm{diag}[\lambda_1(k) \quad \lambda_1(k) \quad F_1(k) \quad F_2(k)], \quad \widetilde{A} = \begin{bmatrix} -0.2 & 0 \\ 0 & 0.3 \\ 0.2 & 0 \\ 0 & 0.1 \end{bmatrix}.$$

The open-loop system is originally unstable. Theorem 3.3 allows us to design a robust stabilizing controller with H_∞ disturbance attenuation $\gamma = 20$:

$$\begin{aligned} \hat{x}(k+1) &= \begin{bmatrix} 0.0971 & -0.0063 \\ 2.3181 & -0.1348 \end{bmatrix} \hat{x}(k) + \begin{bmatrix} 3.8463 \\ 7.9861 \end{bmatrix} y(k), \\ u(k) &= [3.0029 \quad -0.1726]\hat{x}(k) \end{aligned}$$

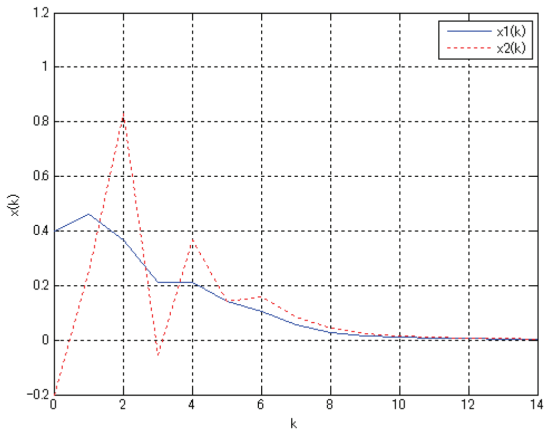


Fig. 3. The state trajectories

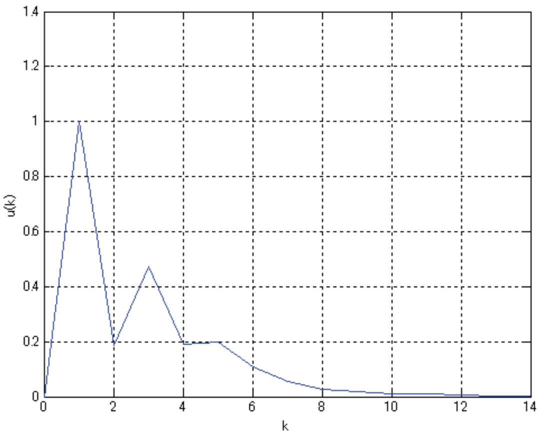


Fig. 4. The control trajectory

This controller is applied to the system. A simulation result with the initial conditions $x(0) = [0.4 \quad -0.3]^T$, $\hat{x}(0) = [0 \quad 0]^T$, the noises $w_1(k) = e^{-k}\cos(k)$, $w_2(k) = e^{-k}\sin(k)$ and the assumption $F_1(k) = F_2(k) = \sin(k)$ is depicted in Figures 3 and 4, which show the trajectories of the state and control, respectively. We easily see that the obtained controller stabilizes the system.

4. Extension to fuzzy time-delay systems

In this section, we consider an extension to robust control problems for Takagi-Sugeno fuzzy time-delay systems. Consider the Takagi-Sugeno fuzzy model, described by the following IF-THEN rule:

IF ξ_1 is M_{i1} and … and ξ_p is M_{ip},

THEN
$$\begin{aligned}
\dot{x}(t) &= (A_i + \Delta A_i)x(t) + (A_{di} + \Delta A_{di})x(t-h) + (B_{1i} + \Delta B_{1i})w(t) + (B_{2i} + \Delta B_{2i})u(t),\\
z(t) &= (C_{1i} + \Delta C_{1i})x(t) + (C_{1di} + \Delta C_{1di})x(t-h) + (D_{11i} + \Delta D_{11i})w(t)\\
&\quad + (D_{12i} + \Delta D_{12i})u(t),\\
y(t) &= (C_{2i} + \Delta C_{2i})x(t) + (C_{2di} + \Delta C_{2di})x(t-h) + (D_{21i} + \Delta D_{21i})w(t)\\
&\quad + (D_{22i} + \Delta D_{22i})u(t), \qquad i = 1,\cdots,r
\end{aligned}$$

where $x(t) \in \Re^n$ is the state, $w(t) \in \Re^{m_1}$ is the disturbance, $u(t) \in \Re^{m_2}$ is the control input, $z(t) \in \Re^{q_1}$ is the controlled output, $y(t) \in \Re^{q_2}$ is the measurement output. r is the number of IF-THEN rules. M_{ij} is a fuzzy set and $\xi_1, \cdots, \xi_p$ are premise variables. We set $\xi = [\xi_1 \ \cdots \ \xi_p]^T$. We assume that the premise variables do not to depend on $u(t)$. A_i, A_{di}, B_{1i}, B_{2i}, C_{1i}, C_{2i}, C_{1di}, C_{2di}, D_{11i}, D_{12i}, D_{21i} and D_{22i} are constant matrices of appropriate dimensions. The uncertain matrices are of the form (1) with $\Delta A_{di} = H_{1i}F_i(t)E_{di}$, $\Delta C_{1di} = H_{2i}F_i(t)E_{di}$ and $\Delta C_{2di} = H_{3i}F_i(t)E_{di}$ where H_{1i}, H_{2i}, H_{3i} and E_{di} are known constant matrices of appropriate dimensions.

Assumption 4.1 The system (A_r, A_{dr}, B_{1r}, B_{2r}, C_{1r}, C_{2r}, C_{1dr}, C_{2dr}, D_{11r}, D_{12r}, D_{21r}, D_{22r}) represents a nominal system that can be chosen as a subsystem including the equilibrium point of the original system.
The state equation, the controlled output and the output equation are defined as follows:

$$\begin{aligned}
\dot{x}(t) &= \sum_{i=1}^{r}\lambda_i(x(t))\{(A_i+\Delta A_i)x(t)+(A_{di}+\Delta A_{di})x(t-h)+(B_{1i}+\Delta B_{1i})w(t) \\
&\quad +(B_{2i}+\Delta B_{2i})u(t)\}, \\
z(t) &= \sum_{i=1}^{r}\lambda_i(x(t))\{(C_{1i}+\Delta C_{1i})x(t)+(C_{1di}+\Delta C_{1di})x(t-h)+(D_{11i}+\Delta D_{11i})w(t) \\
&\quad +(D_{12i}+\Delta D_{12i})u(t)\}, \\
y(t) &= \sum_{i=1}^{r}\lambda_i(x(t))\{(C_{2i}+\Delta C_{2i})x(t)+(C_{2di}+\Delta C_{2di})x(t-h)+(D_{21i}+\Delta D_{21i})w(t) \\
&\quad +(D_{22i}+\Delta D_{22i})u(t)\}
\end{aligned} \tag{22}$$

where $\lambda_i(x(t))$ is defined in (3) and satisfies (4). Our problem is to find a control $u(\cdot)$ for the system (22) given the output measurements $y(\cdot)$ such that the controlled output $z(\cdot)$ satisfies (5) for a prescribed scalar $\gamma > 0$. Using the same technique as in the previous sections, we have an equivalent description for (22):

$$\begin{aligned}
\dot{x}(t) &= (A_r+\Delta A)x(t)+(A_{dr}+\Delta A_d)x(t-h)+(B_{1r}+\Delta B_1)w(t)+(B_{2r}+\Delta B_2)u(t) \\
&\triangleq A_\Delta x(t)+A_{d\Delta}x(t-h)+B_{1\Delta}w(t)+B_{2\Delta}u(t), \\
z(t) &= (C_{1r}+\Delta C_1)x(t)+(C_{1dr}+\Delta C_{1d})x(t-h)+(D_{11r}+\Delta D_{11})w(t)+(D_{12r}+\Delta D_{12})u(t) \\
&\triangleq C_{1\Delta}x(t)+C_{1d\Delta}x(t-h)+D_{11\Delta}w(t)+D_{12\Delta}u(t), \\
y(t) &= (C_{2r}+\Delta C_2)x(t)+(C_{2dr}+\Delta C_{2d})x(t-h)+(D_{21r}+\Delta D_{21})w(t)+(D_{22r}+\Delta D_{22})u(t) \\
&\triangleq C_{2\Delta}x(t)+C_{2d\Delta}x(t-h)+D_{21\Delta}w(t)+D_{22\Delta}u(t)
\end{aligned} \tag{23}$$

where $\Delta A_d = H_1\widetilde{F}(t)\widetilde{A}_d$, $\Delta C_{1d} = H_1\widetilde{F}(t)\widetilde{C}_{1d}$, $\Delta C_{2d} = H_1\widetilde{F}(t)\widetilde{C}_{2d}$, and other uncertain matrices are given in (7). As we can see from (23) that uncertain Takagi-Sugeno fuzzy time-delay system (22) can be written as an uncertain linear time-delay system. Thus, robust control problems for uncertain fuzzy time-delay system (22) can be converted into those for an uncertain linear time-delay system (23). Solutions to various control problems for an uncertain linear time-delay system have been given(for example, see Gu et al., 2003; Mahmoud, 2000) and hence the existing results can be applied to solve robust control problems for fuzzy time-delay systems.

5. Conclusion

This chapter has considered robust H_∞ control problems for uncertain Takagi-Sugeno fuzzy systems with immeasurable premise variables. A continuous-time Takagi-Sugeno fuzzy system was first considered. Takagi-Sugeno fuzzy system with immeasurable premise variables can be written as an uncertain linear system. Based on such an uncertain system representation, robust stabilization and robust H_∞ output feedback controller design method was proposed. The same control problems for discrete-time counterpart were also considered. For both continuous-time and discrete-time control problems, numerical examples were shown to illustrate our design methods. Finally, an extension to fuzzy time-delay systems was given and a way to robust control problems for them was shown. Uncertain system approach taken in this chapter is applicable to filtering problems where the state variable is assumed to be immeasurable.

6. References

Assawinchaichote, W.; Nguang, S.K. & Shi, P. (2006). *Fuzzy Control and Filter Design for Uncertain Fuzzy Systems*, Springer

Boyd, S.; El Ghaoui, L.; Feron, E. & Balakrishnan, V. (1994). *Linear Matrix Inequalities in Systems and Control Theory*, SIAM

Cao, G.; Rees, W. & Feng, G. (1996). H_∞ control of nonlinear continuous-time systems based on dynamical fuzzy models, *International Journal of Systems Science* 27, pp.821-830

Chen, C.L.; Feng, G.; Sun, D. & Guan, X.P. (2005). H_∞ output feedback control of discrete-time fuzzy systems with application to chaos control, *IEEE Transactions on Fuzzy Systems* 13, pp.531-543

Feng, G.; Cao, G. & Rees, W. (1996). An approach to H_∞ control of a class of nonlinear systems, *Automatica* 32, pp.1469-1474

Gu, K; Kharitonov, V. and Chen, J. (2003). *Stability of Time-Delay Systems*, Boston: Birkhauser

Guerra, T.M.; Kruszewski, A.; Vermeiren, L. & Tiemant, H. (2006). Conditions of output stabilization for nonlinear models in the Takagi-Sugeno's form, *Fuzzy Sets and Systems* 157, pp.1248-1259

Hong, S.-K. & Langari, R. (1998). Synthesis of an LMI-based fuzzy control system with guaranteed optimal H_∞ performance, *Proc. of IEEE International Conference on Fuzzy Systems*, pp.422-427

Katayama, H. & Ichikawa, A. (2002). H_∞ control for discrete-time Takagi-Sugeno fuzzy systems, *International Journal of Systems Science* 33, pp.1099-1107

Ma, X.J.; Sun, Z.Q. & He, Y.Y (1998). Analysis and design of fuzzy controller and fuzzy observer, *IEEE Transactions on Fuzzy Systems* 6, pp.41-50

Mahmoud, M.S. (2000). *Robust Control and Filtering for Time-Delay Systems*, New York-Basel: Mercel-Dekker, Inc.

Takagi, T. & Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on System, Man, Cybernetics* 15, pp.116-132

Tanaka, K. & Sugeno, M. (1992). Stability analysis and design of fuzzy control systems, *Fuzzy Sets and Systems* 45, pp.135-156

Tanaka, K. & Sano, M. (1994). On the concepts of regulator and observer of fuzzy control systems, *Proc. of IEEE International Conference on Fuzzy Systems*, pp.767-772

Tanaka, K.; Ikeda, T. & Wang, H.O. (1996). Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: quadratic stabilizability, H_∞ control theory, and linear ,atrix inequalities, *IEEE Transactions on Fuzzy Systems* 4, pp.1-13

Tanaka, K.; Ikeda, T. & Wang, H.O. (1998). Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs, *IEEE Transactions on Fuzzy Systems* 6, pp.1-16

Tanaka, K. & Wang, H.O. (1997). Fuzzy regulators and fuzzy observers: a linear matrix inequality approach, *Proc. of IEEE Conference on Decision and Control*, pp.1315-1320

Tanaka, K. & Wang, H.O. (2001). *Fuzzy Control Systems Design and Analysis: A Linear Matrix Inequality Approach*, Wiley-Interscience

Xie, L. (1996). Output feedback H_∞ control of systems with parameter uncertainty, *International Journal of Control* 63, pp.741-759

Yoneyama, J.; Nishikawa, M.; Katayama, H. & Ichikawa, A. (2000). Output stabilization of Takagi-Sugeno fuzzy systems, *Fuzzy Sets and Systems* 111, pp.253-266

Yoneyama, J.; Nishikawa, M.; Katayama, H. & Ichikawa, A. (2001a). Design of output feedback controllers for Takagi-Sugeno fuzzy systems, *Fuzzy Sets and Systems* 121, pp.127-148

Yoneyama, J.; Nishikawa, M.; Katayama, H. & Ichikawa, A. (2001b). H_∞-Control for Takagi-Sugeno fuzzy systems, *International Journal of Systems* 32, pp.915-924

Yoneyama, J. (2008a). Robust H_∞ output feedback control for uncertain fuzzy systems with immeasurable premise variables, *Advances in Fuzzy Sets and Systems* 3, pp.99-113

Yoneyama, J. (2008b). H_∞ output feedback control for fuzzy systems with immeasurable premise variables: discrete-time case, *Applied Soft Computing* 8, pp.949-958

3

Control of T-S Fuzzy Systems Using Fuzzy Weighting-Dependent Lyapunov Function

Sung Hyun Kim and PooGyeon Park

Division of Electrical and Computer Engineering, Pohang University of Science and Technology, Pohang, Kyungbuk, 790-784, Korea

1. Introduction

Over the past two decades, there has been a rapidly growing interest in approximating a nonlinear system by a Takagi-Sugeno (T-S) fuzzy model (Takagi & Sugeno, 1985). In general, this model is represented by using a set of fuzzy rules to describe a global nonlinear system in terms of a set of local linear models which are smoothly connected by fuzzy membership functions. Based on the T-S fuzzy model, recently, various fuzzy controllers have been developed under the so-called parallel-distributed compensation (PDC) scheme (in which each control rule is distributively designed for the corresponding rule of a T-S fuzzy model) and have been widely and successfully applied in fields ranging from aerospace to process control. The reason is because the fuzzy model-based control method provides a natural, simple and effective design approach to complement other nonlinear control techniques that require special and rather involved knowledge. Below are listed the main features of T-S model-based fuzzy control method:

1. It does not require severe structural assumptions on the plant model.
2. It preserves well-understood linear intuition.
3. It is naturally compatible with decompositions of the overall control problem. The decompositions are typically not hierarchical, and the interactions of sub-problems are captured by physical variables that are typically state variables in a more complete model of the overall system.
4. It enables control systems to respond rapidly to changing operating conditions. For this reason, it is important that the selected physical variables reflect changes in plant dynamics as operating conditions change.

In fact, the T-S model-based fuzzy control method (of divide and conquer type) constructs a nonlinear controller, with certain required dynamic properties, by combining, in some sense, the members of appropriate family of linear time-invariant controllers. Here, nonlinear control design task is broken into a number of linear sub-problems, which enable linear design methods to be applied to nonlinear problems. Within the general framework of the T-S fuzzy model-based control method, a flurry of research activities have quickly yielded many important results on the design of fuzzy control systems by means of the following Lyapunov function approaches:

1. Common quadratic Lyapunov function approach (Tanaka & Sugeno, 1992; Tanaka et al, 1996; Wang et al, 1996; Cao & Frank, 2000; Assawinchaichote, 2004).

2. Piecewise Lyapunov function approach (Cao et al, 1996; Cao et al, 1997; Cao et al, 1999; Han et al, 2000; Feng, 2003; Feng, 2004; Chen et al, 2005).
3. Fuzzy weighting-dependent Lyapunov function approach (Tanaka et al, 2001; Park & Choi, 2001; Choi & Park, 2003; Kim & Park 2008; Kim et al, 2009).

The basic idea of these approaches is to design a feedback controller for each local model and to construct a global controller from the local controller in such a way that global stability of the closed-loop fuzzy control system is guaranteed. In the context of these approaches, various studies have attempted to tackle the robust control problem (Tanaka et al, 1996; Chen et al, 1999; Tsai & Li, 2009), performance-oriented control problem (Chen et al, 2000; Xiaodong & Qingling, 2003; Zhou et al, 2005), networked control problem (Hwang & Chang, 2008; Jiang & Han, 2008; Gao et al, 2009), and delayed system control problem (Cao & Frank, 2000; Chen et al, 2005; Wu, 2008).

A. T-S fuzzy model and control synthesis

In general, it is not possible to exactly reformulate a nonlinear system as a T-S fuzzy system. However, it is possible to over-bound the nonlinear system in the sense that every solution to the nonlinear system is a solution to the T-S fuzzy system (but not *vice versa*). Thus, rather conservative results are expected in the procedure of modeling the T-S fuzzy system. Moreover, since the T-S fuzzy model is not unique, there may be a potential in reducing the conservatism occurring when approximating the nonlinear system. Hence, one always needs to discuss how to non-conservatively construct the T-S fuzzy model for the given nonlinear system. As a result, various methods for the reformulation of nonlinear systems into T-S fuzzy systems have been presented in the literature (Tanaka & Wang, 2001; and references therein). In this chapter, we would like to introduce a geometric method for some particular nonlinearities: sector nonlinearity, saturation nonlinearity, and fault nonlinearity (see Section 2).

Meanwhile, based on the aforementioned Lyapunov function approaches, numerous investigations and researches have been carried out to develop the T-S model-based fuzzy control system. Here, it should be noted that recent research efforts have focused on using the PLF or FWLF approach when establishing a feedback control law since the CQLF approach leads to over-conservative design solutions for a large number of T-S fuzzy subsystems. Thus, we shall also focus on taking advantage of the FWLF approach to derive less conservative conditions for the solvability of the stabilization problem (for lack of space, the PLF approach will be not discussed in this chapter).

B. Main issues

Most stabilization conditions based on the FWLF are formulated in terms of parameterized linear matrix inequalities (PLMIs), which causes the following primary practical difficulty: the PLMI-based condition involves an infinite number of LMI-based conditions and thus the task of establishing a controller is intractable numerically. This arises because the PLMI-based condition must be satisfied for every allowable parameter value that leads to uncountably many conditions since there is a continuum of parameter values. To overcome this difficulty, Becker et al (1993) proposed an approximate, *ad hoc* approach whereby the parameter space is divided into a fine grid, and a controller is designed so that the solvability conditions are satisfied at a finite number of parameter values. However, it should be noted that there appears to be little guidance as to how perform the gridding.

Moreover, for a particular grid spacing, the number of grid points required grows extremely rapidly as the number of parameters increases. Hence, despite the relative efficiency of the available numerical algorithms for solving linear matrix inequalities (LMIs), the utility of this approach with present computing facilities is strictly limited to systems with a small number of parameters (less than three or four).

To deal with this problem, we shall select an appropriate structure for variables, *say* $\mathcal{X}(\theta(t))$, of the PLMI-based condition under consideration in such a way that the variables are polynomially dependent on parameters denoting fuzzy weighting functions, *say* $\theta_i(t)$:

$$\mathcal{X}(\theta(t)) = \sum_{i_1=1}^{r}\sum_{i_2=1}^{r}\cdots\sum_{i_g=1}^{r}\theta_{i_1}(t)\theta_{i_2}(t)\cdots\theta_{i_g}(t)\mathcal{X}_{i_1 i_2\cdots i_g} \tag{1}$$

subject to $\sum_{i=1}^{r}\theta_i(t) = 1$ and $\theta_i(t) \geq 0$, $i = 1,\cdots,r$. In particular, for simplicity of presentation, we shall take the following two cases into consideration in this chapter:

Affine Parameter Dependence (APD):

$$\mathcal{X}(\theta(t)) = \sum_{i=1}^{r}\theta_i(t)\mathcal{X}_i,\ s.t.\ \sum_{i=1}^{r}\theta_i(t) = 1, \theta_i(t) \geq 0, i = 1,\cdots,r, \tag{2}$$

Quadratic Parameter Dependence (QPD):

$$\mathcal{X}(\theta(t)) = \sum_{i=1}^{r}\sum_{j=1}^{r}\theta_i(t)\theta_j(t)\mathcal{X}_{ij},\ s.t.\ \sum_{i=1}^{r}\theta_i(t) = 1, \theta_i(t) \geq 0, i = 1,\cdots,r, \tag{3}$$

In fact, the use of (1) yields a polynomially parameter-dependent condition such as

$$0 < \sum_{i_1=1}^{r}\sum_{i_2=1}^{r}\cdots\sum_{i_p=1}^{r}\theta_{i_1}(t)\theta_{i_2}(t)\cdots\theta_{i_p}(t)\mathcal{L}_{i_1 i_2\cdots i_p} \tag{4}$$

subject to $\sum_{i=1}^{r}\theta_i(t) = 1$ and $\theta_i(t) \geq 0$, $i = 1,\cdots,r$. Thus, the condition in (4) naturally reduces to a feasibility problem with a finite number of conditions as follows:

$$0 < \mathcal{L}_{i_1 i_2\cdots i_p}, \forall i_1, i_2, \cdots, i_p \in [1, r]. \tag{5}$$

However, it is very conservative to use (5) to numerically solve the feasible problem of (4): Thus, we shall propose an efficient relaxation technique to reduce the conservatism caused by the use of (5) (see Section 3), which may achieve better system performances compared with those of other techniques appeared in the literature (Tanaka et al, 1998; Kim & Lee, 2000; Xiaodong & Qingling, 2003; Tuan et al, 2001; Teixeira, 2003; Sala & Ariño , 2007; Fang et al, 2006).

C. Organization

This chapter is organized as follows: Section 2 gives the information on the T-S fuzzy system description and its modeling. Further, Section 3 illustrates about the parameterized linear matrix inequality (PLMI) and introduces our main relaxation technique in detail. Based on CQLFs and FWLFs, Section 4 gives the LMI-based stabilization conditions, derived using the proposed relaxation technique in Section 3, for a class of T-S fuzzy systems.

Notation and symbols

We collect here, for ease of reference, a list of the main notation and symbols that represent the same meaning throughout the chapters.

$\mathcal{R}^n$	denotes the n-dimensional real space.
$\lVert x \rVert = (x^T x)^{1/2}$	is taken to be the standard Euclidian norm.
$\mathcal{L}_{2+} = \mathcal{L}_2[0,\infty)$	denotes the Lebesgue space consisting of square-integrable functions on $[0,\infty)$.
diag(A,B)	denotes a diagonal matrix with diagonal entries A and B.
$A \oplus B$	stands for the Kronecker sum of two matrices A and B, which is the same as diag(A,B)
(*)	is used, in symmetric block matrices, as an ellipsis for terms that are induced by symmetry.
$X \geq Y$	mean that $X - Y$ is positive semi-definite, respectively.
$X > Y$	mean that $X - Y$ is positive definite, respectively.
Tr(Q)	returns the sum of the diagonal elements of the matrix Q.

2. T-S fuzzy system description and modeling

T-S fuzzy systems have recently received much attention in the engineering field, such as chemical processes, robotics systems, automatic systems, aerospace or vehicle systems, and manufacturing processes, owing to their ability to represent the nonlinear system and their systematic means of computing feedback controllers.

A. T-S fuzzy system description

The ith rules of the T-S fuzzy models are of the following forms:

Model Rule i : IF $\eta_1(t)$ is $\mathcal{F}_{i1}$ and … and $\eta_s(t)$ is $\mathcal{F}_{is}$, THEN

$$\begin{cases} \nabla x(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases}, \; for \; i = 1,2,\cdots,r, \tag{6}$$

where the consequent subsystems in (6) represent linear systems in local operating regions; $\mathcal{F}_{ij}$ denotes a fuzzy set; $\eta_1(t),\cdots,\eta_s(t)$ denote the premise variables of the model; r denotes the number of IF - THEN rules; $x(t) \in \mathcal{R}^{n_x}$, $u(t) \in \mathcal{R}^{n_u}$, $y(t) \in \mathcal{R}^{n_y}$ denote the state, the input, the measured output, respectively; and ∇ represents the derivative operator for continuous-time and the forward operator for discrete-time systems. Here, it is assumed that the premise variables not not explicitly depend on the control input $u(t)$. This assumption is needed to avoid a complicated defuzzification process of fuzzy controller, under which the overall fuzzy model is inferred as

$$\begin{cases} \nabla x(t) = A(\theta(t))x(t) + B(\theta(t))u(t) \\ y(t) = C(\theta(t))x(t) \end{cases} \tag{7}$$

where

$$\begin{bmatrix} A(\theta(t)) & B(\theta(t)) \\ C(\theta(t)) & 0 \end{bmatrix} \overset{\Delta}{=} \sum_{i=1}^{r} \theta_i(t) \begin{bmatrix} A_i & B_i \\ C_i & 0 \end{bmatrix}, \tag{8}$$

$$\theta_i(t) = \theta_i(\eta(t)) = \frac{g_i(\eta(t))}{\sum_{i=1}^{r} g_i(\eta(t))}, \; g_i(\eta(t)) = \prod_{j=1}^{s} f_{ij}(\eta_j(t)) \tag{9}$$

in which A_i, B_i, and C_i are real constant matrices with appropriate dimensions, the notation $\eta(t) = [\, \eta_1(t), \cdots, \eta_s(t) \,]^T \in \mathcal{R}^s$, and $f_{ij}(\eta_j(t))$ denotes the grade of membership of $\eta_j(t)$ in $\mathcal{F}_{ij}$, and $\theta(t) = [\, \theta_1(t), \cdots, \theta_r(t) \,] \in \mathcal{R}^r$. Moreover, let $g_i(\eta(t)) \geq 0$, for $i = 1, \cdots, r$, and $\sum_{i=1}^{r} g_i(\eta(t)) > 0$ for all time t. Then, we can claim that $\theta_i(\eta(t)) \geq 0$, for $i = 1, \cdots, r$, and $\sum_{i=1}^{r} \theta_i(\eta(t)) = 1$ for all time t.

As shown in (8), the T-S fuzzy system is defined as linear systems whose dynamics depend on time-varying parameters $\theta_i(t)$ referred to as the scheduling or weight sequence. Further, it is worth pointing out that the parameters $\theta_i(t)$ are generally subject to the following constraints:

Continuous-Time Case:

$$\bar{\theta}_{\min} \leq \bar{\theta}(t) \left(\overset{\Delta}{=} \sum_{i=1}^{r} \theta_i(t) \right) \leq \bar{\theta}_{\max}, \tag{10}$$

$$\alpha_i \leq \theta_i(t) \leq \beta_i, \text{for } i = 1, \cdots, r, \tag{11}$$

$$\mu_i \leq \dot{\theta}_i(t) \leq \nu_i, \text{for } i = 1, \cdots, r. \tag{12}$$

Discrete-Time Case:

$$\bar{\theta}_{\min} \leq \bar{\theta}(t) \left(\overset{\Delta}{=} \sum_{i=1}^{r} \theta_i(t) \right) \leq \bar{\theta}_{\max}, \tag{13}$$

$$\alpha_i \leq \theta_i \leq \beta_i, \text{for } i = 1, \cdots, r, \tag{14}$$

$$| \theta_i(t) - \theta_i(t-1) | \leq \delta_i, \text{for } i = 1, \cdots, r. \tag{15}$$

Remark 1 In general, the parameter $\theta_i(t)$ is a function of time t, states $x(t)$, and inputs $u(t)$. Except for the actuator nonlinearity (see Section 2-B), the parameter $\theta_i(t)$ is mostly associated with the state of the system in (7).

B. T-S fuzzy model construction

Nonlinear dynamic models for mechanical systems can be readily obtain by, for example, the Lagrange method and the Newton-Euler method. In such cases, we can represent the

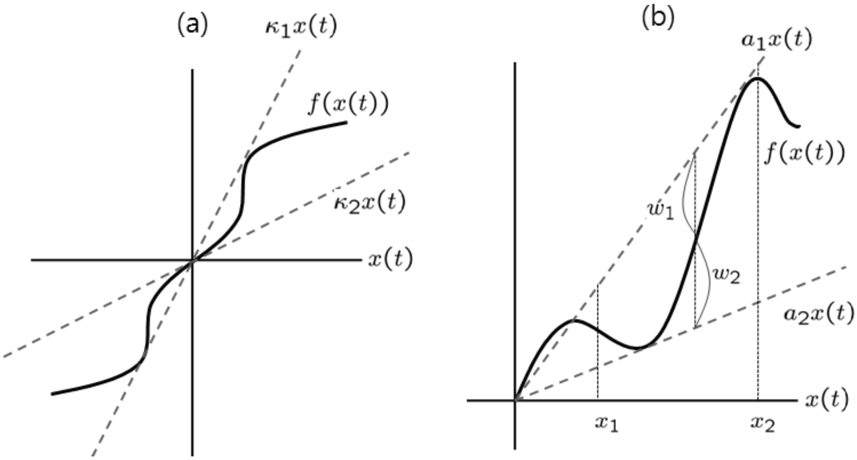


Fig. 1. (a) Global sector nonlinearity; and (b) local sector nonlinearity.

given nonlinear dynamical models as T-S fuzzy systems by using the idea of "sector nonlinearity", "saturation nonlinearity", or a combination of them. Prior to modeling an T-S fuzzy system, we need to simplify the original nonlinear model as much as possible. The procedure is important for practical applications since it always leads to the reduction of the number of the parameters $\theta_i(t)$, which plays an important role in reducing the effort for analysis and design of control systems.

Sector Nonlinearity:

Consider a simple nonlinear system $\nabla x(t) = f(x(t))$, where $f(0) = 0$. The goal of the sector nonlinearity approach is to find the global sector such that $\nabla x(t) = f(x(t)) \in [\kappa_1\ \kappa_2]x(t)$ (see Fig. 1- (a)). This approach guarantees an exact T-S fuzzy model construction. However, note that it is sometimes difficult to find global sectors for general nonlinear systems. Thus, we consider local sector nonlinearity. This is reasonable as variables of physical systems are always bounded. Fig.1- (b) shows the local sector nonlinearity, where two lines become the local sectors under $x(t) \in [t_1, t_2]$. The T-S fuzzy model exactly represents the nonlinear system in the "local" region, that is, $x(t) \in [t_1, t_2]$, which is described as follows:

$$\nabla x(t) = f(x(t)) = \sum_{i=1}^{2} \theta_i(t) a_i x(t), \tag{16}$$

where

$$\theta_1(t) = \frac{w_2}{w_1 + w_2} = \frac{f(x(t)) - a_2 x(t)}{a_1 x(t) - a_2 x(t)}, \tag{17}$$

$$\theta_2(t) = \frac{w_1}{w_1 + w_2} = \frac{a_1 x(t) - f(x(t))}{a_1 x(t) - a_2 x(t)}. \tag{18}$$

In addition, we can claim that $\theta_1(t) + \theta_2(t) = 1.$, $\theta_1(t) \geq 0$, and $\theta_2(t) \geq 0$.

Saturation Nonlinearity:

Saturation nonlinearity is usually caused by limits on component size, properties of materials, and available power. Most actuators present saturation characteristics. For example, the output torque of a two-phase servomotor cannot increase infinitely and tends to saturate, due to the properties of the magnetic material. Similarly, valve-controlled hydraulic servomotors are saturated by the maximum flow rate. To address the saturation problem, three methods are exploited: the circle method (which basically deals with saturation as a sector-bounded nonlinearity), the so-called linear analysis method which consists in determining a region in which a linear controller does not saturate, and the polytopic representation method proposed by Hu and Lin (2001). This chapter will introduce the third method.

Consider the following saturation function:

$$f(u(t)) = sat(u(t),\bar{u}), \tag{19}$$

where the notation $sat(u,\bar{u})$ means

$$sat(u,\bar{u}) = [s_1 \cdots s_{n_u}]^T, s_i = sign(u_i)\min\{\bar{u}_i, |u_i|\}, \tag{20}$$

in which $\bar{u} \in \mathcal{R}^{n_u}$ denotes the saturation level, *sign* returns the signs of the corresponding argument, and u_i and $\bar{u}_i$ denote the *i*-th element of $u \in \mathcal{R}^{n_u}$ and $\bar{u} \in \mathcal{R}^{n_u}$, respectively. From the following lemma (Hu & Lin, 2001), we can obtain a T-S fuzzy model for the saturation nonlinearity:

Lemma 1 Let $\mathcal{G}$ be the set of $n_u \times n_u$ diagonal matrices whose diagonal elements are 1 or 0. Suppose that $|v_i| \le \bar{u}_i$ for all $i = 1,\cdots,n_u$, where v_i and $\bar{u}_i$ denote the *i*-th element of $v \in \mathcal{R}^{n_u}$ and $\bar{u} \in \mathcal{R}^{n_u}$, respectively. Then

$$sat(u(t),\bar{u}) = \sum_{i=1}^{2^{n_u}} \theta_i(t)\left(G_i u(t) + \bar{G}_i v(t)\right), \sum_{i=1}^{2^{n_u}} \theta_i(t) = 1, \theta_i(t) \ge 0, \tag{21}$$

where G_i denote all elements of $\mathcal{G}$, $\bar{G}_i = I - G_i$. ■

In particular, for the case of $n_u = 1$, the parameters are given as

$$\theta_1(t) = \frac{sat(u(t),\bar{u}) - (G_2 u(t) + \bar{G}_2 v(t))}{(G_1 - G_2)u(t) + (\bar{G}_1 - \bar{G}_2)v(t)}, \theta_2(t) = 1 - \theta_1(t). \tag{22}$$

Fault-Related Nonlinearity:

The actuator fault can be modeled as follows:

$$f(u(t)) = u^F(t) = \Lambda(t)u(t), \tag{23}$$

where $\Lambda(t) = \text{diag } \{\lambda_1(t), \lambda_2(t), \cdots, \lambda_{n_u}$ with $\underline{\lambda}_i \le \lambda_i(t) \le \bar{\lambda}_i \le 1$, $i \in [1, n_u]$. Obviously, when $0 \le \underline{\lambda}_i < \bar{\lambda}_i \le 1$, it corresponds to the case of partial fault of the *i*-th actuator. When $\underline{\lambda}_i = \bar{\lambda}_i = 1$, it implies that there is no fault in the *i*-th actuator. Define that the matrix set

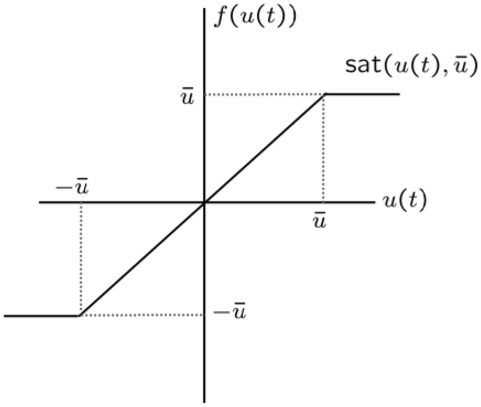


Fig. 2. Saturation nonlinearity.

$$\mathcal{S} \overset{\Delta}{=} \left\{ \Lambda_i \mid \Lambda_i = diag\{\lambda_{i1}, \lambda_{i2}, \cdots, \lambda_{in_u}\}, \lambda_{ij} = \bar{\lambda}_i \; or \; \underline{\lambda}_i, j \in [1, n_u] \right\},$$

where $i = 1, 2, \cdots, 2^{n_u - l}$ and $l(\leq n_u)$ is the number of the actuators without faults. Then, it is obvious that

$$f(u(t)) = \sum_{i=1}^{2^{n_u - l}} \theta_i(t) \Lambda_i u(t).$$

Here, let us assume that $u^F(t)$ is measurable. Then, in the case of $n_u - \ell = 1$, the parameters can be calculated as follows:

$$\theta_1(t) = \frac{u^F(t) - \Lambda_2 u(t)}{(\Lambda_1 - \Lambda_2) u(t)}, \theta_2(t) = \frac{\Lambda_1(t) - u^F(t)}{(\Lambda_1 - \Lambda_2) u(t)}. \tag{24}$$

Remark 2: Since the actuator fault often act as the source of instability in many control systems, the study of reliable control has recently received a considerable amount of attention in control engineering. In particular, three different approaches to address the actuator fault problem have appeared in the literature from the 1980s: pole region assignment (Zhao & Jiang, 1998), algebraic Riccati equation approach (Yang et al, 2001), and linear matrix inequality (LMI) approach (Liao et al, 2002; Wu & Zhang, 2006), which achieve various reliability goals for linear systems.

3. PLMI description and relaxation technique

Solving the PLMIs is solving an infinite number of LMIs and is an extremely difficult problem. To overcome it, we shall take all the fuzzy weighting-dependent variables to be of polynomially parameter dependent structure, and then we shall propose an efficient relaxation technique (Kim & Park, 2008; and Kim et al, 2009) that can replace the PLMIs into a finite number of LMIs.

A. Parameterized Linear Matrix Inequality (PLMI)

The LMI technique is well-known as a unifying framework for formulating and solving problems in control theory. The main advantage of this technique is that complicated control problem can be solved very efficiently with interior point methods (Nesterov & Nemirovski, 1994). A simple feasibility problem in semi-definite programming (SDP) is to find a solution to the following LMI:

$$F(x) = F_0 + \sum_{i=1}^{s} x_i F_i < 0, \tag{25}$$

where the x_i's are the decision variables and the F_i's are given real symmetric matrices. A more complicated generalization of (25) has the following form:

$$F(x(\theta)) = F_0(\theta) + \sum_{i=1}^{s} x_i(\theta) F_i(\theta) < 0, \tag{26}$$

where $\theta \in \mathcal{R}^r$ is an additional parameter allowed to take any value in a compact set Γ (the compact Γ is typically polytopic). One calls (26) parameterized linear matrix inequality (PLMI) to stress the connection with the LMI control theory literature. The goal of (26) is to find $x_i(\theta)$ such that (26) holds for any admissible value of θ, but it is very difficult to numerically solve the PLMI in (26) because of the following:

1. It is infinite-dimensional since the $x_i(\theta)$ are obtained in the infinite- dimensional space of the functions of θ.
2. This is an infinitely constrained LMI problem for which each constraint corresponds to a given point in the range of θ.

Thus, to overcome the difficulties arising from dimensionality, one needs a systematic technique that can turn an PLMI problem into a standard LMI problem. Motivated by the concern, we shall also deal with the problem by selecting an appropriate structure for the parameter-dependent variable $\mathcal{X}(\theta)$ to find a finite number of solvable LMIs from the PLMI (refer to Section 1).

B. Relaxation technique

In the case of adopting the structure in (1) for the analysis and synthesis of T-S fuzzy systems, the relaxation technique plays an important role in finding a less conservative set of solutions since the stability and stabilization conditions are generally of the following structure:

$$\sum_{i_1=1}^{r} \cdots \sum_{i_p=1}^{r} \theta_{i_1}(t) \cdots \theta_{i_p}(t) \mathcal{L}_{i_1 \cdots i_p} > 0, \sum_{i=1}^{r} \theta_i(t) = 1, \theta_i(t) \geq 0, i = 1, \cdots, r. \tag{27}$$

Without loss of generality, the following two statements are equivalent in the case where p=1:

$$i)\sum_{i=1}^{r}\theta_i(t)\mathcal{L}_i > 0,\ subject\ to\ \sum_{i=1}^{r}\theta_i(t)=1,\ \theta_i(t)\geq 0, \tag{28}$$

$$ii)\ \mathcal{L}_i > 0, \forall i = 1,\cdots,r. \tag{29}$$

However, in the case where p=2:

$$\sum_{i_1=1}^{r}\sum_{i_2=1}^{r}\theta_{i_1}(t)\theta_{i_2}(t)\mathcal{L}_{i_1 i_2} > 0,\ subject\ to\ \sum_{i_1=1}^{r}\theta_{i_1}(t)=1,\ \theta_{i_1}(t)\geq 0, \tag{30}$$

we cannot conclude that (30) is equivalent to

$$\mathcal{L}_{i_1 i_2} > 0, \forall i_1, i_2 = 1,\cdots,r. \tag{31}$$

Of course, if (31) holds, then (30) also holds, but (31) leads to very conservative results with respect to (30). Thus, to reduce the conservatism caused by (31), various relaxation schemes have appeared in the literature (Tanaka et al, 1998; Kim & Lee, 2000; Xiaodong & Qingling, 2003; Tuan et al, 2001; Teixeira, 2003; Sala & Ariño , 2007; Fang et al, 2006).

In this section, we shall introduces an useful relaxation technique for Cases 1 to 3, which is made by incorporating some additional constraints on parameters into the interactions among the T-S fuzzy subsystems.

Henceforth, for a simple description, we use the following notations: $\theta_i = \theta_i(t)$, $\dot{\theta}_i = \dot{\theta}_i(t)$, $\theta_i^- = \theta_i(t-1)$, and $\theta_i^+ = \theta_i(t+1)$.

Case 1 (Continuous-time):

Consider the following codition with quadratic dependence for $\theta_i(t)$:

$$0 < \mathcal{L}(\theta(t)) \overset{\Delta}{=} \mathcal{L}_0 + \sum_{i=1}^{r}\theta_i\left(\mathcal{L}_i + \mathcal{L}_i^T\right) + \sum_{i=1}^{r}\theta_i^2\mathcal{L}_{ii} + \sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i\theta_j\mathcal{L}_{ij} + \sum_{j=i+1}^{r}\theta_i\theta_j\mathcal{L}_{ij}^T\right) \tag{33}$$

subject to, for $i, j \in [1, r]$, $j \neq i$,

$$\text{(C1)}\ \sum_{i=1}^{r}\theta_i = 1,\ \text{(C2)}\ 0 \leq \theta_i \leq \beta_i,\ \text{(C3)}\ 0 \leq \theta_i\theta_j. \tag{34}$$

By the S-procedure (Boyd et al, 1994) and Finsler's lemma (de Oliveira & Skelton, 2001; Fang et al, 2004), the condition in Case 1 can be written as follows:

$$0 < \mathcal{L}(\theta(t)) - \mathcal{N}(\theta(t)), \tag{35}$$

where $0 \leq \mathcal{N}(\theta(t))$ is given by

$$\mathcal{N}(\theta(t)) = \mathcal{C}_1 + \mathcal{C}_1^T + \sum_{i=1}^{r}\mathcal{C}_{2i}(\Lambda_i + \Lambda_i^T) + \sum_{i=1}^{r}\sum_{j=1, j\neq i}^{r}\mathcal{C}_{3ij}\left(\Xi_{ij} + \Xi_{ij}^T\right), \tag{36}$$

in which $\mathcal{C}_1$, $\mathcal{C}_{2i}$, and $\mathcal{C}_{3ij}$ are from (C1), (C2), and (C3), respectively:

$$0 = \mathcal{C}_1 \triangleq \begin{bmatrix} I \\ \theta_1 I \\ \vdots \\ \theta_r I \end{bmatrix}^T \begin{bmatrix} I \\ -I \\ \vdots \\ -I \end{bmatrix} \begin{bmatrix} S_0 & S_1 & \cdots & S_r \end{bmatrix} \begin{bmatrix} I \\ \theta_1 I \\ \vdots \\ \theta_r I \end{bmatrix}, \tag{37}$$

$$0 \le \mathcal{C}_{2i} \triangleq -\theta_i^2 + \beta_i \theta_i, \, 0 \le \mathcal{C}_{3ij} \triangleq \theta_i \theta_j. \tag{38}$$

Here, note that the relaxation variables S_0, S_i, Λ_i, and Ξ_{ij} are in $\mathcal{R}^{n_c \times n_c}$ and should satisfy that $0 < \Lambda_i + \Lambda_i^T$ and $0 < \Xi_{ij} + \Xi_{ij}^T$. Further, with some algebraic manipulations, the constraint, $0 \le \mathcal{N}(\theta(t))$, can be rewritten as follows:

$$0 \le \mathcal{N}(\theta(t)) = N_0 + \sum_{i=1}^{r} \theta_i \left(N_i + N_i^T \right) + \sum_{i=1}^{r} \theta_i^2 N_{ii} + \sum_{i=1}^{r} \left(\sum_{j=1}^{i-1} \theta_i \theta_j N_{ij} + \sum_{j=i+1}^{r} \theta_i \theta_j N_{ij}^T \right), \tag{39}$$

where $N_0 = S_0 + S_0^T$, $N_i = \beta_i \Lambda_i - S_0 + S_i$, $N_{ii} = -\left(\Lambda_i + \Lambda_i^T \right) - \left(S_i + S_i^T \right)$, $N_{ij} = -\left(S_i + S_j \right) + \left(\Xi_{ij} + \Xi_{ji} \right)$. Hence, the condition in (35) becomes, for all $\ell, m \in [1, r]$,

$$0 < \Gamma_0 + \sum_{i=1}^{r} \theta_i \left(\Gamma_i + \Gamma_i^T \right) + \sum_{i=1}^{r} \theta_i^2 \Delta_i + \sum_{i=1}^{r} \left(\sum_{j=1}^{i-1} \theta_i \theta_j \Phi_{ij} + \sum_{j=i+1}^{r} \theta_i \theta_j \Phi_{ij}^T \right), \tag{40}$$

where

$$\begin{cases} \Gamma_0 = \mathcal{L}_0 - N_0 = \mathcal{L}_0 - S_0 - S_0^T \\ \Gamma_i = \mathcal{L}_i - N_i = \mathcal{L}_i - \beta_i \Lambda_i + S_0 - S_i \\ \Delta_i = \mathcal{L}_{ii} - N_{ii} = \mathcal{L}_{ii} + \left(\Lambda_i + \Lambda_i^T \right) + \left(S_i + S_i^T \right) \\ \Phi_{ij} = \mathcal{L}_{ij} - N_{ij} = \mathcal{L}_{ij} + \left(S_i + S_j \right) - \left(\Xi_{ij} + \Xi_{ji} \right) \end{cases}. \tag{41}$$

As a result, the condition in (40) boils down to

$$0 < \begin{bmatrix} I & \theta_1 I & \cdots & \theta_r I \end{bmatrix} \tilde{\mathcal{L}} \begin{bmatrix} I & \theta_1 I & \cdots & \theta_r I \end{bmatrix}^T, \tag{42}$$

where

$$\tilde{\mathcal{L}} \overset{\Delta}{=} \left[\begin{array}{c|cccc} \Gamma_0 & (*) & (*) & \cdots & (*) \\ \hline \Gamma_1 & \Delta_1 & (*) & \cdots & (*) \\ \Gamma_2 & \Phi_{21} & \Delta_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & (*) \\ \Gamma_r & \Phi_{r1} & \cdots & \Phi_{r(r-1)} & \Delta_r \end{array} \right]. \tag{43}$$

Proposition 1 The condition in (33) subject to (C1) and (C2) holds if the following conditions hold: for all $i, j \in [1, r]$, $j \neq i$,

$$0 < \tilde{\mathcal{L}}, 0 < \Lambda_i + \Lambda_i^T, 0 < \Xi_{ij} + \Xi_{ij}^T. \tag{44}$$

Case 2 (Continuous-time):
Consider the following condition with quadratic dependence $\theta_i(t)$ and $\dot{\theta}_i(t)$:

$$0 < \mathcal{L}(\theta(t), \dot{\theta}(t)) \triangleq \mathcal{L}_0 + \sum_{i=1}^{r} \theta_i \left(\mathcal{L}_i + \mathcal{L}_i^T \right) + \sum_{i=1}^{r} \dot{\theta}_i \left(\breve{\mathcal{L}}_i + \breve{\mathcal{L}}_i^T \right) + \sum_{i=1}^{r} \theta_i^2 \mathcal{L}_{ii}$$

$$+ \sum_{i=1}^{r} \dot{\theta}_i^2 \breve{\mathcal{L}}_{ii} + \sum_{i=1}^{r} \sum_{j=1}^{r} \dot{\theta}_i \theta_j \left(\bar{\mathcal{L}}_{ij} + \bar{\mathcal{L}}_{ij}^T \right) + \sum_{i=1}^{r} \left(\sum_{j=1}^{i-1} \theta_i \theta_j \mathcal{L}_{ij} + \sum_{j=i+1}^{r} \theta_i \theta_j \mathcal{L}_{ij}^T \right)$$

$$+ \sum_{i=1}^{r} \left(\sum_{j=1}^{i-1} \dot{\theta}_i \dot{\theta}_j \breve{\mathcal{L}}_{ij} + \sum_{j=i+1}^{r} \dot{\theta}_i \dot{\theta}_j \breve{\mathcal{L}}_{ij}^T \right) \tag{45}$$

subject to, for $i, j \in [1, r]$, $j \neq i$,

$$\text{(C1)} \sum_{i=1}^{r} \theta_i = 1, \text{(C2)}\, 0 \le \theta_i \le \beta_i, \text{(C3)}\, 0 \le \theta_i \theta_j, \text{(C4)}\, \rho_i \le \dot{\theta}_i \le \nu_i. \tag{46}$$

By the S-procedure (Boyd et al, 1994) and Finsler's lemma (de Oliveira & Skelton, 2001; Fang et al, 2004), the condition in Case 2 can be written as follows:

$$0 < \mathcal{L}(\theta(t), \dot{\theta}(t)) - \mathcal{N}(\theta(t), \dot{\theta}(t)), \tag{47}$$

where $0 \le \mathcal{N}(\theta(t), \dot{\theta}(t))$ is given by

$$\mathcal{N}(\theta(t), \dot{\theta}(t)) = \mathcal{C}_1 + \mathcal{C}_1^T + \sum_{i=1}^{r} \mathcal{C}_{2i} (\Lambda_i + \Lambda_i^T) + \sum_{i=1}^{r} \sum_{j=1, j \neq i}^{r} \mathcal{C}_{3ij} \left(\Xi_{ij} + \Xi_{ij}^T \right) + \sum_{i=1}^{r} \mathcal{C}_{4i} \left(Z_i + Z_i^T \right), \tag{48}$$

in which $\mathcal{C}_1$, $\mathcal{C}_{2i}$, $\mathcal{C}_{3ij}$, and $\mathcal{C}_{4i}$ are from (C1), (C2), (C3), and (C4), respectively:

$$0 = \mathcal{C}_1 \triangleq \begin{bmatrix} I \\ \theta_1 I \\ \vdots \\ \theta_r I \end{bmatrix}^T \begin{bmatrix} I \\ -I \\ \vdots \\ -I \end{bmatrix} \begin{bmatrix} S_0 & S_1 & \cdots & S_r \end{bmatrix} \begin{bmatrix} I \\ \theta_1 I \\ \vdots \\ \theta_r I \end{bmatrix}, \tag{49}$$

$$0 \le \mathcal{C}_{2i} \triangleq -\theta_i^2 + \beta_i \theta_i, 0 \le \mathcal{C}_{3ij} \triangleq \theta_i \theta_j, 0 \le \mathcal{C}_{4i} \triangleq -\dot{\theta}_i^2 + (\rho_i + \nu_i) \dot{\theta}_i - \rho_i \nu_i. \tag{50}$$

Here, note that the multiplier variables S_0, S_i, Λ_i, Z_i and Ξ_{ij} are in $\mathcal{R}^{n_c \times n_c}$ and should satisfy that $0 < \Lambda_i + \Lambda_i^T$, $0 < Z_i + Z_i^T$ and $0 < \Xi_{ij} + \Xi_{ij}^T$. Further, with some algebraic manipulations, the constraint $0 \le \mathcal{N}(\theta(t))$ can be represented as follows:

$$0 \le \mathcal{N}(\theta(t),\dot{\theta}(t)) = N_0 + \sum_{i=1}^{r}\theta_i\left(N_i + N_i^T\right) + \sum_{i=1}^{r}\dot{\theta}_i\left(\breve{N}_i + \breve{N}_i^T\right) + \sum_{i=1}^{r}\theta_i^2 N_{ii}$$

$$+\sum_{i=1}^{r}\dot{\theta}_i^2 \breve{N}_{ii} + \sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i\theta_j N_{ij} + \sum_{j=i+1}^{r}\theta_i\theta_j N_{ij}^T\right),$$

where $N_0 = S_0 + S_0^T - \sum_{i=1}^{r}\rho_i\nu_i\left(Z_i + Z_i^T\right)$, $N_i = \beta_i\Lambda_i - S_0 + S_i$, $\breve{N}_i = (\rho_i + \nu_i)Z_i$, $N_{ii} = -\left(\Lambda_i + \Lambda_i^T\right) - \left(S_i + S_i^T\right)$, $\breve{N}_{ii} = -\left(Z_i + Z_i^T\right)$, and $N_{ij} = -\left(S_i + S_j\right) + \left(\Xi_{ij} + \Xi_{ji}\right)$. Hence, the condition in (47) becomes, for all $\ell, m \in [1, r]$,

$$0 < \Gamma_0 + \sum_{i=1}^{r}\theta_i\left(\Gamma_i + \Gamma_i^T\right) + \sum_{i=1}^{r}\dot{\theta}_i\left(\breve{\Gamma}_i + \breve{\Gamma}_i^T\right) + \sum_{i=1}^{r}\theta_i^2\Delta_i$$

$$+\sum_{i=1}^{r}\dot{\theta}_i^2\breve{\Delta}_i + \sum_{i=1}^{r}\sum_{j=1}^{r}\dot{\theta}_i\theta_j\left(\Pi_{ij} + \Pi_{ij}^T\right) + \sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i\theta_j\Phi_{ij} + \sum_{j=i+1}^{r}\theta_i\theta_j\Phi_{ij}^T\right)$$

$$+\sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\dot{\theta}_i\dot{\theta}_j\breve{\Phi}_{ij} + \sum_{j=i+1}^{r}\dot{\theta}_i\dot{\theta}_j\breve{\Phi}_{ij}^T\right), \tag{51}$$

where

$$\begin{cases}\Gamma_0 = \mathcal{L}_0 - N_0 = \mathcal{L}_0 - S_0 - S_0^T + \sum_{i=1}^{r}\rho_i\nu_i\left(Z_i + Z_i^T\right)\\ \Gamma_i = \mathcal{L}_i - N_i = \mathcal{L}_i - \beta_i\Lambda_i + S_0 - S_i\\ \breve{\Gamma}_i = \breve{\mathcal{L}}_i - \breve{N}_i = \breve{\mathcal{L}}_i - (\rho_i + \nu_i)Z_i\\ \Delta_i = \mathcal{L}_{ii} - N_{ii} = \mathcal{L}_{ii} + \left(\Lambda_i + \Lambda_i^T\right) + \left(S_i + S_i^T\right)\\ \breve{\Delta}_i = \breve{\mathcal{L}}_{ii} - \breve{N}_{ii} = \breve{\mathcal{L}}_{ii} + Z_i + Z_i^T\\ \Phi_{ij} = \mathcal{L}_{ij} - N_{ij} = \mathcal{L}_{ij} + \left(S_i + S_j\right) - \left(\Xi_{ij} + \Xi_{ji}\right)\\ \breve{\Phi}_{ij} = \breve{\mathcal{L}}_{ij}\\ \Pi_{ij} = \bar{\mathcal{L}}_{ij}\end{cases}. \tag{52}$$

As a result, the condition (51) boils down to

$$0 < \left[I \quad \theta_1 I \quad \cdots \quad \theta_r I \quad \dot{\theta}_1 I \quad \cdots \quad \dot{\theta}_r I\right]\tilde{\mathcal{L}}\left[I \quad \theta_1 I \quad \cdots \quad \theta_r I \quad \dot{\theta}_1 I \quad \cdots \quad \dot{\theta}_r I\right]^T,$$

where

$$\tilde{\mathcal{L}} \triangleq \left[\begin{array}{c|cccc|cccc} \Gamma_0 & (*) & (*) & \cdots & (*) & (*) & (*) & (*) & (*) \\ \hline \Gamma_1 & \Delta_1 & (*) & \cdots & (*) & (*) & \cdots & & (*) \\ \Gamma_2 & \Phi_{21} & \Delta_2 & \ddots & \vdots & (*) & \cdots & & (*) \\ \vdots & \vdots & \ddots & \ddots & (*) & \vdots & & & \vdots \\ \Gamma_r & \Phi_{r1} & \cdots & \Phi_{r(r-1)} & \Delta_r & (*) & \cdots & & (*) \\ \hline \breve{\Gamma}_1 & \Pi_{11} & \cdots & & \Pi_{1r} & \breve{\Delta}_1 & (*) & \cdots & (*) \\ \breve{\Gamma}_2 & \Pi_{21} & \cdots & & \Pi_{2r} & \breve{\Phi}_{21} & \breve{\Delta}_2 & \ddots & \vdots \\ \vdots & \vdots & & & \vdots & \vdots & \ddots & \ddots & (*) \\ \breve{\Gamma}_r & \Pi_{r1} & \cdots & & \Pi_{rr} & \breve{\Phi}_{r1} & \cdots & \breve{\Phi}_{r(r-1)} & \breve{\Delta}_r \end{array}\right]. \quad (53)$$

Proposition 2 The condition in (45) subject to (C1) - (C4) holds if the following conditions hold: for all $i, j \in [1, r]$, $j \neq i$,

$$0 < \tilde{\mathcal{L}}, 0 < \Lambda_i + \Lambda_i^T, 0 < \Xi_{ij} + \Xi_{ij}^T, 0 < Z_i + Z_i^T.$$

Case 3 (Discrete-time):
Consider the following condition with quadratic dependence $\theta_i(t)$ and $\dot{\theta}_i(t-1)$:

$$0 < \mathcal{L}(\theta(t-1), \theta(t)) \triangleq \mathcal{L}_0 + \sum_{i=1}^{r} \theta_i \left(\mathcal{L}_i + \mathcal{L}_i^T \right) + \sum_{i=1}^{r} \theta_i^- \left(\breve{\mathcal{L}}_i + \breve{\mathcal{L}}_i^T \right) + \sum_{i=1}^{r} \theta_i^2 \mathcal{L}_{ii}$$

$$+ \sum_{i=1}^{r} (\theta_i^-)^2 \breve{\mathcal{L}}_{ii} + \sum_{i=1}^{r} \sum_{j=1}^{r} \theta_i^- \theta_j \left(\bar{\mathcal{L}}_{ij} + \bar{\mathcal{L}}_{ij}^T \right) + \sum_{i=1}^{r} \left(\sum_{j=1}^{i-1} \theta_i \theta_j \mathcal{L}_{ij} + \sum_{j=i+1}^{r} \theta_i \theta_j \mathcal{L}_{ij}^T \right)$$

$$+ \sum_{i=1}^{r} \left(\sum_{j=1}^{i-1} \theta_i^- \theta_j^- \breve{\mathcal{L}}_{ij} + \sum_{j=i+1}^{r} \theta_i^- \theta_j^- \breve{\mathcal{L}}_{ij}^T \right) \quad (55)$$

subject to, for $i, j \in [1, r]$, $j \neq i$,

$$(C1) \sum_{i=1}^{r} \theta_i = 1, \sum_{i=1}^{r} \theta_i^- = 1, (C2)\, 0 \leq \theta_i \leq \beta_i, 0 \leq \theta_i^- \leq \beta_i,$$

$$(C3)\, 0 \leq \theta_i \theta_j, 0 \leq \theta_i^- \theta_j^-, (C4) |\, \theta_i - \theta_i^- | \leq \delta_i \leq 1. \quad (56)$$

By the S-procedure (Boyd et al, 1994) and Finsler's lemma (de Oliveira and Skelton, 2001; Fang et al, 2004), the condition in Case 3 can be written as follows:

$$0 < \mathcal{L}(\theta(t-1), \theta(t)) - \mathcal{N}(\theta(t-1), \theta(t)), \quad (57)$$

where $0 \leq \mathcal{N}(\theta(t-1), \theta(t))$ is given by

$$\mathcal{N}(\theta(t-1), \theta(t)) = \mathcal{C}_1 + \mathcal{C}_1^T + \sum_{i=1}^{r} \mathcal{C}_{2i} (\Lambda_i + \Lambda_i^T) + \sum_{i=1}^{r} \breve{\mathcal{C}}_{2i} (\breve{\Lambda}_i + \breve{\Lambda}_i^T)$$

$$+\sum_{i=1}^{r}\sum_{j=1,j\neq i}^{r}\mathcal{C}_{3ij}\left(\Xi_{ij}+\Xi_{ij}^T\right)+\sum_{i=1}^{r}\sum_{j=1,j\neq i}^{r}\breve{\mathcal{C}}_{3ij}\left(\breve{\Xi}_{ij}+\breve{\Xi}_{ij}^T\right)+\sum_{i=1}^{r}\mathcal{C}_{4i}\left(Z_i+Z_i^T\right), \tag{58}$$

in which $\mathcal{C}_1$, $\mathcal{C}_{2i}$, $\breve{\mathcal{C}}_{2i}$, $\mathcal{C}_{3ij}$, $\breve{\mathcal{C}}_{3ij}$ and $\mathcal{C}_{4i}$ are from (C1)–(C4):

$$0=\mathcal{C}_1 \triangleq \begin{bmatrix} I \\ \theta_1 I \\ \vdots \\ \theta_r I \\ \theta_1^- I \\ \vdots \\ \theta_r^- I \end{bmatrix}^T \begin{bmatrix} I & I \\ -I & 0 \\ \vdots & \vdots \\ -I & 0 \\ 0 & -I \\ \vdots & \vdots \\ 0 & -I \end{bmatrix} \begin{bmatrix} X_0^T & Y_0^T \\ X_1^T & Y_1^T \\ \vdots & \vdots \\ X_r^T & Y_r^T \\ R_1 & S_1 \\ \vdots & \vdots \\ R_r & S_r \end{bmatrix}^T \begin{bmatrix} I \\ \theta_1 I \\ \vdots \\ \theta_r I \\ \theta_1^- I \\ \vdots \\ \theta_r^- I \end{bmatrix}, \tag{59}$$

$$0 \le \mathcal{C}_{2i} \triangleq -\theta_i^2+\beta_i\theta_i,\ 0 \le \breve{\mathcal{C}}_{2i} \triangleq -(\theta_i^-)^2+\beta_i\theta_i^-, \tag{60}$$

$$0 \le \mathcal{C}_{3ij} \triangleq \theta_i\theta_j,\ 0 \le \breve{\mathcal{C}}_{3ij} \triangleq \theta_i^-\theta_j^-,\ 0 \le \mathcal{C}_{4i} \triangleq \delta_i^2-\theta_i^2+2\theta_i\theta_i^- -(\theta_i^-)^2. \tag{61}$$

Here, note that the multiplier variables X_0, X_i, Y_0, Y_i, R_i, S_i, Λ_i, $\breve{\Lambda}_i$, Ξ_{ij}, $\breve{\Xi}_{ij}$, and Z_i are in $\mathcal{R}^{n_c\times n_c}$ and should satisfy that $0<\Lambda_i+\Lambda_i^T$, $0<\breve{\Lambda}_i+\breve{\Lambda}_i^T$, $0<\Xi_{ij}+\Xi_{ij}^T$, $0<\breve{\Xi}_{ij}+\breve{\Xi}_{ij}^T$, and $0<Z_i+Z_i^T$. Further, with some algebraic manipulations, the constraint $0\le\mathcal{N}(\theta(t-1),\theta(t))$ can be rewritten as follows:

$$0\le\mathcal{N}(\theta(t-1),\theta(t))=N_0+\sum_{i=1}^{r}\theta_i\left(N_i+N_i^T\right)+\sum_{i=1}^{r}\theta_i^-\left(N_i+N_i^T\right)+\sum_{i=1}^{r}\theta_i^2 N_{ii}$$

$$+\sum_{i=1}^{r}(\theta_i^-)^2 N_{ii}+\sum_{i=1}^{r}\sum_{j=1}^{r}\theta_i^-\theta_j\left(N_{ij}+N_{ij}^T\right)+\sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i\theta_j N_{ij}+\sum_{j=i+1}^{r}\theta_i\theta_j N_{ij}^T\right)$$

$$+\sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i^-\theta_j^- N_{ij}+\sum_{j=i+1}^{r}\theta_i^-\theta_j^- N_{ij}^T\right),$$

where

$$N_0=(X_0+X_0^T)+(Y_0+Y_0^T)+\sum_{i=1}^{r}\delta_i^2(Z_i+Z_i^T),\ N_i=X_i+Y_i-X_0+\beta_i\Lambda_i,$$

$$N_i=R_i+S_i-Y_0+\beta_i\breve{\Lambda}_i,\ N_{ii}=-(X_i+X_i^T)-(\Lambda_i+\Lambda_i^T)-(Z_i+Z_i^T),$$

$$N_{ii}=-(S_i+S_i^T)-(\breve{\Lambda}_i+\breve{\Lambda}_i^T)-(Z_i+Z_i^T),\ N_{ij}=\begin{cases}-R_i-Y_i+2Z_i, & j=i\\ -R_i-Y_j, & j\neq i\end{cases}',$$

$$N_{ij} = -X_i - X_j + \Xi_{ij} + \Xi_{ji}, N_{ij} = -S_i - S_j + \breve{\Xi}_{ij} + \breve{\Xi}_{ji}.$$

Hence, the condition (35) becomes, for all $\ell, m \in [1, r]$,

$$0 < \Gamma_0 + \sum_{i=1}^{r}\theta_i\left(\Gamma_i + \Gamma_i^T\right) + \sum_{i=1}^{r}\theta_i^-\left(\breve{\Gamma}_i + \breve{\Gamma}_i^T\right) + \sum_{i=1}^{r}\theta_i^2\Delta_i$$

$$+\sum_{i=1}^{r}(\theta_i^2)^-\breve{\Delta}_i + \sum_{i=1}^{r}\sum_{j=1}^{r}\theta_i^-\theta_j\left(\Pi_{ij} + \Pi_{ij}^T\right) + \sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i\theta_j\Phi_{ij} + \sum_{j=i+1}^{r}\theta_i\theta_j\Phi_{ij}^T\right)$$

$$+\sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i^-\theta_j^-\breve{\Phi}_{ij} + \sum_{j=i+1}^{r}\theta_i^-\theta_j^-\breve{\Phi}_{ij}^T\right), \qquad (62)$$

where $\Gamma_0 = \mathcal{L}_0 - N_0$, $\Gamma_i = \mathcal{L}_i - N_i$, $\breve{\Gamma}_i = \breve{\mathcal{L}}_i - N_i$, $\Delta_i = \mathcal{L}_{ii} - N_{ii}$, $\breve{\Delta}_i = \breve{\mathcal{L}}_{ii} - N_{ii}$, $\Phi_{ij} = \mathcal{L}_{ij} - N_{ij}$, $\breve{\Phi}_{ij} = \breve{\mathcal{L}}_{ij} - N_{ij}$, $\Pi_{ij} = \bar{\mathcal{L}}_{ij} - N_{ij}$. As a result, the condition in (62) boils down to

$$0 < \begin{bmatrix} I & \theta_1 I & \cdots & \theta_r I & \theta_1^- I & \cdots & \theta_r^- I \end{bmatrix} \tilde{\mathcal{L}} \begin{bmatrix} I & \theta_1 I & \cdots & \theta_r I & \theta_1^- I & \cdots & \theta_r^- I \end{bmatrix}^T,$$

where

$$\tilde{\mathcal{L}} \triangleq \left[\begin{array}{c|cccc|cccc}
\Gamma_0 & (*) & (*) & \cdots & (*) & (*) & (*) & (*) & (*) \\
\hline
\Gamma_1 & \Delta_1 & (*) & \cdots & (*) & (*) & \cdots & & (*) \\
\Gamma_2 & \Phi_{21} & \Delta_2 & \ddots & \vdots & (*) & \cdots & & (*) \\
\vdots & \vdots & \ddots & \ddots & (*) & \vdots & & & \vdots \\
\Gamma_r & \Phi_{r1} & \cdots & \Phi_{r(r-1)} & \Delta_r & (*) & \cdots & & (*) \\
\hline
\breve{\Gamma}_1 & \Pi_{11} & \cdots & & \Pi_{1r} & \breve{\Delta}_1 & (*) & \cdots & (*) \\
\breve{\Gamma}_2 & \Pi_{21} & \cdots & & \Pi_{2r} & \breve{\Phi}_{21} & \breve{\Delta}_2 & \ddots & \vdots \\
\vdots & \vdots & & & \vdots & \vdots & \ddots & \ddots & (*) \\
\breve{\Gamma}_r & \Pi_{r1} & \cdots & & \Pi_{rr} & \breve{\Phi}_{r1} & \cdots & \breve{\Phi}_{r(r-1)} & \breve{\Delta}_r
\end{array}\right]. \qquad (63)$$

Proposition 3 The condition in (55) subject to (C1) - (C4) holds if the following conditions hold: for all $i, j \in [1, r]$, $j \neq i$,

$$0 < \tilde{\mathcal{L}}, 0 < \Lambda_i + \Lambda_i^T, 0 < \breve{\Lambda}_i + \breve{\Lambda}_i^T, 0 < \Xi_{ij} + \Xi_{ij}^T, 0 < \breve{\Xi}_{ij} + \breve{\Xi}_{ij}^T, 0 < Z_i + Z_i^T. \qquad (64)$$

4. Stabilization of T-S fuzzy systems

A. Lyapunov function

Based on the modeled T-S fuzzy systems (see Section 2), various feedback controllers have been recently developed under the well-known relaxation techniques (see Section 3), by taking advantage of Lyapunov functions, such as CQLF, PLF, and FWLF (refer to Section 1). The common quadratic Lyapunov function (CQLF) is given as follows:

$$V(x(t)) = x^T(t)Px(t), P > 0, \tag{65}$$

and the fuzzy weighting-dependent Lyapunov function (FWLF) can be written as follows:

$$V(x(t)) = x^T(t)P(\theta(t))x(t), P(\theta(t)) > 0. \tag{66}$$

As mentioned in Section 1, since, for a large number of T-S fuzzy subsystems (A_i, B_i, C_i), the former approach leads to over-conservative design solutions, recent research efforts have focused on using the latter approach when establishing a feedback controller that ensures the stability of the closed-loop system.
When using the FWLF approach for stabilization of continuous-time T-S fuzzy systems, one needs to assume that the upper and lower bounds of $\dot{\theta}(t)$ are measurable and the matrix inverse is computed in real time. Since it is very strict to make the assumptions, one has made less use of the FWLF in continuous-time F-S fuzzy systems, compared with the discrete-time case. Furthermore, since it is impractical to fully measure the rate vector $\dot{\theta}(t)$, the controller should be constructed only by the current-time information on parameters, which leads to an inflexible result for more changeable T-S fuzzy systems. In the discrete-time case, on the other hand, it is possible to make a feedback control law dependent not only on the current-time parameters but also the one-step-past parameters.
Now, let us consider the matrix $P(\theta(t))$ of the following form:

FWLF with APD (FWLF-APD):

$$P(\theta(t)) = \sum_{i=1}^{r}\theta_i(t)P_i, P_i > 0, \sum_{i=1}^{r}\theta_i(t) = 1, \theta_i(t) \geq 0, i = 1,\cdots,r, \tag{67}$$

FWLF with QPD (PDLF-QPD):

$$P(\theta(t)) = \sum_{i=1}^{r}\sum_{j=1}^{r}\theta_i(t)\theta_j(t)P_{ij}, P_{ij} > 0, \sum_{i=1}^{r}\theta_i(t) = 1, \theta_i(t) \geq 0, i = 1,\cdots,r. \tag{68}$$

B. Stabilization of continuous-time T-S fuzzy systems

Consider a T-S fuzzy system described by the following differential equation:

$$\dot{x}(t) = A(\theta(t))x(t) + B(\theta(t))u(t), \tag{69}$$

where $x(t) \in R^{n_x}$ and $u(t) \in R^{n_u}$ denote the state and control input, respectively; and

$$A(\theta(t)) = A_0 + \sum_{i=1}^{r}\theta_i(t)A_i, B(\theta(t)) = B_0 + \sum_{i=1}^{r}\theta_i(t)B_i. \tag{70}$$

For the stabilization of (69), consider the following fuzzy weighting-dependent state-feedback controller:

$$u(t) = F(\theta(t))x(t). \tag{71}$$

Then, the closed-loop system under (71) is described as follows:

$$\dot{x}(t) = \left(A(\theta(t)) + B(\theta(t))F(\theta(t))\right)x(t). \tag{72}$$

Stabilization using CQLF:

In the case where $\theta(t)$ is measurable with known bounds but the full information on $\dot{\theta}(t)$ is unknown, the CQLF approach is valuable and applicable to T-S fuzzy systems.
*Theorem 1***:** (PLMI-based condition) Suppose that there exist matrices $\overline{P} > 0$ and $\overline{F}(\theta(t))$ such that

$$\overline{P}A^T(\theta(t)) + \overline{F}^T(\theta(t))B^T(\theta(t)) + A(\theta(t))\overline{P} + B(\theta(t))\overline{F}(\theta(t)) < 0, \tag{73}$$

where $\overline{P} = P^{-1}$ and $\overline{F}(\theta(t)) = F(\theta(t))\overline{P}$. Then, the closed-loop system in (72) is asymptotically stable for all admissible grades. Moreover, the controller is reconstructed as follows:

$$F(\theta(t)) = \overline{F}(\theta(t))\overline{P}^{-1}. \tag{74}$$

Proof The proof is straightforward, and hence is omitted here. ■
In the following, we shall derive a finite number of LMIs from (73). To this end, let us first consider

$$\overline{F}(\theta(t)) = \overline{F}_0 + \sum_{i=1}^{r}\theta_i(t)\overline{F}_i. \tag{75}$$

Then, we can obtain the following theorem with the help of the relaxation technique in Case 1.
Theorem 2: (LMI-based condition) Suppose that there exist matrices $\overline{P} > 0$, $\overline{F}_0$, $\overline{F}_i$, S_0, S_i, Λ_i, and Ξ_{ij} such that

$$0 < \left[\begin{array}{c|cccc} \Gamma_0 & (*) & (*) & \cdots & (*) \\ \hline \Gamma_1 & \Delta_1 & (*) & \cdots & (*) \\ \Gamma_2 & \Phi_{21} & \Delta_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & (*) \\ \Gamma_r & \Phi_{r1} & \cdots & \Phi_{r(r-1)} & \Delta_r \end{array}\right], \tag{76}$$

$$0 \le \Lambda_i + \Lambda_i^T, 0 \le \Xi_{ij} + \Xi_{ji}^T, \forall i, j \in [1, r], j \ne i. \tag{77}$$

Then, the closed-loop system in (72) is asymptotically stable for all admissible grades satisfying the constraints (34). Moreover, the controller is given by

$$F(\theta(t)) = \left(\overline{F}_0 + \sum_{i=1}^{r}\theta_i(t)\overline{F}_i\right)\overline{P}^{-1}. \tag{78}$$

Proof The condition in (73) is equivalent to

$$0 < \mathcal{L}(\theta(t)) \triangleq -\left(\overline{P}A^T(\theta(t)) + \overline{F}^T(\theta(t))B^T(\theta(t)) + A(\theta(t))\overline{P} + B(\theta(t))\overline{F}(\theta(t))\right),$$

which can be rewritten by (70) and (75) into

$$0<\mathcal{L}_0+\sum_{i=1}^{r}\theta_i(t)\left(\mathcal{L}_i+\mathcal{L}_i^T\right)+\sum_{i=1}^{r}\theta_i^2(t)\mathcal{L}_{ii}+\sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i(t)\theta_j(t)\mathcal{L}_{ij}+\sum_{j=i+1}^{r}\theta_i(t)\theta_j(t)\mathcal{L}_{ij}^T\right), \tag{79}$$

where

$$\mathcal{L}_0=-\left(A_0\overline{P}+B_0\overline{F}_0+\overline{P}A_0^T+\overline{F}_0^TB_0^T\right), \mathcal{L}_i=-\left(A_i\overline{P}+B_0\overline{F}_i+B_i\overline{F}_0\right), \tag{80}$$

$$\mathcal{L}_{ii}=-\left(B_i\overline{F}_i+\overline{F}_i^TB_i^T\right), \mathcal{L}_{ij}=-\left(B_i\overline{F}_j+B_j\overline{F}_i\right). \tag{81}$$

Hence, reminding the relaxation technique in Case 1, we can clearly see that the condition $\mathcal{L}(\theta(t))>0$ subject to (34) is guaranteed by (76) and (77), where

$$\Gamma_0=\mathcal{L}_0-S_0-S_0^T, \Gamma_i=\mathcal{L}_i-\beta_i\Lambda_i+S_0-S_i, \tag{82}$$

$$\Delta_i=\mathcal{L}_{ii}+\left(\Lambda_i+\Lambda_i^T\right)+\left(S_i+S_i^T\right), \tag{83}$$

$$\Phi_{ij}=\mathcal{L}_{ij}+\left(S_i+S_j\right)-\left(\Xi_{ij}+\Xi_{ji}\right). \tag{84}$$

■

Stabilization using FWLF-APD:

Let us assume that $\theta(t)$ is measurable and the bounds of $\dot{\theta}(t)$ are given as

$$\rho_i\le\dot{\theta}_i(t)\le\nu_i, \text{for } i=1,\cdots,r. \tag{85}$$

Then, in this sense, we can consider a fuzzy weighting-dependent Lyapunov function $V(x(t))$ such as

$$V(x(t))=x^T(t)P(\theta(t))x(t), P(\theta(t))>0, \tag{86}$$

based on which we can obtain the following PLMI-based stabilization condition.
Theorem 3: (PLMI-based condition) Suppose that there exist matrices $\overline{P}(\theta(t))>0$, $\dot{\overline{P}}(\theta(t))$, and $\overline{F}(\theta(t))$ such that

$$\dot{\overline{P}}(\theta(t))>\overline{P}(\theta(t))A^T(\theta(t))+\overline{F}^T(\theta(t))B^T(\theta(t))+A(\theta(t))\overline{P}(\theta(t))+B(\theta(t))\overline{F}(\theta(t)), \tag{87}$$

where $\overline{P}(\theta(t))=P^{-1}(\theta(t))$ and $\overline{F}(\theta(t))=F(\theta(t))\overline{P}(\theta(t))$. Then, the closed-loop system in (72) is asymptotically stable for all admissible grades. Moreover, the controller is given by

$$F(\theta(t))=\overline{F}(\theta(t))\overline{P}^{-1}(\theta(t)). \tag{88}$$

Proof The proof is straightforward, and hence is omitted here. ■
In the following, we shall derive a finite number of LMIs from (87). To this end, let us take

$$\overline{P}(\theta(t)) = \overline{P} + \sum_{i=1}^{r}\theta_i(t)\overline{P}_i,\ \overline{P} > 0, \overline{P}_i > 0. \tag{89}$$

Then, we can obtain the following theorem with the help of the relaxation scheme in Case 2.

Theorem 4: (LMI-based condition) Suppose that there exist matrices $\overline{P} > 0$, $\overline{P}_i > 0$, $\overline{F}_0$, $\overline{F}_i$, S_0, S_i, Λ_i, Ξ_{ij} and Z_i such that

$$0 < \left[\begin{array}{c|cccc|cccc}
\Gamma_0 & (*) & (*) & \cdots & (*) & (*) & (*) & (*) & (*) \\
\hline
\Gamma_1 & \Delta_1 & (*) & \cdots & (*) & 0 & \cdots & & 0 \\
\Gamma_2 & \Phi_{21} & \Delta_2 & \ddots & \vdots & 0 & \cdots & & 0 \\
\vdots & \vdots & \ddots & \ddots & (*) & \vdots & & & \vdots \\
\Gamma_r & \Phi_{r1} & \cdots & \Phi_{r(r-1)} & \Delta_r & 0 & \cdots & & 0 \\
\hline
\breve{\Gamma}_1 & 0 & \cdots & & 0 & \breve{\Delta}_1 & 0 & \cdots & 0 \\
\breve{\Gamma}_2 & 0 & \cdots & & 0 & 0 & \breve{\Delta}_2 & \ddots & \vdots \\
\vdots & \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\
\breve{\Gamma}_r & 0 & \cdots & & 0 & 0 & \cdots & 0 & \breve{\Delta}_r
\end{array}\right], \tag{90}$$

$$0 \le \Lambda_i + \Lambda_i^T,\ 0 \le \Xi_{ij} + \Xi_{ij}^T,\ 0 \le Z_i + Z_i^T,\ \forall i, j \in [1, r],\ j \ne i. \tag{91}$$

Then, the closed-loop system in (72) is asymptotically stable for all admissible grades satisfying the constraints (46). Moreover, the controller is given by

$$F(\theta(t)) = \left(\overline{F}_0 + \sum_{i=1}^{r}\theta_i(t)\overline{F}_i\right)\left(\overline{P} + \sum_{i=1}^{r}\theta_i(t)\overline{P}_i\right)^{-1}. \tag{92}$$

Proof The condition in (87) is equivalent to

$$0 < \mathcal{L}(\theta(t)) \triangleq \dot{\overline{P}}(\theta(t)) - \overline{P}(\theta(t))A^T(\theta(t)) - \overline{F}^T(\theta(t))B^T(\theta(t))$$

$$-A(\theta(t))\overline{P}(\theta(t)) - B(\theta(t))\overline{F}(\theta(t)),$$

which can be rewritten by (70), (75), (89) into

$$0 < \mathcal{L}_0 + \sum_{i=1}^{r}\theta_i(t)\left(\mathcal{L}_i + \mathcal{L}_i^T\right) + \sum_{i=1}^{r}\dot{\theta}_i(t)\left(\breve{\mathcal{L}}_i + \breve{\mathcal{L}}_i^T\right) + \sum_{i=1}^{r}\theta_i^2(t)\mathcal{L}_{ii}$$

$$+\sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i(t)\theta_j(t)\mathcal{L}_{ij} + \sum_{j=i+1}^{r}\theta_i(t)\theta_j(t)\mathcal{L}_{ij}^T\right), \tag{93}$$

where

$$\mathcal{L}_0 = -\left(A_0\overline{P} + B_0\overline{F}_0 + \overline{P}A_0^T + \overline{F}_0^TB_0^T\right), \tag{94}$$

$$\mathcal{L}_i = -\left(A_0\overline{P}_i + A_i\overline{P} + B_0\overline{F}_i + B_i\overline{F}_0\right), \breve{\mathcal{L}}_i = \frac{1}{2}\overline{P}_i, \tag{95}$$

$$\mathcal{L}_{ii} = -\left(A_i\overline{P}_i + B_i\overline{F}_i + \overline{P}_i A_i^T + \overline{F}_i^T B_i^T\right), \tag{96}$$

$$\mathcal{L}_{ij} = -\left(A_i\overline{P}_j + B_i\overline{F}_j + A_j\overline{P}_i + B_j\overline{F}_i\right). \tag{97}$$

Hence, reminding the relaxation technique in Case 2, we can clearly see that the condition $\mathcal{L}(\theta(t)) > 0$ subject to (46) is guaranteed by (90) and (91), where

$$\Gamma_0 = \mathcal{L}_0 - S_0 - S_0^T + \sum_{i=1}^{r}\rho_i\nu_i(Z_i + Z_i^T), \Gamma_i = \mathcal{L}_i - \beta_i\Lambda_i + S_0 - S_i, \tag{98}$$

$$\breve{\Gamma}_i = \breve{\mathcal{L}}_i - (\rho_i + \nu_i)Z_i, \Delta_i = \mathcal{L}_{ii} + \left(\Lambda_i + \Lambda_i^T\right) + \left(S_i + S_i^T\right), \tag{99}$$

$$\breve{\Delta}_i = Z_i + Z_i^T, \Phi_{ij} = \mathcal{L}_{ij} + \left(S_i + S_j\right) - \left(\Xi_{ij} + \Xi_{ji}\right). \tag{100}$$

■

C. Stabilization of discrete-time LPV systems

Consider the following discrete-time T-S fuzzy systems:

$$x(k+1) = A(\theta(k))x(k) + B(\theta(k))u(k), \tag{101}$$

where $x(k) \in R^{n_x}$ and $u(k) \in R^{n_u}$ denote the state and control input, respectively; $\theta(k)$ denotes the time-varying parameter vector; and

$$A(\theta(k)) = \sum_{i=1}^{r}\theta_i(k)A_i, B(\theta(k)) = \sum_{i=1}^{r}\theta_i(k)B_i. \tag{102}$$

For the stabilization of (101), we shall consider a state-feedback controller dependent not only the current-time parameter vector $\theta(k)$ but also on the one-step-past vector $\theta(k-1)$ for time k:

$$u(k) = F(\theta(k-1), \theta(k))x(k). \tag{103}$$

Remark 3: The reason for using both $\theta(k)$ and $\theta(k-1)$ in (103) is twofold. One is to enhance the causality between the control gain and the Lyapunov function whose forward difference is a function of $\theta(k)$ and $\theta(k-1)$. The other is to use the information existing between $\theta(k-1)$ and $\theta(k)$ as well as the instant information $\theta(k)$ when performing the control action (Choi & Park, 2003; Kim et al, 2004).

As a result, the resulting closed-loop system under (103) is described as follows:

$$x(k+1) = \hat{A}(\theta(k-1), \theta(k))x(k), \tag{104}$$

where

$$\hat{A}(\theta(k-1),\theta(k)) \triangleq A(\theta(k)) + B(\theta(k))F(\theta(k-1),\theta(k)). \tag{105}$$

Stabilization using CQLF:

First, let us consider a common quadratic Lyapunov function $V(x(k))$:

$$V(x(k)) = x^T(k)Px(k), P > 0, \tag{106}$$

whose forward difference along the closed-loop system trajectories is given by

$$\Delta V(x(k)) = V(x(k+1)) - V(x(k)) = x^T(k+1)Px(k+1) - x^T(k)Px(k). \tag{107}$$

Then, the stabilization condition for the closed-loop system in (104) is readily given as follows:

$$0 < P - \hat{A}^T(\theta(k-1),\theta(k))P\hat{A}(\theta(k-1),\theta(k)). \tag{108}$$

Theorem 5: (PLMI-based condition) Suppose that there exist matrices $\overline{P}$ and $\overline{F}(\theta(k-1),\theta(k))$ such that

$$0 < \begin{bmatrix} \overline{P} & (*) \\ A(\theta(k))\overline{P} + B(\theta(k))\overline{F}(\theta(k-1),\theta(k)) & \overline{P} \end{bmatrix}, \tag{109}$$

where $\overline{P} = P^{-1}$ and $\overline{F}(\theta(k-1),\theta(k)) = F(\theta(k-1),\theta(k))\overline{P}$. Then, the closed-loop system in (104) is asymptotically stable for all admissible grades $\theta(k-1)$ and $\theta(k)$. Moreover, the controller is given by

$$F(\theta(k-1),\theta(k)) = \overline{F}(\theta(k-1),\theta(k))\overline{P}^{-1}. \tag{110}$$

Proof The proof is straightforward, and hence is omitted here. ■

In the following, we shall derive a finite number of LMIs from (109). To this end, let us first take

$$\overline{F}(\theta(k-1),\theta(k)) = \sum_{i=1}^{r}\theta_i^- \overline{F}_i^- + \overline{F}(\theta(k)), \overline{F}(\theta(k)) = \sum_{i=1}^{r}\theta_i\overline{F}_i. \tag{111}$$

Then, we can obtain the following theorem with the help of the relaxation scheme in Case 1.

Theorem 6: (LMI-based condition) Suppose that there exist matrices $\overline{P} > 0$, $\overline{F}_i^-$, $\overline{F}_i$, S_0, S_i, Λ_i and Ξ_{ij} such that

$$0 < \left[\begin{array}{c|cccc} \Gamma_0 & (*) & (*) & \cdots & (*) \\ \hline \Gamma_1^{(l)} & \Delta_1 & (*) & \cdots & (*) \\ \Gamma_2^{(l)} & \Phi_{21} & \Delta_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & (*) \\ \Gamma_r^{(l)} & \Phi_{r1} & \cdots & \Phi_{r(r-1)} & \Delta_r \end{array}\right], \forall l \in [1,r], \tag{112}$$

$$0 \le \Lambda_i + \Lambda_i^T, 0 \le \Xi_{ij} + \Xi_{ji}^T, \forall i, j \in [1,r], j \ne i. \tag{113}$$

Then, the closed-loop system in (104) is asymptotically stable for all admissible grades $\theta(k-1)$ and $\theta(k)$. Moreover, the controller is given by

$$F(\theta(k-1),\theta(k)) = \left(\sum_{i=1}^{r} \theta_i^- \bar{F}_i^- + \sum_{i=1}^{r} \theta_i \bar{F}_i \right)^{-1} \bar{P}^{-1}. \tag{114}$$

Proof Define

$$\mathcal{L}_l(\theta(k)) \triangleq \begin{bmatrix} \bar{P} & (*) \\ A(\theta(k))\bar{P} + B(\theta(k))(\bar{F}_l^- + \bar{F}(\theta(k))) & \bar{P} \end{bmatrix}. \tag{115}$$

Then, the PLMI-based condition in (109) can be written as follows:

$$0 < \sum_{l=1}^{r} \theta_l^- \mathcal{L}_l(\theta(k)), \tag{116}$$

which is equivalent to $0 < \mathcal{L}_l(\theta(k))$, for all $l \in [1,r]$, that is,

$$0 < \mathcal{L}_0 + \sum_{i=1}^{r} \theta_i(k)\left(\mathcal{L}_i^{(l)} + \mathcal{L}_i^{(l)T}\right) + \sum_{i=1}^{r} \theta_i^2(k)\mathcal{L}_{ii} + \sum_{i=1}^{r}\left(\sum_{j=1}^{i-1} \theta_i(k)\theta_j(k)\mathcal{L}_{ij} + \sum_{j=i+1}^{r} \theta_i(k)\theta_j(k)\mathcal{L}_{ij}^T \right), \tag{117}$$

where

$$\mathcal{L}_0 = \begin{bmatrix} \bar{P} & 0 \\ 0 & \bar{P} \end{bmatrix}, \mathcal{L}_i^{(l)} = \begin{bmatrix} 0 & 0 \\ A_i\bar{P} + B_i\bar{F}_l^- & 0 \end{bmatrix}, \tag{118}$$

$$\mathcal{L}_{ii} = \begin{bmatrix} 0 & (*) \\ B_i\bar{F}_i & 0 \end{bmatrix}, \mathcal{L}_{ij} = \begin{bmatrix} 0 & 0 \\ B_i\bar{F}_j + B_j\bar{F}_i & 0 \end{bmatrix}. \tag{119}$$

Hence, reminding the relaxation technique in Case 1, we can clearly see that the condition $0 < \mathcal{L}_l(\theta(k))$ subject to (34) is guaranteed by (112) and (113), where

$$\Gamma_0 = \mathcal{L}_0 - S_0 - S_0^T, \Gamma_i^{(l)} = \mathcal{L}_i^{(l)} - \beta_i\Lambda_i + S_0 - S_i, \tag{120}$$

$$\Delta_i = \mathcal{L}_{ii} + \left(\Lambda_i + \Lambda_i^T\right) + \left(S_i + S_i^T\right), \tag{121}$$

$$\Phi_{ij} = \mathcal{L}_{ij} + \left(S_i + S_j\right) - \left(\Xi_{ij} + \Xi_{ji}\right). \tag{122}$$

■

Stabilization using FWLF-APD:

Consider an FWLF $V(x(k))$ of the following form:

$$V(x(k)) = x^T(k)P(\theta(k))x(k), P(\theta(k)) > 0, \tag{123}$$

whose forward difference along the closed-loop system trajectories is given by

$$\Delta V(x(k)) = x^T(k+1)P(\theta(k+1))x(k+1) - x^T(k)P(\theta(k))x(k). \tag{124}$$

Then, the stabilization condition for (104) is readily given as follows:

$$0 < P(\theta(k)) - \hat{A}^T(\theta(k-1),\theta(k))P(\theta(k+1))\hat{A}(\theta(k-1),\theta(k)). \tag{125}$$

Theorem 7: (PLMI-based condition) Suppose that there exist matrices $\overline{P}(\theta(k+1))$, $\overline{P}(\theta(k))$, and $\overline{F}(\theta(k-1),\theta(k))$ such that

$$0 < \begin{bmatrix} \overline{P}(\theta(k)) & (*) \\ A(\theta(k))\overline{P}(\theta(k)) + B(\theta(k))\overline{F}(\theta(k-1),\theta(k)) & \overline{P}(\theta(k+1)) \end{bmatrix}, \tag{126}$$

where $\overline{P}(\cdot) = P^{-1}(\cdot)$ and $\overline{F}(\theta(k-1),\theta(k)) = F(\theta(k-1),\theta(k))\overline{P}(\theta(k))$. Then, the closed-loop system in (104) is asymptotically stable for all admissible grades. Moreover, the controller is reconstructed as follows:

$$F(\theta(k-1),\theta(k)) = \overline{F}(\theta(k-1),\theta(k))\overline{P}^{-1}(\theta(k)). \tag{127}$$

Proof The proof is straightforward, and hence is omitted here. ■

In the following, we shall derive a finite number of LMIs from (126). To this end, let us first take

$$\overline{P}(\theta(k)) = \overline{P} + \sum_{i=1}^{r}\theta_i(k)\overline{P}_i, \overline{P} > 0, \overline{P}_i > 0, \tag{128}$$

$$\overline{F}(\theta(k-1),\theta(k)) = \sum_{i=1}^{r}\theta_i(k-1)\overline{F}_i^- + \overline{F}(\theta(k)), \overline{F}(\theta(k)) = \sum_{i=1}^{r}\theta_i(k)\overline{F}_i. \tag{129}$$

Then, we can obtain the following theorem with the help of the relaxation technique in Case 1.

Theorem 8: (LMI-based condition) Suppose that there exist matrices $\overline{P} > 0$, $\overline{P}_i > 0$, $\overline{F}_i^-$, $\overline{F}_i$, S_0, S_i, Λ_i, and Ξ_{ij} such that

$$0 < \left[\begin{array}{c|cccc} \Gamma_0^{(s)} & (*) & (*) & \cdots & (*) \\ \hline \Gamma_1^{(l)} & \Delta_1 & (*) & \cdots & (*) \\ \Gamma_2^{(l)} & \Phi_{21} & \Delta_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & (*) \\ \Gamma_r^{(l)} & \Phi_{r1} & \cdots & \Phi_{r(r-1)} & \Delta_r \end{array}\right], \forall l,s \in [1,r], \tag{130}$$

$$0 \le \Lambda_i + \Lambda_i^T, 0 \le \Xi_{ij} + \Xi_{ji}^T, \forall i,j \in [1,r], j \ne i. \tag{131}$$

Then, the closed-loop system in (104) is asymptotically stable for all admissible grades satisfying the constraints (34). Moreover, the controller is given by

$$F(\theta(k-1),\theta(k)) = \left(\sum_{i=1}^{r}\theta_i(k-1)\overline{F}_i^- + \sum_{i=1}^{r}\theta_i(k)\overline{F}_i\right)\left(\overline{P} + \sum_{i=1}^{r}\theta_i(k)\overline{P}_i\right)^{-1}. \tag{132}$$

Proof Define

$$\mathcal{L}_{ls}(\theta(k)) \triangleq \begin{bmatrix} \overline{P}(\theta(k)) & (*) \\ A(\theta(k))\overline{P}(\theta(k)) + B(\theta(k))(\overline{F}_l^- + \overline{F}(\theta(k))) & \overline{P} + \overline{P}_s \end{bmatrix}. \tag{133}$$

Then, the PLMI condition (126) can be rewritten as follows:

$$0 < \sum_{l=1}^{r}\sum_{s=1}^{r}\theta_l(k-1)\theta_s(k+1)\mathcal{L}_{ls}(\theta(k)), \tag{134}$$

which is equivalent to $0 < \mathcal{L}_{ls}(\theta(k))$, for all $l,s \in [1,r]$, that is,

$$0 < \mathcal{L}_0^{(s)} + \sum_{i=1}^{r}\theta_i(k)\left(\mathcal{L}_i^{(l)} + \mathcal{L}_i^{(l)T}\right) + \sum_{i=1}^{r}\theta_i^2(k)\mathcal{L}_{ii} + \sum_{i=1}^{r}\left(\sum_{j=1}^{i-1}\theta_i(k)\theta_j(k)\mathcal{L}_{ij} + \sum_{j=i+1}^{r}\theta_i(k)\theta_j(k)\mathcal{L}_{ij}^T\right), \tag{135}$$

where

$$\mathcal{L}_0^{(s)} = \begin{bmatrix} \overline{P} & 0 \\ 0 & \overline{P} + \overline{P}_s \end{bmatrix}, \mathcal{L}_i^{(l)} = \begin{bmatrix} \frac{1}{2}\overline{P}_i & 0 \\ A_i\overline{P} + B_i\overline{F}_l^- & 0 \end{bmatrix}, \tag{136}$$

$$\mathcal{L}_{ii} = \begin{bmatrix} 0 & (*) \\ A_i\overline{P}_i + B_i\overline{F}_i & 0 \end{bmatrix}, \mathcal{L}_{ij} = \begin{bmatrix} 0 & 0 \\ A_i\overline{P}_j + A_j\overline{P}_i + B_i\overline{F}_j + B_j\overline{F}_i & 0 \end{bmatrix}. \tag{137}$$

Hence, reminding the relaxation technique in Case 1, we can clearly see that the condition $0 < \mathcal{L}_{ls}(\theta(k))$ subject to (34) is guaranteed by (130) and (131), where

$$\Gamma_0^{(s)} = \mathcal{L}_0^{(s)} - S_0 - S_0^T, \Gamma_i^{(l)} = \mathcal{L}_i^{(l)} - \beta_i\Lambda_i + S_0 - S_i, \tag{138}$$

$$\Delta_i = \mathcal{L}_{ii} + \left(\Lambda_i + \Lambda_i^T\right) + \left(S_i + S_i^T\right), \tag{139}$$

$$\Phi_{ij} = \mathcal{L}_{ij} + \left(S_i + S_j\right) - \left(\Xi_{ij} + \Xi_{ji}\right). \tag{140}$$

■

Stabilization using FWLF-QPD:

The relevant formulation has been given in our previous research works (Kim & Park, 2008), associate with the $\mathcal{H}_\infty$ performance. In the derivation, the $\mathcal{H}_\infty$ stabilization conditions are formulated in terms of PLMIs, which are reconverted into a finite set of LMI-based conditions with the help of the relaxation technique in Case 3.

5. Acknowledgment

This research was supported by WCU (World Class University) program through the Korea Science and Engineering Foundation funded by the Ministry of Education, Science and Technology (Project No. R31-2008-000-10100-0).

6. References

[1] M. Sugeno, "On stability of fuzzy systems expressed by fuzzy rules with singleton consequents," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 201–224, 1999.

[2] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 1, pp. 116--132, 1985.

[3] K. Tanaka and M. Sugeno, "Stability analysis and design of fuzzy control systems," *Fuzzy Sets Syst.*, vol. 45, pp. 135–156, 1992.

[4] K. Tanaka, T. Ikeda, and H. O. Wang, "Robust stabilization of a class of uncertain nonlinear systems via fuzzy control: Quadratic stabilizability $\mathcal{H}_\infty$ control theory, and linear matrix inequality," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 1, pp. 1–13, 1996.

[5] H. O.Wang, K. Tanaka, and M. Griffin, "An approach to fuzzy control on nonlinear sytems: Stability and design issues," *IEEE Trans. Fuzzy Syst.*, vol. 4, no. 1, pp. 14–23, Feb. 1996.

[6] Y.-Y. Cao and P. M. Frank, "Robust $\mathcal{H}_\infty$ disturbance attenuation for a class of uncertain discrete-time fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 4, pp. 406–415, 2000.

[7] W. Assawinchaichote, S. K. Nguang, and P. Shi, " $\mathcal{H}_\infty$ output feedback control design for uncertain fuzzy singularly perturbed systems: An LMI approach," *Automatica*, vol. 40, no. 12, pp. 2147–2152, Dec. 2004.

[8] S.-G. Cao, N. W. Ress, and G. Feng, "Stability analysis and design for a class of continuous time fuzzy control system," *Int. J. Control*, vol. 64, pp. 1069–1087, 1996.

[9] S.-G. Cao, N. W. Ress, and G. Feng, "Analysis and design for a class of complex control systems, Part II: Fuzzy controller design," *Automatica*, vol. 33, pp. 1029–1039, 1997.

[10] S.-G. Cao, N. W. Ress, and G. Feng, "Analysis and design of fuzzy control systems using dynamic fuzzy-state space models," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 2, pp. 192–200, Apr. 1999.

[11] Z. X. Han, G. Feng, B. L. Walcott, and J. Ma, "Dynamic output feedback controller design for fuzzy systems," *IEEE Trans. Syst., Man, Cybern., B, Cybern.*, vol. 30, no. 1, pp. 204–210, Feb. 2000.

[12] G. Feng, "Controller synthesis of fuzzy dynamic systems based on piecewise Lyapunov functions," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 10, pp. 605–612, Oct. 2003.

[13] G. Feng, " $\mathcal{H}_\infty$ controller design of fuzzy dynamic systems based on piecewise Lyapunov functions," *IEEE Trans. Syst., Man, Cybern., B., Cybern.*, vol. 34, no. 1, pp. 283–292, Feb. 2004.

[14] C. L. Chen, G. Feng, and X. P. Guan, "Delay-dependent stability analysis and controller synthesis for discrete time T-S fuzzy systems with time delays," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 5, pp. 630–643, Oct. 2005.

[15] K. Tanaka, T.Hori, and H.O.Wang, "A fuzzy Lyapunov approach to fuzzy conrol system design," *in Proc. Amer. Control. Conf.*, 2001, pp. 4790–4795.
[16] P. Park and D. J. Choi, "LPV controller design for the nonlinear RTAC system," *Int. J. Robust Nonlinear Control,* vol. 11, pp. 1342–1363, 2001.
[17] D. J. Choi and P. Park, "$\mathcal{H}_\infty$ state-feedback controller design for discrete-time fuzzy systems using fuzzy weighting-dependent Lyapunov functions," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 2, pp. 271–278, 2003.
[18] S. H. Kim and P. Park, "$\mathcal{H}_\infty$ state-feedback control design for fuzzy systems using Lyapunov functions with quadratic dependence on fuzzy weighting functions," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 6, pp. 1655–1663, 2008.
[19] S. H. Kim, C. H. Lee, and P. Park, "$\mathcal{H}_\infty$ state-feedback control for fuzzy systems with input saturation via fuzzy weighting-dependent Lyapunov functions," *Computers & Mathmatics with applications,* vol. 57, no. 6, pp. 981–990, 2009.
[20] B. S. Chen, C. S. Tseng, and H. J. Uang, "Robustness design of nonlinear dynamic systems via fuzzy linear control," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 5, pp. 571–585, Oct. 1999.
[21] S.-H. Tsai and T.-H. S. Li, "Robust fuzzy control of a class of fuzzy bilinear systems with time-delay," *Chaos, Solitons and Fractals*, vol. 39, pp. 2028–2040, 2009.
[22] B. S. Chen, C. S. Tseng, and H. J. Uang, "Mixed $\mathcal{H}_2 / \mathcal{H}_\infty$ fuzzy output feedback control design for nonlinear dynamic systems: An LMI approach," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 3, pp. 249–265, Jun. 2000.
[23] S. Zhou, G. Feng, J. Lam, and S. Xu, "Robust $\mathcal{H}_\infty$ control for discrete-time fuzzy systems via basis-dependent Lyapunov functions," *Information Sciences,* vol. 174, pp. 197–217, 2005.
[24] L. Xiaodong and Z. Qingling, "New approaches to $\mathcal{H}_\infty$ controller designs based on fuzzy observers for T-S fuzzy systems via LMI," *Automatica,* vol. 39, pp. 1571–1582, 2003.
[25] C.-L. Hwang and L.-J. Chang, "Internet-based smart-space navigation of a car-like wheeled robot using fuzzy-neural adaptive control," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1271–1284, Oct. 2008.
[26] X. Jiang and Q.-L. Han, "On designing fuzzy controllers for a class of nonlinear networked control systems," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 4, pp. 1050–1060, Aug. 2008.
[27] H. Gao, Y. Zhao, and T. Chen, "$\mathcal{H}_\infty$ fuzzy control of nonlinear systems under unreliable communication links," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 2, pp. 265–278, Apr. 2009.
[28] Y.-Y. Cao and P. M. Frank, "Analysis and synthesis of nonlinear time-delay systems via fuzzy control approach," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 2, pp. 200–211, Apr. 2000.
[29] C.-L. Chen, G. Feng, and X.-P. Guan, "Delay-dependent stability analysis and controller synthesis for discrete-time T-S fuzzy systems with time delays," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 5, pp. 630–643, Oct. 2005.
[30] H.-N. Wu, "Delay-dependent $\mathcal{H}_\infty$ fuzzy observer-based control for discrete-time nonlinear systems with state delay," *Fuzzy Sets and Systems,* vol. 159, pp. 2696–2712, 2008.
[31] G. Feng, "A survey on analysis and design of model-based fuzzy control systems," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 5, pp. 676–697, 2006.

[32] G. Becker, A. Packard, D. Philbrick, and G. Balas, “Control of parametrically dependent linear systems: a single quadratic lyapunov approach,” *Proc. ACC-93,* San Francisco, pp. 2795–2799, 1993.

[33] T. Hu and Z. Lin, “Control systems with actuator saturation: Analysis and design,” Vol. xvi (392p). Boston: Birkhäuser.

[34] Q. Zhao and J. Jiang, “Reliable state feedback control system design against actuator failures,” *Automatica*, vol. 34(10), pp. 1267–1272, 1998.

[35] G. H. Yang, J. L. Wang, and Y. C. Soh, “Reliable $\mathcal{H}_\infty$ controller design for linear systems,” *Automatica*, vol. 37(5), pp. 717–725, 2001.

[36] F. Liao, J. L. Wang, and G. H. Yang, “Reliable robust flight tracking control: An LMI approach,” *IEEE Trans. Control Syst. Technol.*, vol. 10(1), pp. 76–89, 2002.

[37] H. N. Wu and H. Y. Zhang, “Reliable $\mathcal{H}_\infty$ fuzzy control for continuous-time nonlinear systems with actuator failures,” *IEEE Trans. Fuzzy Syst.*, vol. 14(5), pp. 600–618, 2006.

[38] Y. Nesterov and A. Nemirovski, “Interior point polynomial methods in convex programming theory and applicqations,” Philadelpia, PA, SIAM, 1994.

[39] S. Boyd, L. E. Chauoi, E. Feron, and V. Balakrishan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia, PA:SIAM, 1994.

[40] L. E. Gahoui, F. Oustry, and H. Levert, “Robust solutions to uncertain semidefinite programs,” *SIAM Journal of Control and Optimization*, vol. 9(1), pp. 33-52, 1998.

[41] K. Tanaka, T. Ikeda, and H. O. Wang, “Fuzzy regulators and fuzzy observers: relaxed stability conditins and LMI-based designs,” *IEEE Trans. Fuzzy Syst.,* vol. 6(2), pp. 250–265, 1998.

[42] E. Kim and H. Lee, “New approaches to relaxed quadratic stablity conditin of fuzzy control systems,” *IEEE Trans. Fuzzy Syst.*, vol. 8(5), pp. 523–534, 2000.

[43] L. Xiaodong and Z. Qingling, “New approaches to $\mathcal{H}_\infty$ controller designs based on fuzzy observers for T-S fuzzy systems via LMI,” *Automatica*, vol. 39, pp. 1571–1582, 2003.

[44] H. D. Tuan, P. Apkarian, T. Narikiyo, and Y. Yamamoto, “Parameterized linear matrix inequality techniques in fuzzy control system design,” *IEEE Trans. Fuzzy Syst.*, vol. 9(2), pp. 324–332, 2001.

[45] M.C.M. Teixeira, E. Assuncão, and R.G. Avellar, “On relaxed LMI-based designs for fuzzy regulators and fuzzy observers,” *IEEE Trans. Fuzzy Syst.*, vol.11 pp.613–623, 2003.

[46] A. Sala and C. Ariño , “Relaxed stability and performance conditions for Takagi-Sugeno fuzzy systems with knowledge on membership function overlap,” *IEEE Trans. Syst., Man., Cybern. B, Cybern.*, vol. 37(3), pp. 727–732, 2007.

[47] C.-H. Fang, Y.-S. Liu, S.-W. Kau, L. Hong, and C.-H. Lee, “A new LMI-based approach to relaxed quadratic stabilization of T-S fuzzy control systems,” *IEEE Trasn. Fuzzy Syst.*, vol. 14(3), pp. 386–397, 2006.

[48] M. C. de Oliveira, R. E. Skelton, *Stability tests for constrained linear systems,* in Perspectives in Robust Control, S. O. Reza Moheimani, Editor. New York: Springer-Verlag, 2001, vol. 268, series Lecture Notes in Control and Information Sciences, pp. 241–257.

[49] L. Fang, H. Lin, and P. J. Antsaklis, “Stabilization and performance analysis for a class of switched systems,” in Proceedings of the 43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas, Dec. 2004, pp. 3265–3270.

[50] D. K. Kim, P.-G. Park and J. W. Ko, “Output-feedback $\mathcal{H}_\infty$ control of systems over communication networks using a deterministic switching system approach,” *Automatica*, 40, pp. 1205–1212, 2004.

4

Digital Stabilization of Fuzzy Systems with Time-Delay and Its Application to Backing up Control of a Truck-Trailer

Chang-Woo Park
Korea Electronics Technology Institute (KETI),
401-402 B/D 193,Yakdae-Dong, Wonmi-Gu, Puchon-Si, Kyunggi-Do, 420-734, Korea

1. Introduction

The control problems for delayed systems have attention over the last few decades since the time-delay is frequently a source of instability and encountered in various engineering systems. Extensive research has already been done in the conventional control to find the solutions [1][2]. However, for fuzzy control systems, there are few studies on the stabilization problem for especially systems with time-delay[3][4]. A linear controller like PID controllers has a short time-delay in calculating the output since its algorithm is so simple. However, in the case of a complex algorithm like fuzzy or neural networks, a considerable time-delay can occur because so many calculations are needed to get the output. Nevertheless, the most conventional discrete time fuzzy controllers are the ideal controllers in which the time-delay is not considered. Recently, to deal with the time-delay, the design methods of the fuzzy control systems with higher order have been proposed in [5]. However the structure of the control system is very complex because the design of higher order fuzzy rule-base is highly difficult.

In this chapter, the digital fuzzy control system considering a time delay is developed and its stability analysis and design method are proposed. We use the discrete Takagi-Sugeno(TS) fuzzy model and parallel distributed compensation(PDC) conception for the controller[6-9]. And we follow the linear matrix inequality(LMI) approach to formulate and solve the problem of stabilization for the fuzzy controlled systems with time-delay. The analysis and the design of the discrete time fuzzy control systems by LMI theory are considered in [10-12].

If the system has a considerable time-delay the analysis and the design of the controller are very difficult since the time-delay makes the output of the controller not synchronized with the sampling time. We propose the PDC-type fuzzy feedback controller whose output is delayed with unit sampling period and predicted using current states and the control input to the plant at previous sampling time. The analysis and the design of the controller are very easy because the output of the proposed controller is synchronized with the sampling time. Therefore, the proposed control system can be designed using the conventional methods for stabilizing the discrete time fuzzy systems and the feedback gains of the controller can be obtained using the concept of the LMI feasibility problem.

The proposed DFC is applied to backing up control of a computer-simulated truck-trailer with time-delay to verify the validity and the effectiveness of the control scheme. Note that the term "Digital fuzzy control system" is used to emphasize the proposed aspect corresponding to the existing "Discrete time fuzzy system".

2. Discrete TS model based fuzzy control

In the discrete time TS fuzzy systems without control input, the dynamic properties of each subspace can be expressed as the following fuzzy IF-THEN rules[6].

$$\begin{array}{c}\text{Rule } i: \text{ If } x_1(k) \text{ is } M_{i1} \cdots \text{and } x_n(k) \text{ is } M_{in} \\ \text{THEN } \mathbf{x}(k+1) = \mathbf{G}_i \mathbf{x}(k)\end{array} \quad i = 1, 2, \cdots, r \tag{1}$$

where $\mathbf{x}(k) = [x_1(k) \quad x_2(k) \quad \cdots \quad x_n(k)]^T \in \Re^n$ denotes the state vector of the fuzzy system, r is the number of the IF-THEN rules, and M_{ij} is fuzzy set.
If the state $\mathbf{x}(k)$ is given, the output of the fuzzy system expressed as the fuzzy rules of Eq. (1) can be inferred as follows.

$$\mathbf{x}(k+1) = \frac{\sum_{i=1}^{r} w_i(k)\, \mathbf{G}_i \mathbf{x}(k)}{\sum_{i=1}^{r} w_i(k)} = \sum_{i=1}^{r} h_i(k)\, \mathbf{G}_i \mathbf{x}(k) \tag{2}$$

where $w_i(k) = \prod_{j=1}^{n} M_{ij}(x_j(k))$, $h_i(k) = \dfrac{w_i(k)}{\sum_{i=1}^{r} w_i(k)}$

A sufficient condition for ensuring the stability of the fuzzy system(2) is given in **Theorem 1**.
Theorem 1: The equilibrium point for the discrete time fuzzy system (2) is asymptotically stable in the large if there exists a common positive definite matrix **P** satisfying the following inequalities.

$$\mathbf{G}_i^T \mathbf{P} \mathbf{G}_i - \mathbf{P} < \mathbf{0}\ ,\ i = 1, 2, \cdots, r \tag{3}$$

Proof: The proof can be given in [7].
In the discrete time fuzzy system with control input to the plant, the dynamic properties of each subspace can be expressed as the following fuzzy IF-THEN rules.

$$\begin{array}{c}\text{Rule } i: \text{ If } x_1(k) \text{ is } M_{i1} \cdots \text{and } x_n(k) \text{ is } M_{in} \\ \text{THEN } \mathbf{x}(k+1) = \mathbf{A}_i \mathbf{x}(k) + \mathbf{B}_i \mathbf{u}(k)\end{array} \quad i = 1, 2, \cdots, r \tag{4}$$

where

$\mathbf{x}(k) = [x_1(k) \quad x_2(k) \quad \cdots \quad x_n(k)]^T \in \Re^n$ denotes the state vector of the fuzzy system.

$\mathbf{u}(k) = [u_1(k) \quad u_2(k) \quad \cdots \quad u_m(k)]^T \in \Re^m$ denotes the input of the fuzzy system.

r is the number of the fuzzy IF-THEN rules, and M_{ij} is the fuzzy set.

If the set of $(\mathbf{x}(k),\mathbf{u}(k))$ is given the output of the fuzzy system (4) can be obtained as follows.

$$\mathbf{x}(k+1)=\frac{\sum_{i=1}^{r} w_i(k)\{\mathbf{A}_i\mathbf{x}(k)+\mathbf{B}_i\mathbf{u}(k)\}}{\sum_{i=1}^{r} w_i(k)}=\sum_{i=1}^{r} h_i(k)\{\mathbf{A}_i\mathbf{x}(k)+\mathbf{B}_i\mathbf{u}(k)\} \tag{5}$$

where

$w_i(k)=\prod_{j=1}^{n} M_{ij}(x_j(k))$, and $h_i(k)=\dfrac{w_i(k)}{\sum_{i=1}^{r} w_i(k)}$.

In PDC, the fuzzy controller is designed distributively according to the corresponding rule of the plant[9]. Therefore, the PDC for the plant (4) can be expressed as follows.

$$\begin{array}{l}\text{Rule } j: \text{ If } x_1(k) \text{ is } M_{j1} \cdots \text{and } x_n(k) \text{ is } M_{jn} \\ \qquad \text{THEN } \mathbf{u}(k)=-\mathbf{F}_j\mathbf{x}(k)\end{array} \qquad j=1,2,\cdots,r \tag{6}$$

The fuzzy controller output of Eq. (6) can be inferred as follows.

$$\mathbf{u}(k)=-\frac{\sum_{i=1}^{r} w_i(k)\,\mathbf{F}_i\mathbf{x}(k)}{\sum_{i=1}^{r} w_i(k)}=-\sum_{j=1}^{r} h_j(k)\,\mathbf{F}_j\mathbf{x}(k) \tag{7}$$

where $h_j(k)$ is the same function in Eq. (5).

Substituting Eq. (7) into Eq. (5) gives the following closed loop discrete time fuzzy system.

$$\mathbf{x}(k+1)=\sum_{i=1}^{r} h_i(k)\{\mathbf{A}_i\mathbf{x}(k)-\mathbf{B}_i\sum_{j=1}^{r} h_j(k)\mathbf{F}_j\mathbf{x}(k)\}=\sum_{i=1}^{r}\sum_{j=1}^{r} h_i(k)\,h_j(k)\{\mathbf{A}_i-\mathbf{B}_i\mathbf{F}_j\}\mathbf{x}(k) \tag{8}$$

Defining $\mathbf{G}_{ij}=\mathbf{A}_i-\mathbf{B}_i\mathbf{F}_j$, the following equation is obtained.

$$\mathbf{x}(k+1)=\sum_{i=1}^{r} h_i(k)\,h_i(k)\,\mathbf{G}_{ii}\mathbf{x}(k)+2\sum_{i<j}^{r} h_i(k)\,h_j(k)\,\{\frac{\mathbf{G}_{ij}+\mathbf{G}_{ji}}{2}\}\mathbf{x}(k) \tag{9}$$

Applying **Theorem 1** to analyze the stability of the discrete time fuzzy system (9), the stability condition of **Theorem 2** can be obtained.

Theorem 2 : The equilibrium point of the closed loop discrete time fuzzy system (9) is asymptotically stable in the large if there exists a common positive definite matrix P which satisfies the following inequalities for all i and j except the set (i,j) satisfying $h_i(k)\cdot h_j(k)=0$.

$$\mathbf{G}_{ii}^T \mathbf{P} \mathbf{G}_{ii} - \mathbf{P} < \mathbf{0} \tag{10a}$$

$$(\frac{\mathbf{G}_{ij} + \mathbf{G}_{ji}}{2})^T \mathbf{P} (\frac{\mathbf{G}_{ij} + \mathbf{G}_{ji}}{2}) - \mathbf{P} \leq \mathbf{0}, \quad i < j \tag{10b}$$

Proof : The proof can be given in [7].

If $\mathbf{B} = \mathbf{B}_1 = \mathbf{B}_2 = \cdots = \mathbf{B}_r$ in the plant (5) is satisfied, the closed loop system (8) can be obtained as follows.

$$\mathbf{x}(k+1) = \sum_{i=1}^{r} h_i(k)\{\mathbf{A}_i \mathbf{x}(k) - \mathbf{B}\sum_{j=1}^{r} h_j(k)\mathbf{F}_j \mathbf{x}(k)\} = \sum_{i=1}^{r} h_i(k)\{\mathbf{A}_i - \mathbf{B}\mathbf{F}_i\}\mathbf{x}(k) = \sum_{i=1}^{r} h_i(k)\mathbf{G}_i \mathbf{x}(k) \tag{11}$$

where $\mathbf{G}_i = \mathbf{A}_i - \mathbf{B}\mathbf{F}_i$

Hence, **Theorem 1** can be applied to the stability analysis of the closed loop system (11).

3. LMI approach for fuzzy system design

To prove the stability of the discrete time fuzzy control system by **Theorem 1** and **Theorem 2**, the common positive definite matrix **P** must be solved. LMI theory can be applied to solving **P** [13]. LMI theory is one of the numerical optimization techniques. Many of the control problems can be transformed into LMI problems and the recently developed Interior-point method can be applied to solving numerically the optimal solution of these LMI problems[14].

Definition 1: Linear matrix inequility can be defined as follows.

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^{m} x_i \mathbf{F}_i > \mathbf{0} \tag{12}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix}^T$ is the parameter, the symmetric matrices $\mathbf{F}_i = \mathbf{F}_i^T \in \Re^{n \times n}, i = 0, \cdots, m$ are given, and the inequality symol " $> \mathbf{0}$ " means that $\mathbf{F}(\mathbf{x})$ is the positive definite matrix.

LMI of Eq. (12) means the convex constraints for **x**. Convex constraint problems for the various **x** can be expressed as LMI of Eq. (12). LMI feasibility problem can be described as follows.

LMI feasibility problem: The problem of finding $\mathbf{x}^{\text{feasp}}$ which satisfies $\mathbf{F}(\mathbf{x}^{\text{feasp}}) > \mathbf{0}$ or proving the unfeasibility in the case that LMI $\mathbf{F}(\mathbf{x}) > \mathbf{0}$ is given.

And the stability condition of **Theorem 1** can be transformed into the LMI feasibility problem as follows.

LMI feasibility problem about the stability condition of Theorem 1 : The problem of finding $\mathbf{P}$ which satisfies the LMI's, $\mathbf{P} > \mathbf{0}$ and $\mathbf{G}_i^T \mathbf{P} \mathbf{G}_i - \mathbf{P} < \mathbf{0}$, $i = 1, 2, \cdots, r$ or proving the unfeasibility in the case that $\mathbf{A}_i \in \Re^{n \times n}$, $i = 1, 2, \cdots, r$ are given.

If the design object of a controller is to guarantee the stability of the closed loop system (5), the design of the PDC fuzzy controller(7) is equivalent to solving the following LMI feasibility problem using Schur complements[13].

LMI feasibility problem equivalent to the PDC design problem *(Case I)* : The problem of finding $\mathbf{X} > \mathbf{0}$ and $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_r$ which satisfy the following inequalities.

$$\begin{bmatrix} \mathbf{X} & \{\mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_i\}^T \\ \mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_i & \mathbf{X} \end{bmatrix} > \mathbf{0}, \quad i = 1, 2, \cdots, r$$

$$\begin{bmatrix} \mathbf{X} & 1/2\{\mathbf{A}_i\mathbf{X} + \mathbf{A}_j\mathbf{X} - \mathbf{B}_i\mathbf{M}_i - \mathbf{B}_j\mathbf{M}_j\}^T \\ 1/2\{\mathbf{A}_i\mathbf{X} + \mathbf{A}_j\mathbf{X} - \mathbf{B}_i\mathbf{M}_i - \mathbf{B}_j\mathbf{M}_j\} & \mathbf{X} \end{bmatrix} > \mathbf{0}, \quad i < j$$

where $\mathbf{X} = \mathbf{P}^{-1}$, $\mathbf{M}_1 = \mathbf{F}_1\mathbf{X}$, $\mathbf{M}_2 = \mathbf{F}_2\mathbf{X}$, $\cdots$, and $\mathbf{M}_r = \mathbf{F}_r\mathbf{X}$.
The feedback gain matrices $\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_r$ and the common positive definite matrix $\mathbf{P}$ can be given by the LMI solutions, $\mathbf{X}$ and $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_r$, as follows.

$$\mathbf{P} = \mathbf{X}^{-1}, \ \mathbf{F}_1 = \mathbf{M}_1\mathbf{X}^{-1}, \ \mathbf{F}_2 = \mathbf{M}_2\mathbf{X}^{-1}, \ \cdots, \text{and } \mathbf{F}_r = \mathbf{M}_r\mathbf{X}^{-1}$$

If $\mathbf{B} = \mathbf{B}_1 = \mathbf{B}_2 = \cdots = \mathbf{B}_r$ is satisfied, the design of the PDC fuzzy controller(7) is equivalent to solving the following LMI feasibility problem.
LMI feasibility problem equivalent to the PDC design problem *(Case II)* : The problem of finding $\mathbf{X} > \mathbf{0}$ and $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_r$ which satisfy the following equations.

$$\begin{bmatrix} \mathbf{X} & \{\mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_i\}^T \\ \mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_i & \mathbf{X} \end{bmatrix} > \mathbf{0} \qquad i = 1, 2, \cdots, r$$

where $\mathbf{X} = \mathbf{P}^{-1}$, $\mathbf{M}_1 = \mathbf{F}_1\mathbf{X}$, $\mathbf{M}_2 = \mathbf{F}_2\mathbf{X}$, $\cdots$, and $\mathbf{M}_r = \mathbf{F}_r\mathbf{X}$.
The feedback gain matrices $\mathbf{F}_1, \mathbf{F}_2, \cdots, \mathbf{F}_r$ and the common positive definite matrix $\mathbf{P}$ can be given by the LMI solutions, $\mathbf{X}$ and $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_r$, as follows.

$$\mathbf{P} = \mathbf{X}^{-1}, \ \mathbf{F}_1 = \mathbf{M}_1\mathbf{X}^{-1}, \ \mathbf{F}_2 = \mathbf{M}_2\mathbf{X}^{-1}, \ \cdots, \text{and } \mathbf{F}_r = \mathbf{M}_r\mathbf{X}^{-1}$$

4. Digital fuzzy control system considering time-delay

In a real control system, a considerable time-delay can occur due to a sensor and a controller. Let τ be defined as the sum of all this time-delay. In the case of the real system, the ideal fuzzy controller of Eq. (6) can be described as follows due to the time-delay.

$$\begin{aligned} &\text{Rule } j: \text{ If } x_1(kT) \text{ is } M_{j1} \cdots \text{and } x_n(kT) \text{ is } M_{jn} \\ &\qquad \text{THEN } \mathbf{u}(kT + \tau) = -\mathbf{F}_j\mathbf{x}(kT) \end{aligned} \qquad j = 1, 2, \cdots, r \tag{13}$$

Because the time-delay makes the output of controller not synchronized with the sampling time, **Theorem 1** can not be applied to this system. Therefore the analysis and the design of the controller are very difficult. In this chapter, DFC which has the following fuzzy rules is proposed to consider the time-delay of the fuzzy plant (4).

$$\begin{array}{l}\text{Rule } j: \text{ If } x_1(k) \text{ is } M_{j1} \cdots \text{and } x_n(k) \text{ is } M_{jn} \\ \text{THEN } \mathbf{u}(k+1) = \mathbf{D}_j\mathbf{u}(k) + \mathbf{E}_j\mathbf{x}(k)\end{array} \qquad j = 1, 2, \cdots, r \tag{14}$$

The output of DFC (14) is inferred as follows.

$$\mathbf{u}(k+1) = \frac{\sum_{j=1}^{r} w_j(k)\{\mathbf{D}_j\mathbf{u}(k) + \mathbf{E}_j\mathbf{x}(k)\}}{\sum_{j=1}^{r} w_j(k)}$$

$$= \sum_{j=1}^{r} h_j(k)\{\mathbf{D}_j\mathbf{u}(k) + \mathbf{E}_j\mathbf{x}(k)\} \tag{15}$$

The general timing diagram of fuzzy control loop is shown in Fig. 1. T is the sampling period of the control loop, τ_v and τ_c are the delay made by sensor system and fuzzy controller respectively. Therefore the output of the controller is applied to the plant after overall dealy $\tau = \tau_v + \tau_c$.

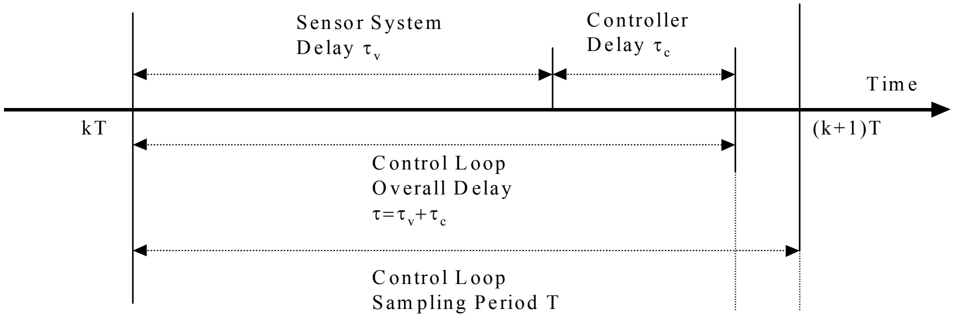


Fig. 1. Timing Diagram of the Fuzzy Control Loop

The output timing of a ideal controller, a delayed controller, and the proposed controller is shown in the Fig. 2. In the ideal controller, it is assumed that there is no time-delay. If this controller is implemented in real systems the time-delay τ is added like Eq. (13). The analysis and the design of this system with delayed controller are very difficult since the output of controller is not syncronized with the sampling time.

On the other hand, the analysis and the design of the proposed controller are very easy because the controller output is syncronized with the sampling time delayed with unit sampling period. Using this proposed controller, we can realize a control algorithm during the time interval $T - \tau_v$ in Fig. 1. In this time interval, a complex algorithm such as not only fuzzy algorithm but also nonlinear control algorithm can be sufficiently realized in real time.

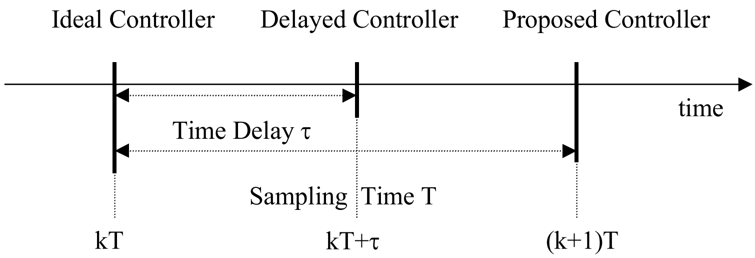


Fig. 2. Output Timing of the Controllers (three cases)

Combining the fuzzy plant (5) with the DFC (15), the closed loop system is given as follows.

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \mathbf{u}(k+1) \end{bmatrix} = \sum_{i=1}^{r} h_i(k) \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{E}_i & \mathbf{D}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix} \tag{16}$$

Defining the new state vector as $\mathbf{w}(k) = \begin{bmatrix} \mathbf{x}(k) \\ \mathbf{u}(k) \end{bmatrix}$, the closed loop system (16) can be modified as

$$\mathbf{w}(k+1) = \sum_{i=1}^{r} h_i(k) \mathbf{G}_i \mathbf{w}(k) \tag{17}$$

where $\mathbf{G}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{E}_i & \mathbf{D}_i \end{bmatrix}$

Hence, the stability condition of the closed loop system (17) becomes the same as the sufficient condition of **Theorem 1** and the stability can be determined by solving ***LMI feasibility problem about the stability condition of Theorem 1***. Also, the design problem of the DFC guaranteeing the stability of the closed loop system can be transformed into ***LMI feasibility problem***. To do this, the design problem of the DFC is transformed into the design problem of the PDC fuzzy controller.

PDC design problem equivalent to DFC design problem:

The problem of designing the PDC fuzzy controller $\mathbf{v}(k) = -\sum_{j=1}^{r} h_j(k) \overline{\mathbf{F}}_j \mathbf{w}(k)$ in the case that the fuzzy plant $\mathbf{w}(k+1) = \sum_{i=1}^{r} h_i(k) \{\overline{\mathbf{A}}_i \mathbf{w}(k) + \overline{\mathbf{B}} \mathbf{v}(k)\}$ is given.

where $\overline{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, $\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$, and $\overline{\mathbf{F}}_j = -\begin{bmatrix} \mathbf{E}_j & \mathbf{D}_j \end{bmatrix}$

Therefore, using the same notation in section 3, the design problem of the DFC can be equivalent to the following **LMI feasibility problem**.

LMI feasibility problem equivalent to DFC design problem:
The problem of finding $\mathbf{X} > \mathbf{0}$ and $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_r$ which satisfy following equation.

$$\begin{bmatrix} \mathbf{X} & \{\overline{\mathbf{A}}_i \mathbf{X} - \overline{\mathbf{B}}\ \mathbf{M}_i\}^T \\ \overline{\mathbf{A}}_i \mathbf{X} - \overline{\mathbf{B}}\ \mathbf{M}_i & \mathbf{X} \end{bmatrix} > \mathbf{0},\ i = 1, 2, \cdots, r$$

where $\mathbf{X} = \mathbf{P}^{-1}$, $\mathbf{M}_1 = \overline{\mathbf{F}}_1 \mathbf{X}$, $\mathbf{M}_2 = \overline{\mathbf{F}}_2 \mathbf{X}$, $\cdots$, and $\mathbf{M}_r = \overline{\mathbf{F}}_r \mathbf{X}$
The feedback gain matrices $\overline{\mathbf{F}}_1, \overline{\mathbf{F}}_2, \cdots, \overline{\mathbf{F}}_r$ and the common positive definite matrix $\mathbf{P}$ can be given by the LMI solutions, $\mathbf{X}$ and $\mathbf{M}_1, \mathbf{M}_2, \cdots, \mathbf{M}_r$, as follows.

$$\mathbf{P} = \mathbf{X}^{-1},\ \overline{\mathbf{F}}_1 = \mathbf{M}_1 \mathbf{X}^{-1},\ \overline{\mathbf{F}}_2 = \mathbf{M}_2 \mathbf{X}^{-1},\ \cdots,\ \overline{\mathbf{F}}_r = \mathbf{M}_r \mathbf{X}^{-1} \tag{18}$$

Therefore, the control gain matrices $\mathbf{D}_1, \cdots, \mathbf{D}_r, \mathbf{E}_1, \cdots, \mathbf{E}_r$ of the proposed DFC can be obtained from the feedback gain matrices $\overline{\mathbf{F}}_1, \overline{\mathbf{F}}_2, \cdots, \overline{\mathbf{F}}_r$.

5. Backing up control of computer-simulated truck-trailer

We have shown an analysis technique of the proposed DFC under the condition that time-delay exists in section 4. Some papers have reported that backing up control of a computer-simulated truck-trailer could be realized by fuzzy control[9][11][15][16]. However, these studies have not analyzed the time-delay effect to the control system. In this section, we apply the controller to backing up control of a truck-trailer system with time-delay.

5.1 Models of a truck-trailer

M. Tokunaga derived the following model about the truck-trailer system [16]. Fig. 3 shows the schematic diagram of this system.

$$x_0(k+1) = x_0(k) + vT / l \tan[u(k)]$$

$$x_1(k) = x_0(k) - x_2(k)$$

$$x_2(k+1) = x_2(k) + vT / L \sin[x_1(k)]$$

$$x_3(k+1) = x_3(k) + vT \cos[x_1(k)] \sin[\{x_2(k+1) + x_2(k)\}/2]$$

$$x_4(k+1) = x_4(k) + vT \cos[x_1(k)] \cos[\{x_2(k+1) + x_2(k)\}/2] \tag{19}$$

where

$x_0(k)$: The angle of the truck referenced to the desired trajectory
$x_1(k)$: The angle difference between the truck and the trailer
$x_2(k)$: The angle of the trailer referenced to the desired trajectory
$x_3(k)$: The vertical position of the trailer tail end
$x_4(k)$: The horizontal position of the trailer tail end

$u(k)$: The steering angle of the truck

l : The length of the truck, L : The length of the trailer

T : Sampling time, v : The constant backward speed

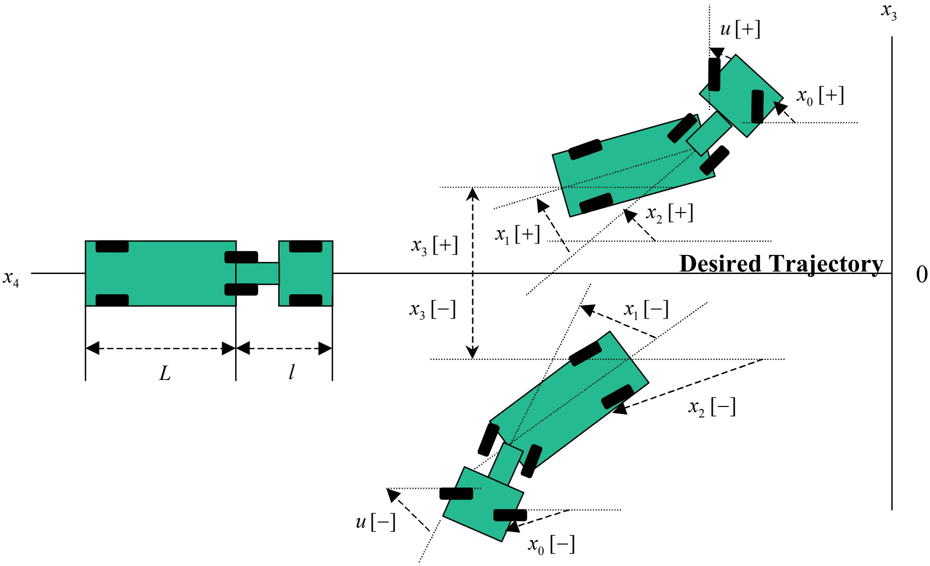


Fig. 3. Truck Trailer Model and Its Coordinate System

K. Tanaka defined the state vector as $\mathbf{x}(k) = [x_1(k) \quad x_2(k) \quad x_3(k)]^T$ in the truck-trailer model (19) and expressed the plant as two following fuzzy rules[9].

$$\begin{aligned}
&\text{Rule 1: If } x_2(k) + vT/\{2L\}\, x_1(k) \text{ is } M_1 \\
&\qquad \text{THEN } \mathbf{x}(k+1) = \mathbf{A}_1\mathbf{x}(k) + \mathbf{B}_1 u(k) \\
&\text{Rule 2: If } x_2(k) + vT/\{2L\}\, x_1(k) \text{ is } M_2 \\
&\qquad \text{THEN } \mathbf{x}(k+1) = \mathbf{A}_2\mathbf{x}(k) + \mathbf{B}_2 u(k)
\end{aligned} \tag{20}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} 1-\frac{vT}{L} & 0 & 0 \\ \frac{vT}{L} & 1 & 0 \\ \frac{v^2T^2}{2L} & vT & 1 \end{bmatrix},\ \mathbf{A}_2 = \begin{bmatrix} 1-\frac{vT}{L} & 0 & 0 \\ \frac{vT}{L} & 1 & 0 \\ \frac{dv^2T^2}{2L} & dvT & 1 \end{bmatrix},\ \mathbf{B} = \mathbf{B}_1 = \mathbf{B}_2 = \begin{bmatrix} \frac{vT}{l} \\ 0 \\ 0 \end{bmatrix}$$

$l = 2.8\,[\text{m}]$, $L = 5.5\,[\text{m}]$, $v = -1.0\,[\text{m/s}]$, $T = 2.0\,[\text{s}]$, $d = 10^{-2}/\pi$

Fig. 4 shows the membership function of the premise part in the fuzzy system (20).

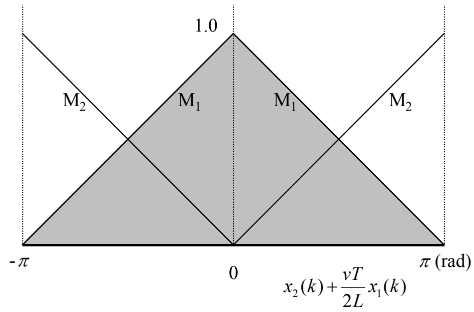


Fig. 4. Membership function

5.2 Discrete time fuzzy controller applied to the system with no time-delay

In this subsection, backing up control of a truck-trailer is simulated by the conventional discrete time fuzzy controller under the assumption that no time-delay exists.
To solve the backward parking problem of Eq. (20), the PDC fuzzy controller can be designed as follows.

$$\begin{aligned}&\text{Rule 1: If } x_2(k)+vT/\{2L\}\cdot x_1(k) \text{ is } M_1\\ &\qquad \text{THEN } u(k)=\mathbf{F}_1^T\mathbf{x}(k)\\ &\text{Rule 2: If } x_2(k)+vT/\{2L\}\cdot x_1(k) \text{ is } M_2\\ &\qquad \text{THEN } u(k)=\mathbf{F}_2^T\mathbf{x}(k)\end{aligned} \tag{21}$$

where $\mathbf{F}_1=\begin{bmatrix}1.2837\\-0.4139\\0.0201\end{bmatrix}$ and $\mathbf{F}_2=\begin{bmatrix}0.9773\\-0.0709\\0.0005\end{bmatrix}$.

Ricatti equation for linear discrete systems was used to determine these feedback gains. The detailed derivation of these feedback gains was given in [9].
Substituting Eq. (21) into Eq. (20) yields the following closed loop system due to $\mathbf{B}=\mathbf{B}_1=\mathbf{B}_2$.

$$\mathbf{x}(k+1)=\sum_{i=1}^{2}h_i(k)\mathbf{G}_i\mathbf{x}(k) \tag{22}$$

where

$$\mathbf{G}_1=\begin{bmatrix}0.448 & 0.296 & -0.014\\ -0.364 & 1 & 0\\ 0.364 & -2 & 1\end{bmatrix} \text{ and } \mathbf{G}_2=\begin{bmatrix}0.448 & 0.296 & -0.014\\ -0.364 & 1 & 0\\ 0.116\times10^{-2} & -0.637\times10^{-2} & 1\end{bmatrix}.$$

Since there exists the common positive matrix **P** which satisfies the stability sufficient condition (3), the closed loop system is asymptotically stable in the large. That is, the backward parking can be accomplished for all initial contitions.

$$\text{Common positive definite matrix : } \mathbf{P} = \begin{bmatrix} 113.9 & -92.61 & 2.540 \\ -92.61 & 110.7 & -3.038 \\ 2.540 & -3.038 & 0.5503 \end{bmatrix}$$

Two initial conditions used for the simulations of the truck-trailer system are given in Table 1.

CASE	$x_1(0)$[deg]	$x_2(0)$[deg]	$x_3(0)$[m]
CASE I	0	0	20
CASE II	-90	135	-10

Table 1: The initial conditions of the truck-trailer system

Fig. 5(a) and (b) show the simulation results for CASE I and CASE II. As can be seen in these Figures, the backing up control for each initial condition is accomplished effectively.

5.3 Discrete time fuzzy controller applied to the system with time-delay

In many cases, vision sensor is generally needed to measure the state $\mathbf{x}(k)$ of the truck-trailer system[17]. The time-delay can be made by the vision sensor in the transferring of image and the image processing. Also, it can be made by the digital hardware in the calculation of the fuzzy algorithm and by the actutor in adjusting the steering angle. Let τ be defined as the sum of all this time-delay. In the case of the real system, the ideal fuzzy controller of Eq. (21) can be described as follows due to the time-delay.

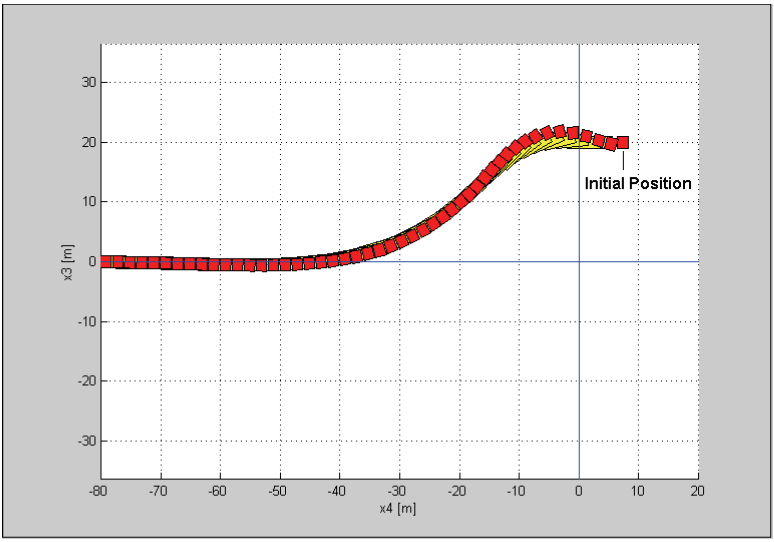


Fig. 5. (a). Simulation result for CASE I

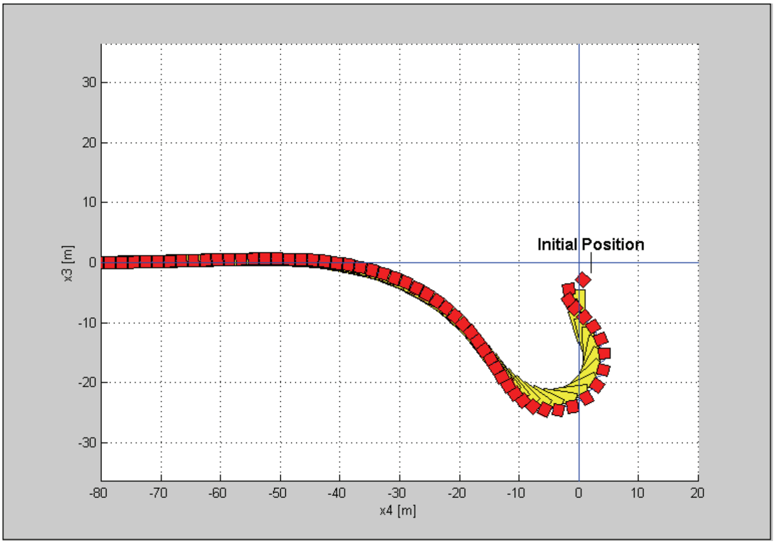


Fig. 5. (b). Simulation result for CASE II

$$\begin{aligned}&\text{Rule 1: If } x_2(kT) + vT/\{2L\} \cdot x_1(kT) \text{ is } M_1\\&\qquad \text{THEN } u(kT+\tau) = \mathbf{F}_1^T \mathbf{x}(kT)\\&\text{Rule 2: If } x_2(kT) + vT/\{2L\} \cdot x_1(kT) \text{ is } M_2\\&\qquad \text{THEN } u(kT+\tau) = \mathbf{F}_2^T \mathbf{x}(kT)\end{aligned} \tag{23}$$

The simulation is executed in the case that the time-dealy τ is a half of the sampling time (τ =1 [sec]). Fig. 6 (a) and (b) show that the truck-trailer system is oscillating and the fuzzy controller can not accomplish the backing up control effectively.

5.4 Proposed digital fuzzy controller applied to the system with time-delay

In this subsection, we design the DFC considering time-delay. Following the design technique of DFC in section 4, we can construct the DFC for the backing up control problem as follows.

$$\begin{aligned}&\text{Rule 1: If } x_2(k) + vT/\{2L\} \cdot x_1(k) \text{ is } M_1\\&\qquad \text{THEN } u(k+1) = \mathbf{D}_1 u(k) + \mathbf{E}_1 \mathbf{x}(k)\\&\text{Rule 2: If } x_2(k) + vT/\{2L\} \cdot x_1(k) \text{ is } M_2\\&\qquad \text{THEN } u(k+1) = \mathbf{D}_2 u(k) + \mathbf{E}_2 \mathbf{x}(k)\end{aligned} \tag{24}$$

Combining Eq. (20) with Eq. (24), the augmented closed loop system is given as follows.

$$\mathbf{w}(k+1) = \sum_{i=1}^{2} h_i(k)\mathbf{G}_i \mathbf{w}(k) \tag{25}$$

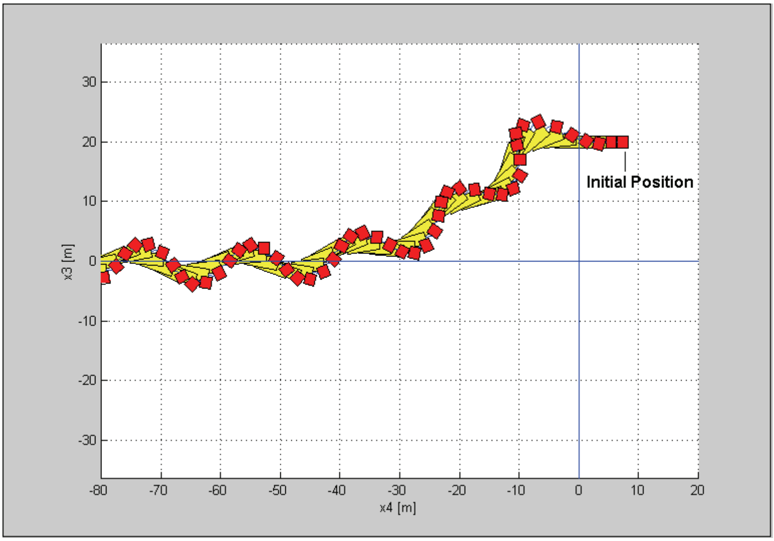


Fig. 6. (a). Simulation result for CASE I ($\tau = 1$)

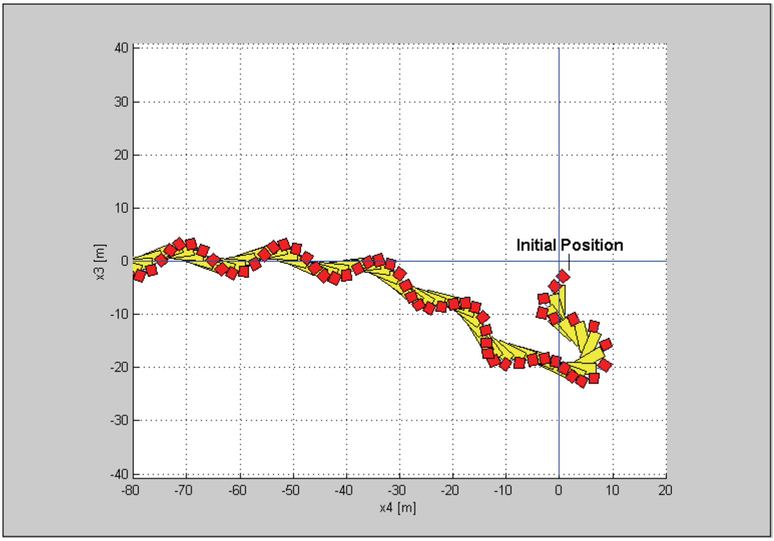


Fig. 6. (b). Simulation result for CASE II ($\tau = 1$)

where $\mathbf{G}_1 = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ \mathbf{E}_1 & \mathbf{D}_1 \end{bmatrix}$, $\mathbf{G}_2 = \begin{bmatrix} \mathbf{A}_2 & \mathbf{B}_2 \\ \mathbf{E}_2 & \mathbf{D}_2 \end{bmatrix}$

To obtain the control gain matrices $\mathbf{D}_1, \mathbf{D}_2, \mathbf{E}_1, \mathbf{E}_2$ guaranteeing the stability of the closed loop system (25), we solve the ***LMI feasibility problem equivalent to DFC design problem*** as follows.

The problem of finding $\mathbf{X} > \mathbf{0}$ and $\mathbf{M}_1, \mathbf{M}_2$ which satisfy the following inequalities:

$$\begin{bmatrix} \mathbf{X} & \{\overline{\mathbf{A}}_i\mathbf{X} - \overline{\mathbf{B}}\ \mathbf{M}_i\}^T \\ \overline{\mathbf{A}}_i\mathbf{X} - \overline{\mathbf{B}}\ \mathbf{M}_i & \mathbf{X} \end{bmatrix} > \mathbf{0}$$

where $\overline{\mathbf{A}}_i = \begin{bmatrix} \mathbf{A}_i & \mathbf{B}_i \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ and $\overline{\mathbf{B}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$, $i = 1, 2$

The matrices $\mathbf{X}$ and $\mathbf{M}_1, \mathbf{M}_2$ in LMI's are determined using a convex optimization technique offered by [18].

$$\mathbf{X} = \begin{bmatrix} 157.0056 & 61.9680 & -1.6565 & 220.727 \\ 61.9680 & 50.4822 & 69.8423 & 53.4329 \\ -1.6565 & 69.8423 & 489.4416 & -2.3866 \\ 220.7727 & 53.4329 & -2.3866 & 442.6866 \end{bmatrix},$$

$$\mathbf{M}_1 = \begin{bmatrix} -96.3672 & -43.1521 & 41.8056 & -5.8356 \end{bmatrix},$$

$$\mathbf{M}_2 = \begin{bmatrix} -116.3143 & -66.0021 & 1.3065 & -22.9842 \end{bmatrix}$$

The feedback gains and a common positive definite matrix, $\mathbf{P}$ are determined by the relationship (18) as follows.

$$\mathbf{P} = \mathbf{X}^{-1} = \begin{bmatrix} 0.0995 & -0.1036 & 0.0149 & -0.0370 \\ -0.1036 & 0.1373 & -0.0198 & 0.0350 \\ 0.0149 & -0.0198 & 0.0049 & -0.0050 \\ -0.0370 & 0.0350 & -0.0050 & 0.0165 \end{bmatrix},$$

$$\overline{\mathbf{F}}_1 = \mathbf{M}_1\mathbf{X}^{-1} = -\begin{bmatrix} \mathbf{E}_1 & \mathbf{D}_1 \end{bmatrix} = \begin{bmatrix} -3.9047 & 2.6765 & -0.3020 & 1.5869 \end{bmatrix},$$

$$\overline{\mathbf{F}}_2 = \mathbf{M}_2\mathbf{X}^{-1} = -\begin{bmatrix} \mathbf{E}_2 & \mathbf{D}_2 \end{bmatrix} = \begin{bmatrix} -3.8624 & 2.1564 & -0.3102 & 1.6123 \end{bmatrix}$$

Therefore, the closed loop system is asymptotically stable in the large and the control gain matrices are given as follows by ***PDC design problem equivalent to DFC design problem***.

$$\mathbf{D}_1 = -1.5869,\ \mathbf{D}_2 = -1.6123,$$

$$\mathbf{E}_1 = \begin{bmatrix} 3.9047 & -2.6765 & 0.3020 \end{bmatrix},\ \mathbf{E}_2 = \begin{bmatrix} 3.8624 & -2.1564 & 0.3102 \end{bmatrix}$$

Fig. 7 (a) and (b) show the simulation results of the designed DFC. As can be seen in these figures, the backward parking is accomplished successfully for CASE I and CASE II although the considerable time-delay($\tau = 1$ [sec]) exists.

6. Conclusions

In this chapter, we have developed a DFC framework for a class of systems with time-delay. Because the proposed controller was syncronized with the sampling time delayed with unit

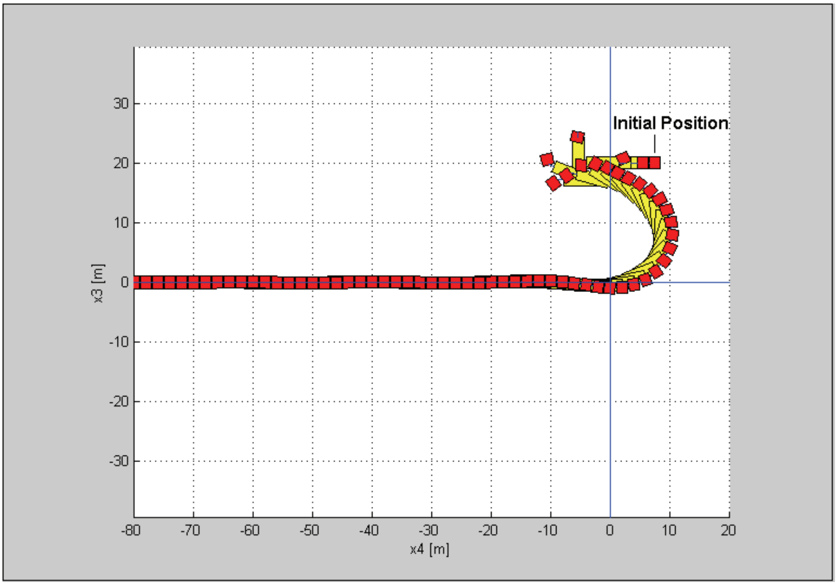


Fig. 7. (a). Simulation result by DFC for CASE I

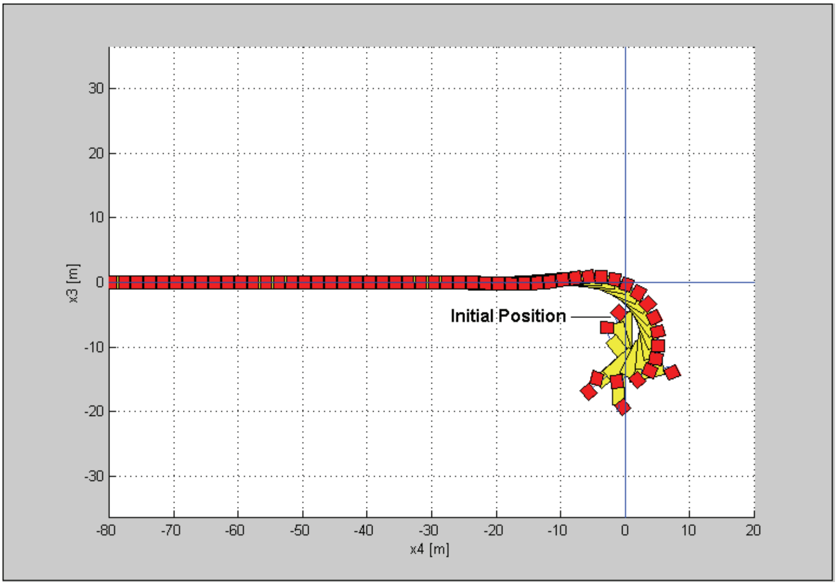


Fig. 7. (b). Simulation result by DFC for CASE II

sampling period and predicted, the analysis and the design problem considering time-delay could be very easy. Convex optimization technique based on LMI has been utilized to solve the problem of finding stable feedback gains and a common Lyapunov function. Therefore the stability of the system was guaranteed in the existence of time-delay and the real-time control processing could be possible. To show the effectiveness and feasibility of the proposed controller we have developed a digital fuzzy control system for backing up a computer-simulated truck-trailer with time-delay. Through the simulations, we have shown that the proposed DFC could achieve backing up control of a truck-trailer successfully although a considerable time-delay existed.

7. References

[1] Horowitz, I. : 'Some properties of delayed controls', *Int. J. Contr.*, 1983, 38 (5), pp. 977-990
[2] Leva, A., C. Maffezzoni and R. Scattolini : 'Self-tuning PI-PID regulators for stable systems with varying delay', *Automatica*, 1994, 30 (7), pp. 1171-1183
[3] Kim, J.H., J.H. Park and S.W. Lee : 'A two-layered fuzzy logic controller for systems with deadzones', IFSA'93 World Congress, July 1993, Seoul, Korea, pp. 826-829
[4] Jang, M.J. and C.L. Chen : 'Fuzzy successive modelling and control for time-delay systems', *International Journal of Systems Science*, 1996, 27 (12), pp. 1483-1486
[5] Li, H.X. and S.K. Tso : 'Higher order fuzzy control structure for higher order or time-delay systems', *IEEE Trans. Fuzzy Systems*, 1999, 7 (5), pp. 540-542
[6] Takagi, T. and M. Sugeno : 'Fuzzy identification of systems and its applications to modeling and control', *IEEE Trans. Systems, Man, and Cybernetics*, 1985, 15 (1), pp. 116-132
[7] Tanaka K. and M. Sugeno : 'Stability analysis and design of fuzzy control systems', *Fuzzy Sets and Systems*, 1992, 45 (2), pp. 135-156
[8] Ting, C.S., T.H.S. Li and F.C. Kung : 'An approach to systematic design of fuzzy control system', *Fuzzy Sets and Systems*, 1996, 77, pp. 151-166
[9] Tanaka, K. and M. Sano : 'A robust stabilization problem of fuzzy control systems and its application to backing up control of a truck-trailer', *IEEE Trans. Fuzzy Systems*, 1994, 2 (2), pp. 119-133
[10] Wang, H.O., K. Tanaka and M.F. Griffin : 'An approach to fuzzy control of nonlinear systems: Stability and design issues', *IEEE Trans. Fuzzy Systems*, 1996, 4 (1), pp. 14-23
[11] Tanaka, K., T. Kosaki and H.O. Wang : 'Backing control problem of a mobile robot with multiple trailers: fuzzy modeling and LMI-based design', *IEEE Trans. Fuzzy Systems*, 1998, 28 (3), pp. 329-337
[12] Kiriakidis, K. : 'Non-linear control system design via fuzzy modelling and LMIs', *Int. J. Control*, 1999, 72 (7/8), pp. 676-685
[13] Boyd, S., L.E. Ghaoui, E. Feron and V. Balakrishnan : 'Linear matrix inequalities in systems and control theory' (SIAM, Philadelphia, 1994)
[14] Nesterov, Y. and A. Nemirovsky : 'Interior-point polynomial methods in convex programming' (SIAM, Philadelphia, 1994)
[15] Kong, S.G. and B. Kosko : 'Adaptive fuzzy systems for backing up a truck-and-trailer', *IEEE Trans. Neural Net.*, 1992, 3, pp. 211-223
[16] Tokunaga, M. and H. Ichihashi : 'Backer-upper control of a trailer truck by neuro-fuzzy optimal control', 8th Fuzzy System Symp., May 1992, Japan, pp. 49-52
[17] Tanaka, K. and T. Kosaki : 'Design of a stable fuzzy controller for an articulated vehicle', *IEEE Trans. Systems, Man, and Cybernetics*, 1997, 27 (3), pp. 552-558
[18] Gahinet, P., A. Nemirovski, A. Laub, and M. Chilali : 'LMI Control Toolbox' (The MathWorks, Inc., Natick, 1995)

5

Adaptive Neuro-Fuzzy Systems

Azar, Ahmad Taher
Electrical Communication & Electronics Systems Engineering department, Modern Science and Arts University (MSA), 6th of October City, Egypt

1. Introduction

One benefit of fuzzy systems (Zadeh, 1965; Ruspini et al., 1998; Cox, 1994) is that the rule base can be created from expert knowledge, used to specify fuzzy sets to partition all variables and a sufficient number of fuzzy rules to describe the input/output relation of the problem at hand. However, a fuzzy system that is constructed by expert knowledge alone will usually not perform as required when it is applied because the expert can be wrong about the location of the fuzzy sets and the number of rules. A manual tuning process must usually be appended to the design stage which results in modifying the membership functions and/or the rule base of the fuzzy system. This tuning process can be very time-consuming and error-prone. Also, in many applications expert knowledge is only partially available or not at all. It is therefore useful to support the definition of the fuzzy rule base by automatic learning approaches that make use of available data samples. This is possible since, once the components of the fuzzy system is put in a parametric form, the fuzzy inference system becomes a parametric model which can be tuned by a learning procedure. Fuzzy logic and artificial neural networks (Haykin, 1998; Mehrotra et al., 1997) are complementary technologies in the design of intelligent systems. The combination of these two technologies into an integrated system appears to be a promising path toward the development of Intelligent Systems capable of capturing qualities characterizing the human brain. Both neural networks and fuzzy logic are powerful design techniques that have their strengths and weaknesses. Table 1 shows a comparison of the properties of these two technologies (Fuller, 2000). The integrated system will have the advantages of both neural networks (e.g. learning abilities, optimization abilities and connectionist structures) and fuzzy systems (humanlike IF-THEN rules thinking and ease of incorporating expert knowledge) (Brown & Harris,1994). In this way, it is possible to bring the low-level learning and computational power of neural networks into fuzzy systems and also high-level humanlike IF-THEN thinking and reasoning of fuzzy systems into neural networks. Thus, on the neural side, more and more transparency is pursued and obtained either by pre-structuring a neural network to improve its performance or by possible interpretation of the weight matrix following the learning stage. On the fuzzy side, the development of methods allowing automatic tuning of the parameters that characterize the fuzzy system can largely draw inspiration from similar methods used in the connectionist community. This combination does not usually mean that a neural network and a fuzzy system are used together in some way.

The neuro-fuzzy method is rather a way to create a fuzzy model from data by some kind of learning method that is motivated by learning procedures used in neural networks. This substantially reduces development time and cost while improving the accuracy of the resulting fuzzy model. Being able to utilize a neural learning algorithm implies that a fuzzy system with linguistic information in its rule base can be updated or adapted using numerical information to gain an even greater advantage over a neural network that cannot make use of linguistic information and behaves as a black-box. Equivalent terms for neuro-fuzzy systems that can be found in the literature are neural fuzzy or sometimes neuro-fuzzy networks (Buckley & Eslami, 1996). Neuro-fuzzy systems are basically adaptive fuzzy systems developed by exploiting the similarities between fuzzy systems and certain forms of neural networks, which fall in the class of generalized local methods. Hence, the behavior of a neuro-fuzzy system can either be represented by a set of humanly understandable rules or by a combination of localized basis functions associated with local models (i.e. a generalized local method), making them an ideal framework to perform nonlinear predictive modeling. Nevertheless, one important consequence of this hybridization between the representational aspect of fuzzy models and the learning mechanism of neural networks is the contrast between readability and performance of the resulting model.

Skills	Type	Fuzzy Systems	Neural Nets
Knowledge acquisition	Inputs	Human experts	Sample sets
	Tools	Interaction	Algorithms
Unceratinity	Information	Quantitative and Qualitative	Quantitative
Reasoning	Cognition	Heuristic approach	Perception
	Mechanism	Low	Parallel Computation
	Speed	low	High
Adaption	Fault-tolerance	Low	Very high
	Learning	Induction	Adjusting weights
Natural language	Implementation	Explicit	Implicit
	Flexibility	High	Low

Table 1. Properties of neural networks and fuzzy Systems (Fuller, 2000).

Summarizing, neural networks can improve their transparency, making them closer to fuzzy systems, while fuzzy systems can self-adapt, making them closer to neural networks (Lin & Lee, 1996). Fuzzy systems can be seen as a special case of local modeling methods, where the input space is partitioned into a number of fuzzy regions represented by multivariate membership functions. For each region, a rule is defined that specifies the output of the system in that region. The class of functions that can be accurately reproduced by the resulting model is determined by the nonlinear mapping performed by the multivariate fuzzy membership functions. This impressive result allows comparison to be drawn between fuzzy systems and the more conventional techniques referred to as generalized local methods. In particular, if bell-shaped (Gaussian) membership functions are used, then a Takagi-Sugeno (TS) fuzzy system is equivalent to a special kind of Radial Basis Function (RBF) network (Jang & Sun, 1993).

Theorems and analysis derived for local modeling methods can directly be applied to fuzzy systems. Also, due to this similarity, fuzzy systems allow relatively easy application of learning techniques used in local methods for identification of fuzzy rules from data. On the other side, fuzzy systems distinguish from other local modeling techniques, for their potentiality of an easy pre-structuring and a convenient integration of a priori knowledge. Many learning algorithms from the area of local modeling, and more specifically techniques developed for some kind of neural networks, have been extended to automatically extract or tune fuzzy rules based on available data. All these techniques exploit the fact that, at the computational level, a fuzzy system can be seen as a layered architecture, similar to an artificial neural network. By doing so, the fuzzy system becomes a neuro-fuzzy system, i.e. special neural network architecture. In 1991, Lin and Lee have proposed the very first implementation of Mamdani fuzzy models using layered feed-forward architecture (Lin & Lee, 1991). Nevertheless, the most famous example of neuro-fuzzy network is the Adaptive Network-based Fuzzy Inference System (ANFIS) developed by Jang in 1993 (Jang, 1993), that implements a TS fuzzy system in a network architecture, and applies a mixture of plain back-propagation and least mean squares procedure to train the system.

2. Types of neuro-fuzzy systems

There are several ways to combine neural networks and fuzzy logic. Efforts at merging these two technologies may be characterized by considering three main categories: neural fuzzy systems, fuzzy neural networks and fuzzy-neural hybrid systems.

2.1 Neural fuzzy systems

Neural fuzzy systems are characterized by the use of neural networks to provide fuzzy systems with a kind of automatic tuning method, but without altering their functionality. One example of this approach would be the use of neural networks for the membership function elicitation and mapping between fuzzy sets that are utilized as fuzzy rules as shown in Fig. 1. In the training process, a neural network adjusts its weights in order to minimize the mean square error between the output of the network and the desired output. In this particular example, the weights of the neural network represent the parameters of the fuzzification function, fuzzy word membership function, fuzzy rule confidences and

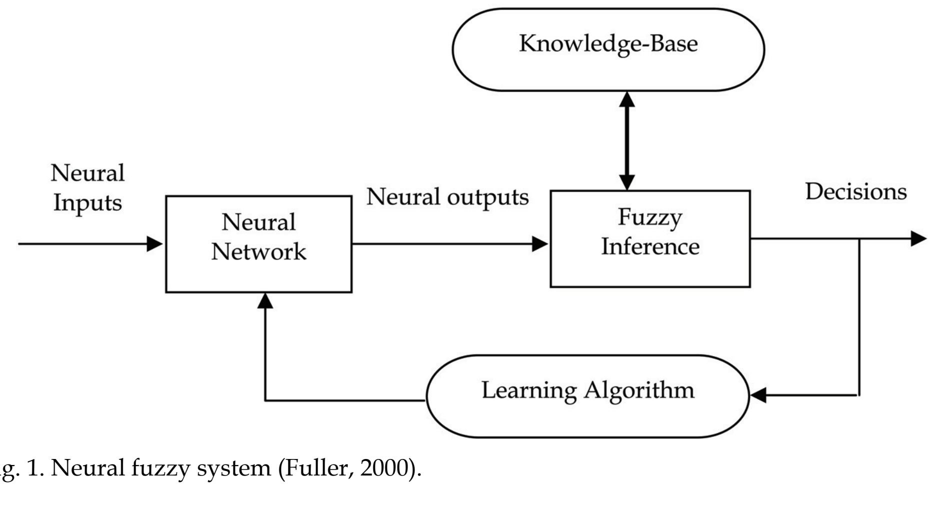


Fig. 1. Neural fuzzy system (Fuller, 2000).

defuzzification function respectively. In this sense, the training of this neural network results in automatically adjusting the parameters of a fuzzy system and finding their optimal values.This kind of combination is mostly used in control applications. Examples of this approach can be found in (Wang & Mendel, 1992; Nomura et al., 1992; Nauck, 1994; Shi & Mizumoto, 2000a; Shi & Mizumoto, 2000b; Yager & Filev, 1994; Cho & Wang, 1996; Ichihashi & Tuksen, 1993).

2.2 Fuzzy neural systems

The main goal of this approach is to 'fuzzify' some of the elements of neural networks, using fuzzy logic (Fig. 2). In this case, a crisp neuron can become fuzzy. Since fuzzy neural networks are inherently neural networks, they are mostly used in pattern recognition applications. In 1996, for instance, Lin and Lee presented a neural network composed of fuzzy neurons (Lin and Lee, 1996). In these fuzzy neurons, the inputs are non-fuzzy, but the weighting operations are replaced by membership functions. The result of each weighting operation is the membership value of the corresponding input in the fuzzy set. Also, the aggregation operation may use any aggregation operators such as min and max and any other t-norms and t-conorms.

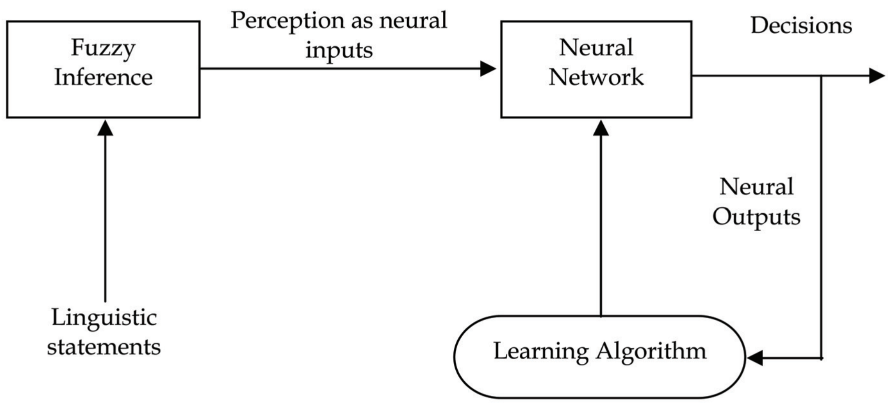


Fig. 2. Fuzzy neural system (Fuller, 2000).

2.3 Hybrid neuro-fuzzy systems

In this approach, both fuzzy and neural networks techniques are used independently, becoming, in this sense, a hybrid system. Each one does its own job in serving different functions in the system, incorporating and complementing each other in order to achieve a common goal. This kind of merging is application-oriented and suitable for both control and pattern recognition applications. The idea of a hybrid model is the interpretation of the fuzzy rule-base in terms of a neural network. In this way the fuzzy sets can be interpreted as weights, and the rules, input variables, and output variables can be represented as neurons. The learning algorithm results, like in neural networks, in a change of the architecture, i.e. in an adaption of the weights, and/or in creating or deleting connections. These changes can be interpreted both in terms of a neural net and in terms of a fuzzy controller. This last aspect is very important as the black box behaviour of neural nets is avoided this way. This means a successful learning procedure results in an explicit increase of knowledge that can be represented in form of a fuzzy controller's rule base. Hybrid neuro-fuzzy controllers are realized by approaches like ARIC (Berenji, 1992), GARIC (Bersini et al., 1993), ANFIS (Jang, 1993) or the NNDFR model (Takagi & Hayashi 1991). These approaches consist all more or less of special neural networks, and they are capable to learn fuzzy sets.

3. Adaptive-Neuro-Fuzzy Inference System: ANFIS

Adaptive Neuro-Fuzzy Inference System (ANFIS) is one of the most successful schemes which combine the benefits of these two powerful paradigms into a single capsule (Jang, 1993). An ANFIS works by applying neural learning rules to identify and tune the parameters and structure of a Fuzzy Inference System (FIS). There are several features of the ANFIS which enable it to achieve great success in a wide range of scientific applications. The attractive features of an ANFIS include: easy to implement, fast and accurate learning, strong generalization abilities, excellent explanation facilities through fuzzy rules, and easy to incorporate both linguistic and numeric knowledge for problem solving (Jang & Sun, 1995; Jang et al., 1997). According to the neuro-fuzzy approach, a neural network is proposed to implement the fuzzy system, so that structure and parameter identification of the fuzzy rule base are accomplished by defining, adapting and optimizing the topology and the parameters of the corresponding neuro-fuzzy network, based only on the available data. The network can be regarded both as an adaptive fuzzy inference system with the capability of learning fuzzy rules from data, and as a connectionist architecture provided with linguistic meaning. A typical architecture of an ANFIS, in which a circle indicates a fixed node, whereas a square indicates an adaptive node, is shown in Figure 3. In this

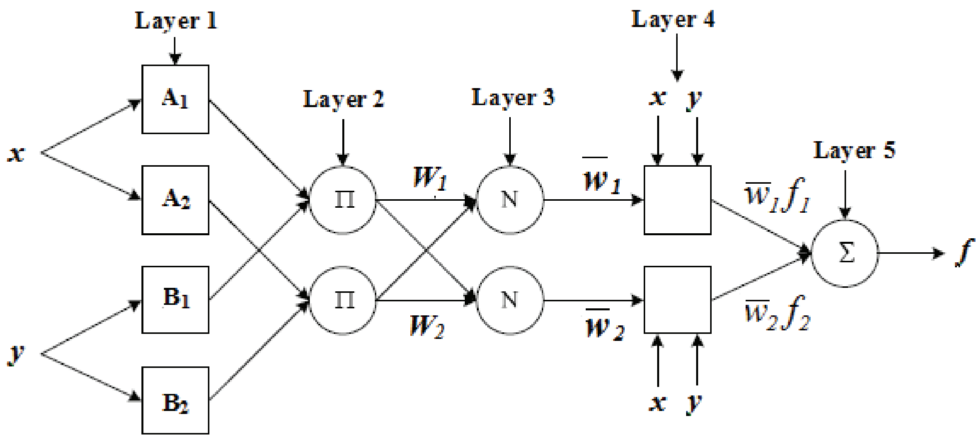


Fig. 3. First order Sugeno ANFIS architecture (Type-3 ANFIS) (Jang, 1993).

connectionist structure, there are input and output nodes, and in the hidden layers, there are nodes functioning as membership functions (MFs) and rules. This eliminates the disadvantage of a normal feedforward multilayer network, which is difficult for an observer to understand or to modify. For simplicity, we assume that the examined FIS has two inputs and one output. For a first-order Sugeno fuzzy model, a typical rule set with two fuzzy "if-then" rules can be expressed as follows:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1 x + q_1 y + r_1$
Rule 2: If x is A_2 and y is B_2, then $f_2 = p_1 x + q_2 y + r_2$

Where x and y are the two crisp inputs, and A_i and B_i are the linguistic labels associated with the node function.
As indicated in Fig. 3, the system has a total of five layers. The functioning of each layer is described as follows (Jang, 1993).

Input node (Layer 1): Nodes in this layer contains membership functions. Parameters in this layer are referred to as premise parameters. Every node i in this layer is a square and adaptive node with a node function:

$$O_i^1 = \mu_{A_i}(x) \qquad \text{For i = 1, 2} \tag{1}$$

Where x is the input to node i, and A_i is the linguistic label (small , large, etc.) associated with this node function. In other words, O_i^1 is the membership function of A_i and it specifies the degree to which the given x satisfies the quantifier A_i.

Rule nodes (Layer 2): Every node in this layer is a circle node labeled II, whose output represents a firing strength of a rule. This layer chooses the minimum value of two input weights. In this layer, the AND/OR operator is applied to get one output that represents the results of the antecedent for a fuzzy rule, that is, firing strength. It means the degrees by which the antecedent part of the rule is satisfied and it indicates the shape of the output function for that rule. The node generates the output (firing strength) by cross multiplying all the incoming signals:

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad i = 1, 2 \tag{2}$$

Average nodes (Layer 3): Every node in this layer is a circle node labeled N. The i^{th} node calculates the ratio between the i^{th} rule's firing strength to the sum of all rules' firing strengths. Every node of these layers calculates the weight, which is normalized.
For convenience, outputs of this layer are called normalized firing strengths.

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2 \tag{3}$$

Consequent nodes (Layer 4): This layer includes linear functions, which are functions of the input signals. This means that the contribution of ith rule's towards the total output or the model output and/or the function defined is calculated. Every node i in this layer is a square node with a node function:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \tag{4}$$

Where $\overline{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set of this node. These parameters are referred to as consequent parameters.

Output node (Layer 5): The single node in this layer is a fixed node labeled $\sum$, which computes the overall output by summing all incoming signals:

$$O_i^5 = \text{overalloutput} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{5}$$

4. Modeling with neuro-fuzzy systems

Whatever may be the adopted vision of fuzzy model, two different phases must be carried out in fuzzy modeling, designated as structural and parametric identification. Structural identification consists of determining the structure of the rules, i.e. the number of rules and the number of fuzzy sets used to partition each variable in the input and output space so as to derive linguistic labels. Once a satisfactory structure is available, the parametric identification must follow for the fine adjustment of the position of all membership functions together with their shape as the main concern. As seen before, to overcome the limitations of using expert knowledge in defining the fuzzy rules, data driven methods to create fuzzy systems are needed. With such methods both structure and parameters are derived from scratch relying only on the training data. There are several ways that structure learning and parameter learning can be combined in a neuro-fuzzy system. They can be performed sequentially: structure learning is used first to find an appropriate structure of the fuzzy rule base, and then parameter learning is used to identify the parameters of each rule. In some neuro-fuzzy systems the structure is fixed and only parameter learning is performed. Algorithms inspired by neural network learning often do parameter learning. Structure learning on the other hand is usually not from neural networks. Indeed, many different approaches exist to automatically determine the structure of neural networks, but none of them is appropriate to perform structure identification in neuro-fuzzy models. In the following, different methods are presented that used for structure and parameter identification in neuro-fuzzy systems. There may be a lot of structure/parameter combinations which make the fuzzy model to behave satisfactorily; hence the search for the best model is not an easy task.

As a rule, simple fuzzy models should be preferred to complex ones; hence in the search for the best model two main objectives must be taken into account: good accuracy and minimal complexity.

4.1 Parametric identification

Two types of parameters characterize a fuzzy model: those determining the shape and distribution of the input fuzzy sets and those describing the output fuzzy sets (or linear models). Many neuro-fuzzy systems use direct nonlinear optimization to identify all the parameters of a fuzzy system. Different optimization techniques can be used to this aim. The most widely used is an extension of the well-known back-propagation algorithm implemented by gradient descent. A very large number of neuro-fuzzy systems are based on backpropagation. One limitation of using gradient descent techniques is that the membership functions and all functions that take part in the inference of the fuzzy rule base must be differentiable. As a consequence, gradient descent learning can be more easily applied to identify the parameters of a TS model, because only the product operator is used for intersection and the output is computed as a weighted sum. Recent neuro-fuzzy approaches choose to implement back-propagation by simple heuristics instead of gradient descent to identify the parameters of a Mamdani-type fuzzy model (Nauck & Kruse, 1999).

The general idea of such heuristics is to slightly modify the membership functions of a fuzzy rule according to how much the rule contributes to the overall output of the fuzzy system. From the proposed type-3 ANFIS architecture (see Fig. 3), it is observed that given the values of premise parameters, the overall output can be expressed as a linear combinations of the consequent parameters. More precisely, the output f in Fig. 3 can be rewritten as:

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 = \overline{w}_1 f_1 + \overline{w}_2 f_2 \\ &= (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2 \end{aligned} \tag{6}$$

Which is linear in the consequent parameters (p_l, q_1, r_l, p_2, q_2 and r_2). Therefore the hybrid learning algorithm can be applied directly. More specifically, in the forward pass of the hybrid learning algorithm, functional signals go forward till layer 4 and the consequent parameters are identified by the least squares estimate (LSE). In the backward pass, the error rates propagate backward and the premise parameters are updated by the gradient descent. Table 2 summarizes the activities in each pass. As mentioned earlier, the consequent parameters thus identified are optimal (in the consequent parameter space) under the condition that the premise parameters are fixed.

	Forward Pass	**Backward pass**
Premise Parameters	Fixed	Gradient Descent
Consequent parameters	Least-squares estimator	Fixed
Signals	Node Outputs	Error Signals

Table 2 The two passes in the hybrid learning algorithm (Jang & Sun, 1995).

However, it should be noted that the computation complexity of the least squares estimate is higher than that of the gradient descent. In fact, there are four methods to update the parameters, as listed below according to their computation complexities (Jang, 1993):

- Gradient Descent Only: All parameters are updated by the gradient descent.
- Gradient Descent and One Pass of LSE: The LSE is applied only once at the beginning to get the initial value of the consequent parameters and then the gradient descent takes over to update all parameters.
- Gradient descent and LSE: This is the proposed hybrid learning rule.
- Sequential (Approximate) LSE Only: The ANFIS is linearized with respect to the premise parameters and the extended Kalman filter algorithm is employed to update all parameters.

The choice of above methods should be based on the trade-off between computation complexity and resulting performance. Other approaches to parameter learning of fuzzy models that do not require gradient computations, and hence differentiability, are reinforcement learning which requires only a single scalar evaluation of the output, and Genetic Algorithms (GAs) that perform a random search in the parameter space, using a population of individuals, each coding the parameters of a potential fuzzy rule base (Seng et al., 1999). One problem with GAs is that with conventional binary coding, the length of individuals increases significantly with the number of inputs, the number of fuzzy sets and the number of rules. Evolution Strategies (ES) are more suitable techniques to tune the fuzzy rule parameters due to their direct coding scheme (Jin et al, 1999). GA's and ES allow also a simultaneous identification of the parameters and the structure (rule number) of a fuzzy model, but in such a case these evolutionary techniques are computationally demanding since very complex individuals need to be manipulated. The identification of the whole set of parameters by nonlinear optimization techniques may be computationally intensive and requiring long convergence rates. To speed up the process of parameter identification, many neuro-fuzzy systems adopt a multi-stage learning procedure to find and optimize the parameters. Typically, two stages are considered. In the first stage the input space is partitioned into regions by unsupervised learning, and from each region the premise (and eventually the consequent) parameters of a fuzzy rule are derived. In the second stage the consequent parameters are estimated via a supervised learning technique. In most cases, the second stage performs also a fine adjustment of the premise parameters obtained in the first stage using a nonlinear optimization technique.

4.2 Structural identification

Before fuzzy rule parameters can be optimized, the structure of the fuzzy rule base must be defined. This involves determining the number of rules and the granularity of the data space, i.e. the number of fuzzy sets used to partition each variable. In fuzzy rule-based systems, as in any other modeling technique, there is a tradeoff between accuracy and complexity. The more rules, the finer the approximation of the nonlinear mapping can be obtained by the fuzzy system, but also more parameters have to be estimated, thus the cost and complexity increase. A possible approach to structure identification is to perform a stepwise search through the fuzzy model space. Once again, these search strategies fall into one of two general categories: forward selection and backward elimination.

- Forward selection. Starting from a very simple rule base, new fuzzy rules are dynamically added or the density of fuzzy sets is incrementally increased (Royas et al., 2000).
- Backward elimination. An initial fuzzy rule base, constructed from a priori knowledge or by learning from data, is reduced, until a minimum of the error function is found (Yen & Wang, 1999). The structure of the fuzzy rules can also be optimized by GA's so that a compact fuzzy rule base can be obtained (Seng et al., 1999).

The learning algorithm is an example of structure adaptation in neuro-fuzzy systems. Rules are dynamically recruited or deleted according to their significance to system performance, so that a parsimonious structure with high performance is achieved. When initial fuzzy rules are generated by clustering, the number of cluster (i.e. of rules) must be specified before clustering. If no prior knowledge is available that suggests the number of clusters, automated procedures can be applied. For example the number of clusters can be found by evaluating a given validity measure, i.e. a criterion that assesses the quality of the clusters, and selecting the number of clusters that minimizes (maximizes) the validity measure. Another approach is cluster merging, that starts with a high number of clusters and reduces them successively by merging compatible clusters until some threshold is reached and no more clusters can be merged.

5. Interpretability versus accuracy of neuro-fuzzy models

As seen in the previous sections, neuro-fuzzy systems are essentially fuzzy systems endowed with learning capabilities inspired (not only) by neural networks. Fuzzy systems join the advantages of modeling methods oriented to provide suitable models for both prediction and understanding. It must be considered whether these advantages of fuzzy systems for predictive modeling are preserved when they are transformed into neuro-fuzzy systems. The twofold face of fuzzy systems leads to a trade-off between readability and accuracy (table 3). Fuzzy systems can be forced to arbitrary precision, but it then loose interpretability. To be very precise, a fuzzy system needs a fine granularity and many fuzzy rules. It is obvious that the larger the rule base of a fuzzy system becomes, the less interpretable it gets (Nauck & Kruse, 1998a; Nauck & Kruse, 1998b).

	Interpretability	**Accuracy**
No. of parameters	Few Parameters	More Parameters
No. of fuzzy rules	Few Rules	More Rules
Type of Fuzzy logic Model	Mamdani Models	TSK models

Table 3. Interpretability vs. accuracy in fuzzy systems.

To keep the model simple, the prediction is usually less accurate. In solving this trade-off the interpretability (meaning also simplicity) of fuzzy systems must be considered the major advantage and hence it should be pursuit more than accuracy.

In fact fuzzy systems are not better function approximators or classifiers than other approaches. If we are interested in a very precise prediction, then we are usually not so much interested in the interpretability of the solution. In this case we use just one feature of fuzzy systems: the convenient combination of local models to an overall solution. For this, Sugeno-type models are more suited than Mamdani-type models because they offer more flexibility in the consequents of the rules. However, if optimal performance is the main objective, we should consider whether a fuzzy system is the most suitable approach and an exhaustive and deep comparison with related methods (local methods and generalized local methods) has to be done, in terms of pure performance, computational cost and practicability. Briefly put, fuzzy systems should be used for predictive modeling if an interpretable model is needed that can also be used to some extent for prediction. Interpretability of a fuzzy model should not mean that there is an exact match between the linguistic description of the model and the model parameters. This is not possible anyway, due to the subjective nature of fuzzy sets and linguistic terms. Usually it is not important that, for example, the term approximately zero be represented by a symmetrical triangular fuzzy set with support [-1, 1]. Interpretability means that the users of the model can accept the representation of the linguistic terms, more or less. The representation must roughly correspond to their intuitive understanding of the linguistic terms. Furthermore, interpretability should not mean that anybody could understand a fuzzy model. It means that users who are at least to some degree experts in the domain where the predictive modeling takes place can understand the model. Since interpretability itself is a fuzzy and subjective concept, it is hard to find an explicit and exhaustive list of conditions which, when violated, make the fuzzy model to lose its readability.

Traditional neuro-fuzzy modeling techniques, and in general data-driven methods for learning fuzzy rules from data, are aimed to optimize the prediction accuracy of the fuzzy model. However, while the accuracy improves, the transparency of the fuzzy models after learning may be lost. The overlap of the membership functions typically increases and peculiar situations may occur, when some membership functions are contained in the others or membership functions swap their positions. This hampers the interpretability of the final model. For the sake of interpretability, the learning procedure should take the semantics of the desired fuzzy system into account, and adhere to certain constraints, so that it cannot apply all the possible modifications to the parameters of a fuzzy system. For example the learning algorithms should be constrained such that adjacent membership functions do not exchange positions, do not move from positive to negative parts of the domains or vice versa, have a certain degree of overlapping, etc. The other important requirement to obtain interpretability is to keep the rule base small. A fuzzy model with interpretable membership functions but a very large number of rules is far from being understandable. By reducing the complexity, i.e. the number of parameters, of a fuzzy model, not only the rule base is kept manageable (hence the inference process is computationally cheaper) but also it can provide a more readable description of the process underlying the data. Also the use of a simple rule base contributes to decrease the overfitting, thus improving generalization. So far, few data-driven fuzzy rule learning methods aiming at improving the interpretability of the fuzzy models in terms of both small rule base and readable fuzzy sets have been proposed.

6. Case study: Adaptive-Neuro-Fuzzy Inference System as a novel approach for post-dialysis urea rebound prediction

6.1 Problem statement

Kinetic models of urea concentration are now widely used to manage hemodialysis (HD) patients. The calculation Kt/V (where K is the dialyzer clearance, t is the time of treatment, and V is the urea distribution volume), is now widely used to quantify HD treatment (Depner, 1994; Depner 1999). The Kt/V calculation is commonly determined from measurements of the pre-and post-HD blood urea nitrogen (BUN) concentrations (Gotch & Sargent,1985). However, because the rapid removal of BUN during HD causes a concentration disequilibrium between intracellular and extracellular fluid spaces, BUN increases immediately following HD. This phenomenon is well known as the urea rebound, and is due to the multiple-pool nature of the human body, and mass transfer resistance of the biological membranes and variations in regional blood flows (Schneditz & Daugirdas, 2001), Yashiro et al., 2004). Since Kt/V calculation is based in part on the post-hemodialysis BUN level, urea rebound has a significant impact upon the calculation of the delivered dose of hemodialysis. While single-pool kinetic modeling ($_{sp}$Kt/V) uses a convenient 30-second post-dialysis BUN sample, it does not take urea rebound into account, which leads to a 12 to 40% of the true equlilibrated dialysis dose ($_{eq}$Kt/V). Double-pool modeling ($_{eq}$Kt/V) uses an equilibrated BUN (C_{eq}) and is the best reflection of the true urea mass removed by hemodialysis. Because a delay of 30 to 60 minutes after dialysis before sampling the urea is inconvenient for both the clinician and patient, several methods have been devised to predict the PDUR in order to estimate the equilibrated Kt/V. The first is based on the standard single-pool Daugirdas Kt/V model that takes into account the dialysis time, which evolved into a double-pool Kt/V ($_{eq}$Kt/V) formula (Daugirdas & Schneditz, 1995). The second, according to Smye (Smye et al., 1994), Daugirdas (Daugirdas et al., 1996), Tattersall (Tattersall et al., 1996), and Maduell (Maduell et al., 1997), is based on an intradialytic urea sample at 33% of the session time. Other methods use a urea sample taken 30 minutes before the end of the hemodialysis session, which corresponds to the 30-minute PDUR (Bhaskaran et al., 1997, Canaud at al., 1997). Finally, Artificial Neural Network (ANN) method was used as a predictor of equilibrated post-dialysis blood urea concentration (C_{eq}) (Guh et al., 1998;

Azar et al., 2008a; Azar et al., 2009a). All of these methods still overestimate the urea rebound and underestimate the equlilibrated dialysis dose ($_{eq}Kt/V$).

6.2 Subjects and methods

The study was carried out at four dialysis centers. BUN was measured in all serum samples at a central laboratory. The overall study period was 5 months from August 1, 2008 to December 31, 2008. No subjects dropped out of the study. The study subjects consisted of 310 hemodialysis patients that gave their informed consent to participate. They are 165 male and 145 female patients, with ages ranging 14-75 years (48.97±12.77, mean and SD), and dialysis therapy duration ranging 6-138 months (50.56±34.67). The etiology of renal failure was chronic glomerulonephritis (65 patients), diabetic nephropathy (60 patients), vascular nephropathy (55 patients), hypertension (51 patients), interstitial chronic nephropathy (45 patients), other etiologies (18 patients) and unknown cause (16 patients). The vascular access was through a native arteriovenous fistula (285 patients), and a permanent jugular catheter (25 patients).

Patients had dialysis three times a week, in 3-4 hour sessions, with a pump arterial blood flow of 200-350 ml/min, and flow of the dialysis bath of 500-800 ml/min. The dialysate consisted of the following constituents: sodium 141 mmol/l, potassium 2.0 mmol/l, calcium 1.3 mmol/l, magnesium 0.2 mmol/l, chloride 108.0 mmol/l, acetate 3.0 mmol/l and bicarbonate 35.0 mmol/l. Special attention was paid to the real dialysis time, so that time-counters were fitted to all machines for all sessions, to record effective dialysis duration (excluding any unwanted interruptions, e.g. due to dialysis hypotensive episodes). All patients were dialyzed with 1.0 m^2 Polyethersulfone low flux dialyzer, 1.2 m^2 cellulose-synthetic low flux dialyzer (hemophane), 1.3 m^2 Polyethersulfone low flux dialyzer, 1.3 m^2 low flux polysulfone dialyzer, 1.6 m^2 low flux polysulfone dialyzer and 1.3 m^2 high flux polysulfone dialyzer. The dialysis technique was conventional hemodialysis, no patient being treated with hemodiafiltration. A Fresenius model 4008B and 4008S dialysis machine equipped with a volumetric ultrafiltration control system was used in each dialysis. Fluid removal was calculated as the difference between the patients' weight before dialysis and their target dry weight. Pre-dialysis body weight, blood pressure, pulse rate and axillary temperature were measured before ingestion of food and drink. Pre-dialysis BUN (C_{pre}) was sampled from the arterial port before the blood pump was started. Post-dialysis BUN (C_{post}) was obtained from the arterial port at the end of HD with the blood flow rate unchanged. Equilibrated post-dialysis BUN (C_{eq}) was obtained from the peripheral vein 30 and 60 minutes after HD. It was then corrected for urea generation. This corrected C_{eq} was used as a "gold standard" or the reference method.

6.3 ANFIS Architecture for equilibrated blood urea concentration prediction

To overcome the problem of overestimating urea rebound, Adaptive Neuro-Fuzzy Inference System (ANFIS) is developed in the form of a zero-order Takagi-Sugeno-Kang fuzzy inference system to predict equilibrated urea (C_{eq}) taken at 30 (C_{eq30}) and 60 (C_{eq60}) min after the end of the hemodialysis (HD) session in order to predict post dialysis urea rebound (PDUR) and equilibrated dialysis dose ($_{eq}Kt/V$) (Azar et al., 2008b; Azar, 2009b). The developed neuro-fuzzy hybrid approach is more accurate and doesn't require the model structure to be known a priori, in contrast to most of the modeling techniques. Also, this system doesn't require 30- or 60-minute post-dialysis urea sample. The proposed ANFIS can construct an input-output mapping based on both expert knowledge (in the form of linguistic rules) and specified input-output data pairs and the least squares estimate (LSE) to identify the parameters (Jang et al., 1997). The ANFIS is a multilayer feed-forward network uses ANN learning algorithms and fuzzy reasoning to characterize an input space to an output space. The architecture of the proposed ANFIS realizes the inference mechanism of zero-order Takagi-Sugeno-Kang (TSK) fuzzy models (Takagi & Sugeno, 1985). The first-order Sugeno models have more freedom degrees and therefore the approximation ability is higher, together with a higher risk to overfit. The use of less freedom degrees is helping to control overfitting for the problem. Then, in this particular problem it is better zero-order. On the other hand, zero-order are more interpretable than first-order (depending on the number of rules required). Therefore, the selection of TSK model type depends on the necessities for the problem and the possibility to overfit the system (if it is important or not to have an interpretable model).

For an n-dimensional input, m-dimensional output fuzzy system, the rule base is composed of a set of fuzzy rules formally defined as:

$$R_k : IF\ (x_1\ is\ A_1^k) AND...AND\ (x_n\ is\ A_n^k)\ THEN\ (y_1\ is\ B_1^k)\ AND...AND\ (y_m\ is\ B_m^k)$$

Where x = (x_1, . . . x_n) are the input variables and $y = (y_1, \dots y_m)$ are the output variables, A_i^k are fuzzy sets defined on the input variables and B_j^k (j =1,…,m) are fuzzy singletons defined on the output variables over the output variables y_j. When y is constant, the resulting model is called "zero-order Sugeno fuzzy model", which can be viewed either as a special case of the Mamdani inference system (Mamdani & Assilian, 1975), in which each rule's consequent is specified by a fuzzy singleton, or a special case of the Tsukamoto fuzzy model (Tsukamoto, 1979), in which each rule's consequent is specified by a MF of a step function center at the constant. Figure 4 illustrate the reasoning mechanism for zero-order Sugeno model. This class of fuzzy models should be used when only performance is the ultimate goal of predictive modeling as in the case of our modeling methodology. This class of fuzzy models can employ all the other types of fuzzy reasoning mechanisms because they represent a special case of each of the above described fuzzy models. More specifically, the consequent part of this simplified fuzzy rule can be seen either as a singleton fuzzy set in the Mamdani model or as a constant output function in TS models. Thus the two fuzzy models are unified under this simplified fuzzy model. Different types of membership functions can be used for the antecedent fuzzy sets. In this work, the membership functions have been tested based on error analysis (calculation of average error). The membership function with minimum error is selected and that will be the suitable membership function to estimate the model. Therefore, triangular-shaped membership functions are used for zero-order TSK based models in this study. Based on a set of K rules, the output for any unknown input vector $x(0)$ is obtained by the following fuzzy reasoning procedure:

- Calculate the degree of fulfillment for the k-*th* rule, for k = 1,…,K, by means of Larsen product operator:

$$\mu_k(X) = \prod_{i=1}^{n} \mu_{ik}(x_i), \qquad k = 1,......,K \tag{7}$$

  Note that when computing the activation strength of a rule, the connective AND can be interpreted through different T-norm operators: typically there is a choice between product and min operators. Here we choose the product operator because it retains more input information than the min operator and generally gives a smoother output surface which is a desirable property in any modeling application.

- Calculate the inferred outputs $\hat{y}_j$ by taking the weighted average of consequent values B_j^k with respect to rule activation strengths μ_k(x):

$$\hat{y}_j = \frac{\sum_{k=1}^{K} \mu_k(X) b_{jk}}{\sum_{k=1}^{K} \mu_k(X)}, \qquad j = 1,....,m \tag{8}$$

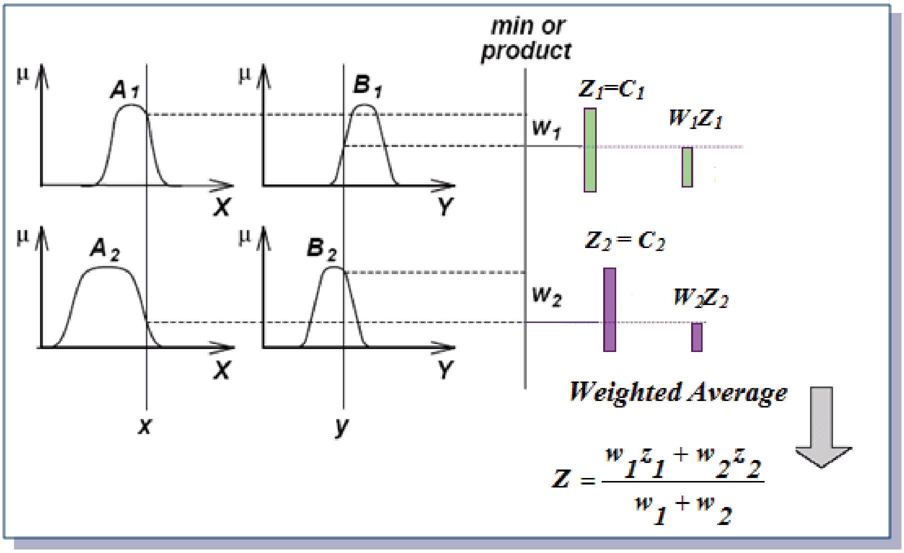


Fig. 4. Zero-order TSK fuzzy inference system with two inputs and two rules (Castillo & Melin, 2001).

6.3.1 Parameter selection for the system

For a real-world modeling problem, it is not uncommon to have tens of potential inputs to the model under construction. An excessive number of inputs not only impair the transparency of the underlying model, but also increase the complexity of computation necessary for building the model. Therefore, it is necessary to do input selection that finds the priority of each candidate inputs and uses them accordingly. Specifically, In order to build a reasonably accurate model for prediction, proper parameters must be selected. The MATLAB function exhsrch performs an exhaustive search within the available inputs to select the set of inputs that most influence the desired output. The first parameter to the function specifies the number of input combinations to be tried during the search. Essentially, exhsrch builds an ANFIS model for each combination and trains it for one epoch and reports the performance achieved. The following are some practical considerations in parameter selection:

- Remove some irrelevant inputs such as the type of dialysate, dialysate temperature, blood pressure of patients, probability of complications, blood volume of patients, intercompartmental urea mass transfer area coefficient, fraction of ultrafiltrate from ICF and access blood flow. This was performed based on the recommendations of an expert in the hemodialysis field. This expert is the medical consultant who supervises the dialysis sessions throughout the research.
- Remove inputs that can be derived from other inputs.
- Make the underlying model more concise and transparent.
- The reduction of the number of parameters results in the reduction of the time required for model construction.
- The selected parameters must affect the target problem, i.e., strong relationships must exist among the parameters and target (or output) variables.
- The selected parameters must be well-populated, and corresponding data must be as clean as possible.

The proposed input selection method is based on the assumption that the ANFIS model with the smallest RMSE (root mean squared error) after one epoch of training has a greater potential of achieving a lower RMSE when given more epochs of training. This assumption is not absolutely true, but it is heuristically reasonable. For instance, if we have a modeling problem with ten candidate inputs and we want to find the most three influential inputs as the inputs to ANFIS, we can construct C_3^{10} =120 ANFIS models, each with different combination of inputs and train them with a single pass of the least-squares method. The ANFIS model with the smallest training error is then selected for further training using the hybrid learning rule to tune the membership functions as well. Note that one-epoch training of 120 ANFIS models in fact involves less computation than 120-epoch training of a single ANFIS model, therefore the input selection procedure is not really as computation intensive as it looks. Therefore, five inputs are selected as the data set for C_{eq} predictor. They are, urea pre-dialysis (Cpre, mg/dl) at the beginning of the procedure, urea post-dialysis (C_{post}, mg/dl), Blood flow rate (BFR, dl/min), desired dialysis Time (T_d, min) and Ultrafiltration rate, the removal of excess water from the patient (UFR, dl/min). All blood samples were obtained from the arterial line at different times for urea determinations. The ANFIS output is the equilibrated post-dialysis BUN (C_{eq}) which was obtained 30 and 60 minutes after HD. Two triangular membership functions (MFs) are assigned to each linguistic variable. The ANFIS structure containing 5^2 = 32 fuzzy rules and 92 nodes. Each fuzzy rule is constructed through several parameters of membership function in layer 2 with a total of 62 fitting parameters, of which 30 are premise (nonlinear) parameters and 32 are consequent (linear) parameters. To achieve good generalization capability, it is important that the number of training data points be several times larger than the number parameters being estimated. In this case, the ratio between data and parameters is five (310/62). Once the FIS structure was identified, the parameters that had to be estimated (Triangular input MF parameters and output constants) were fitted by the hybrid-learning algorithm.

6.4 Training methodology of the developed ANFIS system

The core of the ANFIS calculations was implemented in a MATLAB environment. Functions from the Mathwork's MATLAB Fuzzy Logic Toolbox (FLT) were included in a MATLAB code programmed by the author[1] to solve the input-output problem with different numbers of input MFs, using all data available. An estimate of the mean square error between observed and modeled values were computed for each trial, and the best structure was determined considering a trade-off between the mean square error and the number of parameters involved in computation.

Input MFs were linked by all possible combinations of if-and-then rules defining an output constant for each rule. The flow chart of proposed training methodology of ANFIS system is shown in Fig. 5. The modeling process starts by obtaining a data set (input-output data pairs) and dividing it into training and checking data sets. Training data constitutes a set of input and output vectors. The data is normalized in order to make it suitable for the training process. This normalized data was utilized as the inputs and outputs to train the ANFIS. To avoid overfitting problems during the estimation, the data set were randomly split into two sets: a training set (70% of the data; 220 samples), and a checking set (30% of the data; 90 samples). When both checking data and training data were presented to ANFIS, the FIS was selected to have parameters associated with the minimum checking data model error. In other words, two vectors are formed in order to train the ANFIS, input vector and the output vector (Fig. 5). The training data set is used to find the initial premise parameters for the membership functions by equally spacing each of the membership functions. A threshold value for the error between the actual and desired output is determined. The consequent parameters are found using the least-squares method. Then an error for each data pair is found. If this error is larger than the threshold value, update the premise parameters using the gradient decent method. The process is terminated when the error becomes less than the threshold value. Then the checking data set is used to compare the model with actual system. A lower threshold value is used if the model does not represent the system. Training of the ANFIS can be stopped by two methods. In the first method, ANFIS will be stopped to learn only when the testing error is less than the tolerance limit. This tolerance limit would be defined at the beginning of the training. It is obvious that the performance of a ANFIS that is trained with lower tolerance is greater than ANFIS that is trained with higher tolerance limit. In this method the learning time will change with the architecture of the ANFIS. The second method to stop the learning is to put constraint on the number of learning iterations.

6.5 Testing and validation process of the developed ANFIS

Once the model structure and parameters have been identified, it is necessary to validate the quality of the resulting model. In principle, the model validation should not only validate the accuracy of the model, but also verify whether the model can be easily interpreted to give a better understanding of the modeled process. It is therefore important to combine data-driven validation, aiming at checking the accuracy and robustness of the model, with more subjective validation, concerning the interpretability of the model. There will usually be a challenge between flexibility and interpretability, the outcome of which will depend on their relative importance for a given application. While, it is evident that numerous cross-validation methods exist, the choice of the suitable cross-validation method to be employed in the ANFIS is based on a trade- off between maximizing method accuracy and stability

---

[1] The ANFIS source code developed by the author for training the system is copyright protected and not authorized for sharing.

and minimizing the operation time. In this research, the hold-out cross-validation method is adopted for ANFIS because of its accuracy and possible implementation. The choice of the hold-out method is attributed to its relative stability and low computational time requirements. A major challenge in applying the temporal cross-validation approach is the need to select the length of the checking data set utilized. Different lengths of the cross-validation data set ranging from one tenth to one third of the window size were examined. Apparently, choosing one third of the original data lead to short data set for the training process that may cause difficulty to reach the error goal.

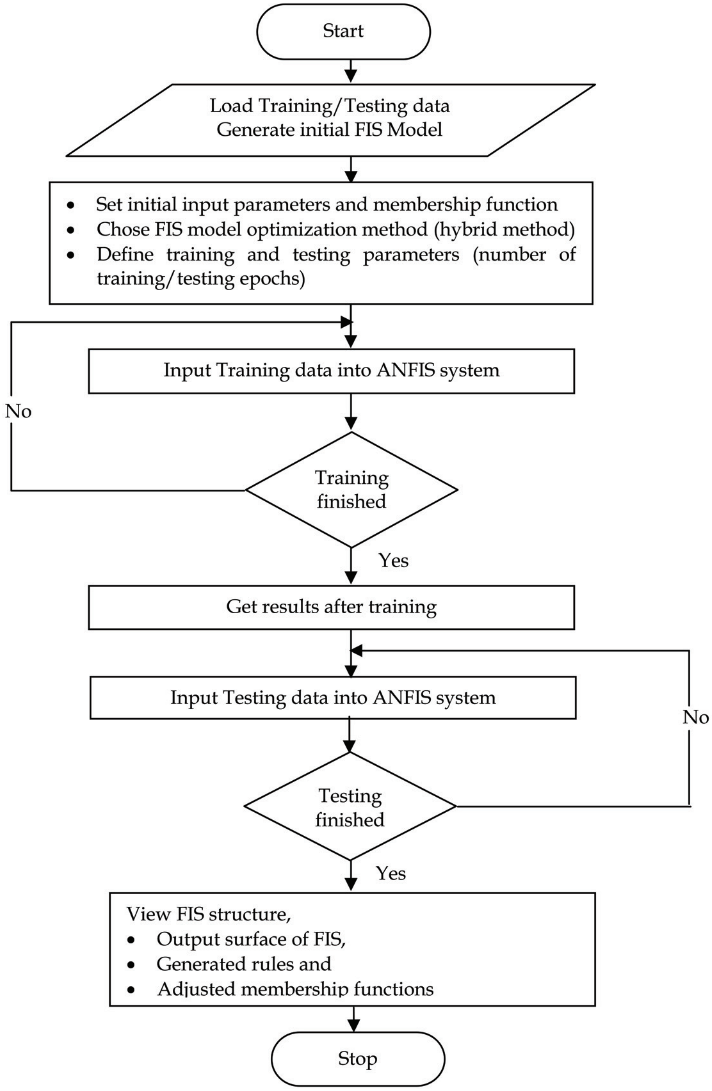


Fig. 5. Flow chart of training methodology of ANFIS system.

While choosing 20% of window size lead to weakness in detecting the features of the expected data set in prediction stage since it leads to relatively short data set for the cross-validation procedure. Therefore, it was decided to select the length of data set for cross-validation utilized in our study to be 30% of the original data-set. Two pair sets were made with different combinations of 70% and 30% of the samples to improve the generalization properties of the adopted ANFIS as follows:

- Pair set 1: training set first 70%; test set last 30%
- Pair set 2: training set last 70%; test set first 30%

For each pair set, two ANFIS models of the same size, but differing in initialization weights, were trained to study the stability and robustness of the each model. The best weights (giving minimum mean-squared error) of two different training sessions over each input/output training set were chosen as the final ANFIS models. The performances of the ANFIS models both training and testing data are evaluated and the best training/testing data set is selected according to mean absolute error (MAE), mean absolute percentage error (MAPE), Root Mean Square Error (RMSE) and normalized root mean squared error (NRMSE). Prediction accuracy is calculated by comparing the difference of predicted and measured values. If the difference is within tolerance, as in $|C_{eq}$ predicted-C_{eq} measured$| \leq \varepsilon$, accurate prediction is achieved. The tolerance ε is defined based based on the recommendations of an expert in the hemodialysis field. In equilibrated blood urea concentration (C_{eq}) prediction, errors of ±1.5% are allowed. So the prediction accuracy is defined as follows:

$$Accuracy = \frac{\left| C_{eq}^{predicted} - C_{eq}^{measured} \right| \leq \varepsilon}{\left\| predicted\ set \right\|} \tag{9}$$

For the five input parameters, each one was assigned two fuzzy sets, i.e. low and high. The membership function μ(k) for each input parameter is divided into two regions, low and high. The number and type of parameters for training ANFIS are shown in table 4.

ANFIS parameter type	**Value**
TSK Type	Zero-order
Numbers of Rules	32
Number of Training Epochs	50
Number of nodes	92
Total fitting parameters	62
• premise (nonlinear) parameters	30
• consequent (linear) parameters	32
Number of Membership functions	Traingular-2
Defuzzification Method	Weighted average
Initial step size, k_{ini}	0.07
Step increasing rate, η	1.6
Step decreasing rate, γ	0.1

Table 4. Training parameters of C_{eq30} ANFIS prediction model.

The collection of well-distributed, sufficient, and accurately measured input data is the basic requirement in order to obtain an accurate model. The data set is divided into separate data sets- the training data set and the test data set. The training data set is used to train the ANFIS, whereas the test data set is used to verify the accuracy and effectiveness of the trained ANFIS. The ANFIS was tuned using a hybrid system that contained a combination of the back propagation and least-squares-type methods. An error tolerance of 0 was used and the ANFIS was trained with 50 epochs. After training and testing, the RMSE became steady, the training and testing were regarded as converged as shown in Fig. 6. RMSE from each of the validating epochs was calculated and averaged to give the mean RMSE. The network error convergence curve achieved mean RMSE values of 0.2978 and 0.3125 for training and testing, respectively. It is noted from error curves that the ANFIS model performed well and it is obvious that the error between the actual and the predicted output of the model is very insignificant.

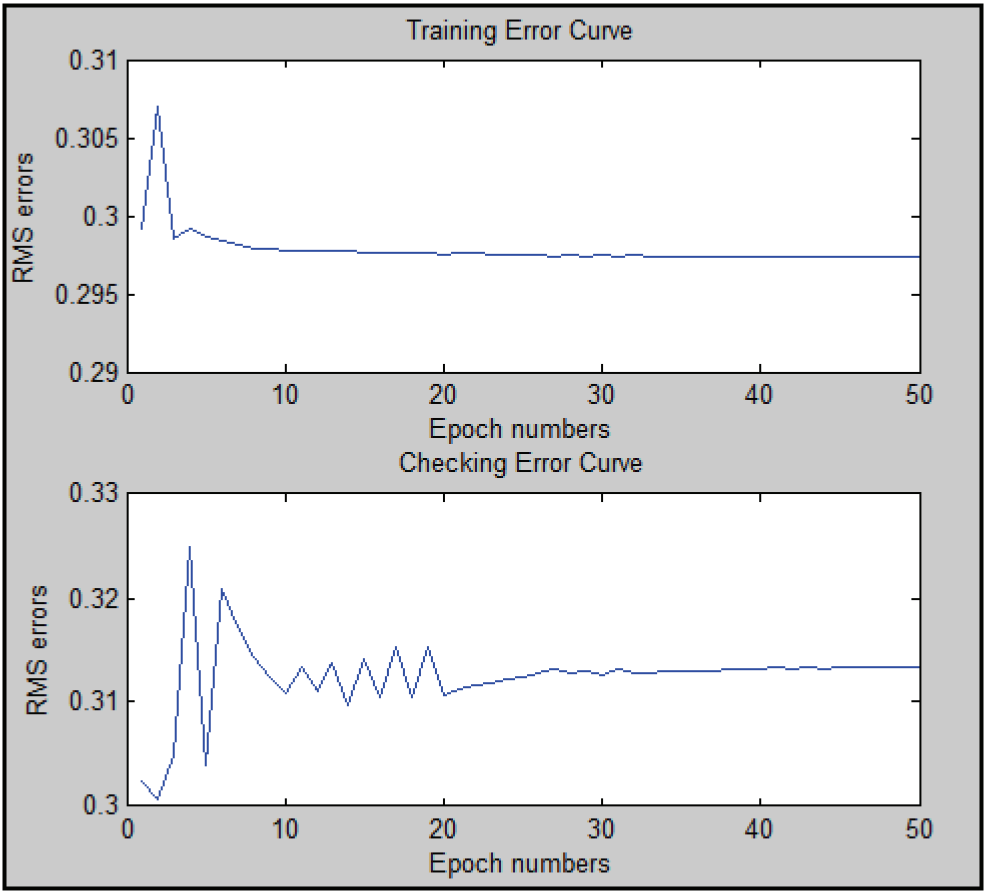


Fig. 6. Training and checking errors obtained by ANFIS for predicting C_{eq30}.

The final member functions of the five inputs are changed through supervised learning. In a real world domain, all the features described may have different levels of relevancy. Moreover, human-determined membership functions are seldom optimal in terms of producing desired outputs. Therefore, the training data are sufficient to provide the available input-output data necessary for fine-tuning the membership function. Figure 7 shows the final membership functions of the input parameters derived by training via the triangular membership function. Considerable changes happened in the final membership function after training especially for ultrafiltration rate (UFR) input. After the training process, the model is validated by comparing the predicted results against the experimental data. The validation tests between the predicted results and the actual results for both training and testing phases are summarized in table 5. The statistical analysis demonstrated that there is no statistically significant difference was found between the predicted and the measured values. The percentage of MAE and RMSE for testing phase is 0.44% and 0.61% respectively.

The same data set were used for predicting equilibrated urea concentration at 60 min (C_{eq60}) post-dialysis session. The C_{eq60} model achieved RMSE values of 0.2707 and 0.3125 for training and testing, respectively. The results obtained indicate that ANFIS is a promising

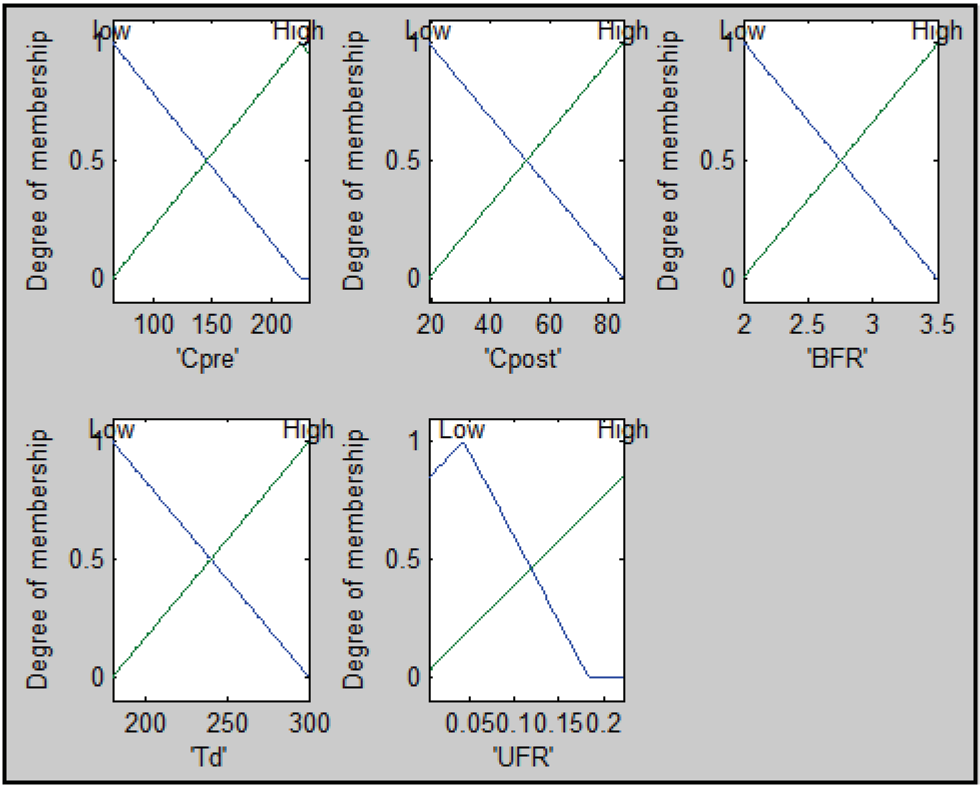


Fig. 7. The final membership functions of selected inputs for C_{eq30} predictor.

Agreement Comparison	$C_{eq30\text{-measured}}$ versus $C_{eq30\text{-ANFIS}}$	
	Training Phase	Testing Phase
Mean Absolute Error (MAE)	0.2383	0.2425
MAPE	0.0044	0.0044
Root mean square error (RMSE)	0.2978	0.3125
Normalized root mean square Error (NRMSE)	0.0040	0.0061
Median algebraic difference Δ	0.2347	0.0113
Median absolute difference |Δ|	0.6529	0.3143

Table 5. Validation tests between the measured C_{eq30} as a reference and the predicted one by the ANFIS system.

tool for predicting equilibrated urea concentration at 30 min (C_{eq30}) and 60 min (C_{eq60}) post-dialysis session. Both C_{eq30} and C_{eq60} models had very low MAE and RMSE values for both training and testing. This model was conducted to determine how the equilibrated urea concentration (C_{eq}) could be predicted without having to take a final urea sample an hour after the patient had completed the dialysis session.

7. Conclusion

Predictive modeling is the process of identifying a model of an unknown or complex process from numerical data. Due to the inherent complexity of many real processes, conventional modeling techniques have proved to be too restrictive. Recently, the hybrid approach to predictive modeling has become a popular research focus. A novel hybrid system combining different soft computing paradigms such as neural networks and fuzzy systems has been developed for predictive modeling of dialysis variables in order to estimate the equilibrated dialysis dose ($_{eq}Kt/V$), without waiting for 30-60 min post-dialysis to get the equilibrated urea sample which is inconvenient for patients and costly to the dialysis unit. The aim of using a neuro-fuzzy network is to find, through learning from data, a fuzzy model that represents the process underlying the data. In neuro-fuzzy models, connection weights, propagation and activation functions differ from common neural networks. Although there are a lot of different approaches, the term neuro-fuzzy is restricted to systems which display the following properties:

- A neuro-fuzzy system is a fuzzy system that is trained by learning algorithm (usually) derived from neural network theory. The (heuristic learning procedure operates on local information, and causes only local modifications in the underlying fuzzy system. The learning process is not knowledge based, but data driven.
- A neuro-fuzzy system can be viewed as a special 3-layer feedforward neural network. The units in this network use *t*-norms or *t*- cononrms instead of the activation functions usually used in neural networks.

The first layer represents input variables, the middle (hidden) layer represents fuzzy rules and the third layer represents output variables. Fuzzy sets are encoded as (fuzzy) connection weights. Some neuro-fuzzy models use more than 3 layers, and encode fuzzy sets as activation functions. In this case, it is usually possible to transform them into 3-layer architecture. This view of fuzzy systems illustrates the data flow within the system and its parallel nature. However this neural network view is not a prerequisite for applying a learning procedure, it is merely a convenience.

- A neuro-fuzzy system can always (i.e. before, during and after learning) be interpreted as a system of fuzzy rules. It is both possible to create the system out of training data from scratch, and it is possible to initialize it by prior knowledge in form of fuzzy rules.
- The learning procedure of a neuro-fuzzy system takes the semantical properties of the underlying fuzzy system into account. This result in constraints on the possible modifications of the system's parameters.
- A neuro-fuzzy system approximates an n-dimensional (unknown) function that is partially given by the training data. The fuzzy rules encoded within the system represent vague samples, and can be viewed as vague prototypes of the training data. A neuro-fuzzy system should not be seen as a kind of (fuzzy) expert system, and it has nothing to do with fuzzy logic in the narrow sense.

Besides accuracy, model transparency is the most important goal of the proposed modeling methodology, which supports this feature at different levels. The first level of transparency supported is the ability to represent the knowledge characterizing the relations between the data in a natural manner, e.g. as a series of linguistic fuzzy rules, which is common to all neuro-fuzzy approaches. The second level of transparency required to models derived from data is a simple structure. Indeed, the use/reliability of a fuzzy model can become limited with too many fuzzy rules. Hence a compromise between model complexity and model accuracy should be found by identifying parsimonious structures, which aid handling and comprehension of the fuzzy rule base. Simple structures also could be beneficial in improving the model generalization capabilities. A further level of model transparency is a truly linguistic representation of the produced fuzzy rules. By supporting these various levels of transparency, the proposed neuro-fuzzy modeling methodology significantly aids the process of knowledge discovery and model validation. With a data-driven methodology, like the proposed one, fundamental to the success of the modeling process is the availability of good empirical data. When data is limited and/or poorly distributed, the modeling task can easily become unmanageable. This reinforces the importance of the human for injecting a priori knowledge, expert judgment and intuition in to the modeling process. The developed methodology enables the incorporation of a-priori knowledge into the modeling process so as to compensate for the lack of data. When a-priori knowledge provided by the expert takes the form of qualitative descriptions of the process underlying the data, it can be easily inserted into the modeling process by building and pre-weighting the neuro-fuzzy network, giving an initial model which can be later refined in the presence of training data. These are the more appealing features of the proposed methodology.

8. References

Azar, A.T.; Kandil, A.H.; Wahba, K. and Massoud, W. (2008a). Prediction Of Post-dialysis Blood Urea Concentration Using Artificial Neural Network. *Proceedings of 2nd International conference on Advanced Control Circuits & systems (ACCS'08)*, March 30-April 2, 2008, Cairo, Egypt.

Azar, A.T.; Kandil, A.H.; Wahba, K. and Massoud, W. (2008b). Neuro-Fuzzy System for Post-dialysis Urea Rebound Prediction, *Proceedings of IEEE 4th Cairo International Biomedical Engineering Conference (CIBEC'08)*, Dec. 18-20, 2008, Cairo, Egypt.

Azar, A.T.; Kandil, A.H.; Wahba, K. (2009a). Artificial Intelligence for Prediction of Equilibrated Dialysis Dose without Intradialytic Sample. *Saudi J Kidney Dis Transplant*, to be published.

Azar, A.T. (2009b). *Adaptive Neuro-Fuzzy System for Hemodialysis Treatment Process*, Ph.D. dissertation, Dept. Sys. & Biomed. Eng., Cairo Univ., Egypt.

Berenji, R.H. (1992). A reinforcement learning-based architecture for fuzzy logic control. *International Journal of Approximate Reasoning*, Vol. 6, Issue 2.

Bersini H.; Nordvik, J.P & Bonarini, A. (1993). A simple direct adaptive fuzzy controller derived from its neutral equivalent, *Proceedings of 2nd IEEE International Conference on Fuzzy Systems*, Vol. 1, pp. 345-350.

Bhaskaran, S.; Tobe, S.; Saiphoo, C.; Moldoveanu, A.; Raj, D.S. and Manuel, M.A. (1997). Blood urea levels 30 minutes before the end of dialysis are equivalent to equilibrated blood urea, *ASAIO J*, Vol. 43, No. 5, pp. M759-762.

Brown M. & Harris C.J. (1995). *Neurofuzzy Adaptive Modelling and Control*. Prentice Hall, 1st edition, Hemel Hempstead.

Buckley, J.J. & Eslami, E. (1996). Fuzzy Neural Networks: Capabilities. In *Fuzzy Modeling: Paradigms and Practice*, Pedrycz W, ed., pp. 167-183, Kluwer, Boston,

Canaud, B.; Bosc, J.Y; Leblanc, M.; Garred, L.; Vaussenat, F.; Bonardet, A. and Mion, C. (1997). A simple and accurate method to determine equilibrated post-dialysis urea concentration, *Kidney Int*, Vol. 51, No. 6, pp. 2000-2005.

Castillo, O & Melin, P. (2001). *Soft Computing for Control of Non-Linear Dynamical Systems*. 1st edition, Physica-Verlag Heidelberg, Germany.

Cho, K. & Wang, B. (1996). Radial basis function based adaptive fuzzy systems and their application to system identification and prediction. *Fuzzy sets and systems*, Vol. 83, No. 3, pp. 325–339.

Cox, E. (1994). *The Fuzzy Systems Handbook*. AP Professional - New York.

Daugirdas, J.T. (1993). Second generation logarithmic estimates of single-pool variable volume Kt/V: An analysis of error. *JAm Soc Nephrol*, Vol. 4, pp. 1205-1213.

Daugirdas, J.T.; Burke, M.S.; Balter, P.; Priester-Coary, A. and Majka, T. (1996). Screening for extreme postdialysis urea rebound using the Smye method: patients with access recirculation identified when a slow flow method is not used to draw the postdialysis blood. *Am J Kidney Dis*, Vol. 28, No. 5, pp. 727-731.

Daugirdas, J.T & Depner, T.A. (1994). A nomogram approach to hemodialysis urea modeling. *Am J Kidney Dis*, Vol. 40, pp. 23-33.

Daugirdas, J.T & Schneditz, D. (1995). Overestimation of hemodialysis dose depends on dialysis efficiency by regional blood flow but not by conventional two pool urea kinetic analysis. *ASAIO J*, Vol. 41, pp. M719-M724.

Depner, T.A. (1994). Assessing Adequacy Of Hemodialysis Urea Modeling. *Kidney Int*, Vol., 45, No. 5, pp. 1522-1535.

Depner, T.A. (1999). History of Dialysis Quantitation, *Semin Dial*, Vol. 12, No.1, pp. S14- S19.

Fuller, R. (2000). *Introduction to Neuro-Fuzzy Systems*. Advances in Soft Computing Series, Springer-Verlag, Berlin/Heildelberg.

Gotch, F.A. & Sargent, J.A. (1985). A Mechanistic Analysis of the National Cooperative Dialysis Study, *Kidney Int*, Vol. 28, No. 3, pp. 526-538.

Guh, J.; Yang, C.; Yang, J.; Chen, L. and Lai, Y. (1998). Prediction of equilibrated postdialysis BUN by an artificial neural network in high-efficiency hemodialysis. *Am J Kidney Dis*, Vol. 31, pp. 638- 646.

Haykin, S. (1998). *Neural Networks, A Comprehensive Foudation*. Second Edition, Prentice Hall.

Ichihashi, H. and Turksen, I. (1993). A neuro-fuzzy approach to data analysis of pairwise comparisons. Int. *Journal of Approximate Reasoning*, Vol. 9, No.3, pp. 227–248.

Jang, J.S.R. & Sun, C.T. (1993). Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems, *IEEE Trans. on Neural Networks*, Vol. 4, No. 1, pp. 156-159.

Jang, J.S.R. (1993). ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No.3, pp. 665–685.

Jang, J.S.R. & Sun, C.T. (1995). Neuro-Fuzzy Modeling and Control, *Proceedings of the IEEE*, Vol. 83, pp. 378-406.

Jang, J.S.R; Sun, C.T & Mizutani, E. (1997). *Neuro-Fuzzy and soft computin*. Prentice-Hall: Englewood Cliffs, NJ.

Jin, Y.; Von Seelen, W.; and Sendhoff, B. (1999). On Generating FC Fuzzy Rule Systems from Data Using Evolution Strategies. *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, Vol. 29, No. 6, pp. 829-845.

Lin, C.T. & Lee, C.S. (1991). Neural-Network-Based Fuzzy Logic Control and Decision Systems. *IEEE Trans. on Computers*, Vol. 40, No. 12, pp. 1320-1336.

Lin, C.T. and Lee, G. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Ed. Prentice Hall.

Maduell, F.; Garcia-Valdecasas, J.; Garcia, H.; Hernandez-Jaras, J.; Sigüenza, F.; del Pozo, C.; Giner, R.; Moll, R.; and Garrigos, E. (1997). Validation of different methods to calculate Kt/V considering post-dialysis rebound. *Nephrol Dial Transplant*, Vol. 12, No. 9, pp. 1928-1933.

Mamdani, E.H. & Assilian, S. (1975). An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller, *International Journal of Man-Machine Studies*, Vol. 7, No. 1, pp. 1-13.

Mehrotra, K., Mohan, C. K., and Ranka, S. (1997). *Elements of Artificial Neural Networks*. The MIT Press.

Nauck, D. (1994). A fuzzy perceptron as a generci model for neuro-fuzzy approaches. *Proceedings of Fuzzy-Systeme'94*, 2^{nd} GI-Workshop.

Nauck, D & Kruse, R. (1998a). A Neuro-Fuzzy Approach to Obtain Interpretable Fuzzy Systems for Function Approximation. *Proceedings of the IEEE International Conference on Fuzzy Systems*, Anchorage, AK, pp. 1106-1111.

Nauck, D & Kruse, R. (1998b). How the Learning of Rule Weights Affects the Interpretability of Fuzzy Systems, *Proceedings of the IEEE International Conference on Fuzzy Systems*, Anchorage, AK, Vol.2, pp. 1235- 1240.

Nauck, D. & Kruse, R. (1999). Neuro-fuzzy systems for function approximation. *Fuzzy Sets and Syst*., Vol. 101, pp. 261-271.

Nomura, H., Hayashi, I., and Wakami, N. (1992). A self-tuning method of fuzzy control by descent method. *Proceedings of IEEE International Conferenceon Fuzzy Systems*, pp. 203–210.

Royas, I.; Pomares, H.; Ortega, J.; and Prieto, A. (2000). Self-Organized Fuzzy System Generation from Training Examples, *IEEE Trans. on Fuzzy Systems*, Vol. 8, No. 1, pp. 23-36.

Ruspini, E., Bonissone, P., and Pedrycz, W. (1998). *Handbook of Fuzzy Computation*. Ed. Iop Pub/Inst of Physics.

Schneditz, D. & Daugirdas, J.T. (2001). Compartment effects in hemodialysis, *Semin Dial*, Vol. 14, No. 4, pp. 271-277.

Seng, T.L.; Khalib, M.B; and Yusof, R. (1999). Tuning of a Neuro-Fuzzy Controller by Genetic Algorithm. *IEEE Trans. on Systems, Man and Cybernetics-Part B, Cybernetics*, Vol. 29, No. 2.

Shi, Y. and Mizumoto, M. (2000a). A new approach of neurofuzzy learning algorithm for tuning fuzzy rules. *Fuzzy sets and systems*, 112(1):99–116.

Shi, Y. and Mizumoto, M. (2000b). Some considerations on conventional neuro-fuzzy learning algorithms by gradient descent method. *Fuzzy sets and systems*, Vol. 112, No. 1, pp. 51–63.

Smye, S.W.; Dunderdale, E.; Brownridge, G. and Will, E. (1994). Estimation of treatment dose in high-efficiency hemodialysis, *Nephron*, Vol. 67, No. 1, pp. 24-29.

Takagi, H. & Hayashi, I. (1991). NN-driven fuzzy reasoning. *International Journal of Approximate Reasoning*, Vol. 5, Issue 3.

Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans Syst Man Cybern*, Vol. 15, No. 1, pp. 116-132.

Tattersall, J.E; DeTakats, D., Chamney, P., Greenwood, RN. and Farrington, K. (1996). The post-hemodialysis rebound: Predicting and quantifying its effect on Kt/V. *Kidney Int*, Vol. 50, pp. 2094-2102.

Tsukamoto, Y. (1979). An approach to fuzzy reasoning method. In *Advances in fuzzy set theory and applications,* M.M. Gupta, R.K. Ragade and Yager R.R., Editors , Elsevier, North-Holland.

Wang, L. and Mendel, J. (1992). Back-propagation fuzzy system as nonlinear dynamic system identifiers. *Proceedings of IEEE International Conferenceon Fuzzy Systems*, pages 1409–1416.

Yager, R. and Filev, D. (1994). Generation of fuzzy rules by mountain clustering. *Journal of Intelligent Fuzzy Systems*, Vol. 2, No. 3, pp. 209–219.

Yashiro, M., Watanabe, H. and Muso, E. (2004). Simulation of post-dialysis urea rebound using regional flow model. *Clin Exp Nephrol*, Vol. 8, No. 2, pp. 139-45.

Yen, J. & Wang, L. (1999). Simplifying Fuzzy Rule-Based Models Using Orthogonal Transformation Methods. *IEEE Trans. on Systems, Man and Cybernetics-Part B: Cybernetics*, Vol. 29, No. 1, pp. 13-24.

Zadeh, L. (1965). Fuzzy sets. *Inf Cont*, Vol. 8, pp. 338–353.

6

A Hybrid Fuzzy System for Real-Time Machinery Health Condition Monitoring

Wilson Wang
Lakehead University
Canada

1. Introduction

Rotary machinery is widely used in various types of engineering systems ranging from simple electric fans to complex machinery systems such as aircraft. A reliable online condition monitoring system is very useful in industries both as a quality control scheme and as a maintenance tool. In quality control, the early detection of faulty components can prevent machinery performance degradation and malfunction. As a maintenance tool, machinery health condition monitoring enables the establishment of a maintenance program based on an early warning. This can be of great value in cases involving critical machines (e.g., airplanes, power turbines, and chemical engineering facilities), where an unexpected shutdown can have serious economic or environmental consequences.

Condition monitoring is an act of fault diagnosis by means of appropriate observations from different information carriers, such as temperature, acoustics, lubricant, or vibration. Vibration-based monitoring, however, is the most commonly used approach in industries because of its ease of measurement, which also will be used in this study.

Fault diagnosis is a sequential process involving two steps: representative feature extraction and pattern classification. Feature extraction is a mapping process from the measured signal space to the feature space. Representative features associated with the health condition of a machinery component (or subsystem) are extracted by using appropriate signal processing techniques. Pattern classification is the process of classifying the characteristic features into different categories. The classical approach, which is also widely used in industry, relies on human expertise to relate the vibration features to the faults. This method, however, is tedious and not always reliable when the extracted features are contaminated by noise. Furthermore, it is difficult for a diagnostician to deal with the contradicting symptoms if multiple features are used. The alternative is to use analytical tools (Li & Lee, 2005, Gusumano et al., 2002) and data-driven paradigms (Isermann, 1998). The latter will be utilized in this work because an accurate mathematical model is difficult to derive for a complex mechanical system, especially when it operates in noisy environments. Data-driven diagnostic classification can be performed by reasoning tools such as neural networks (Rish et al, 2005, Uluyol, 2006), fuzzy logic (Mansoori et al., 2007, Ishibuchi & Yamamoto, 2005), and neural fuzzy synergetic schemes (Wang, 2008, Uluyol et al., 2006).

Even though several techniques have been proposed in the literature for machinery condition monitoring, it still remains a challenge in implementing a diagnostic tool for real-world monitoring applications because of the complexity of machinery structures and operating conditions. When a monitoring system is used in real-time industrial applications, the critical issue is its reliability. Unreasonably missed alarms (i.e., the monitoring system cannot pick up existing faults) and false alarms (i.e., the monitoring system triggers an alarm because of noise instead of real faults) will seriously mitigate its validity. To tackle these challenges, the objective of this research work is to develop a new technique, an integrated classifier, for real-time condition monitoring in, especially, gear transmission systems. In this novel classifier, the monitoring reliability is enhanced by integrating the information of the object's future states forecast by a multiple-step predictor; furthermore, the diagnostic scheme is adaptively trained by a novel recursive hybrid algorithm to improve its convergence and adaptive capability.

This chapter is organized as follows: Section 2 describes integrated classifier, whereas the multiple-step predictor and monitoring indices are described in Section 3. Section 4 discusses the hybrid online training algorithm. In Section 5, the viability of the proposed integrated classifier is verified by experimental tests corresponding to different gear conditions.

2. Diagnostic system

The diagnostic classifier is used to integrate the selected features obtained by implementing appropriate signal processing techniques. The purpose is to make a more positive assessment of the health condition of the mechanical component (or subsystem) of interest. The diagnostic reliability in this suggested classifier will be enhanced by implementing the future (multi-step-ahead) states of the object's conditions. The forecasting in this integrated classifier is performed for input variables so as to make it easier to track the error sources in diagnostic operations.

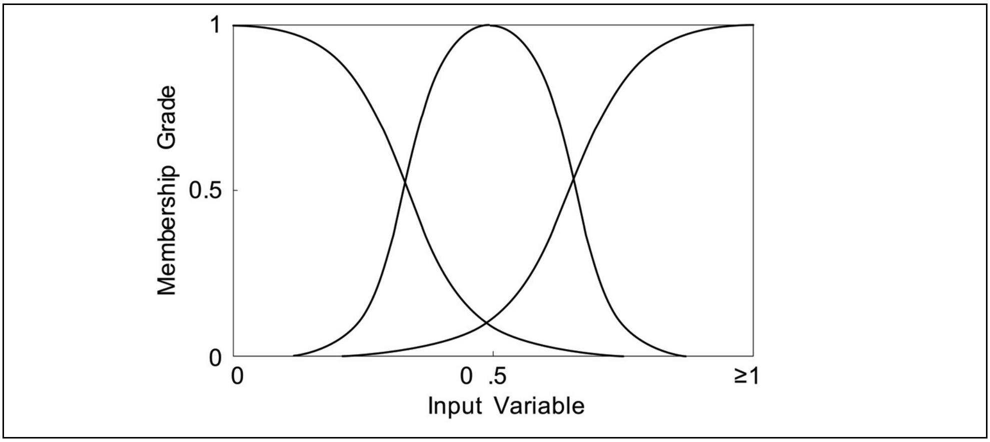


Fig. 1. The initial membership functions (MFs) for the input state variables.

The developed classifier is an NF paradigm which is able to facilitate the incorporation of diagnostic knowledge from expertise and to extract new knowledge in operations by online/offline training. The diagnostic classification is performed by fuzzy logic (Jang 1993), whereas an adaptive training algorithm, as discussed in Section 4, is utilized to fine-tune the fuzzy system parameters and structures. The conditions of each object (machinery component or subsystem) are classified into three categories: *healthy* (C_1), *possible (initial) damage* (C_2), and *damage* (C_3), respectively. $\{x_1, x_2, \ldots, x_n\}$ are the input variables at the current time step. Three membership functions (MFs), *small*, *medium*, and *large*, are assigned to each input variable with the initial states as shown in Fig. 1 where the fuzzy completeness (or the minimum fuzzy membership grade) is at 50%.

The diagnostic classification, in terms of the diagnostic indicator y, is formulated in the following form:

$$\Re_j : \text{ If } (x_1 \text{ is } A_{1j}) \text{ and } (x_2 \text{ is } A_{2j}) \text{ and} \ldots \text{and } (x_n \text{ is } A_{nj}) \Rightarrow (y \subset S_j \text{ with } w_j) \quad (1)$$

where A_{ij} are MFs; $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, m$, m denotes the number of rules; S_j represents one of the states C_1, C_2 or C_3, depending on the values of the diagnostic indicator.

When multiple features (input indices) are employed for diagnostic classification operations, the contribution of each feature combination (association) to the final decision depends, to a large degree, on the situation under which the diagnostic decision is made. Such a contribution is characterized by a weight factor w_j which is related to the feature association in each rule. The initial values of these rule weights are chosen to be unity; That is, all input state variables have initially assumed to have identical importance or robustness to the overall diagnostic output.

Similarly, the diagnostic classification based on the predicted monitoring indices, $\{x'_1, x'_2, \ldots, x'_n\}$, is formulated as:

$$\Re_j : \text{If } (x'_1 \text{ is } A_{1j}) \text{ and } (x'_2 \text{ is } A_{2j}) \text{ and} \ldots \text{and } (x'_n \text{ is } A_{nj}) \Rightarrow (y' \subset S_j \text{ with } w_j) \quad (2)$$

where y' is the diagnostic indicator based on forecast input variables.

The number of rules is associated with the diagnostic reasoning operations of input state variables. In general, if all monitoring indices are *small*, then the object is considered healthy (C_1). Otherwise, the object is possibly damaged. In this case, the diagnostic classification indicator y represents *faulty* condition only. Different feature association (rule) corresponds to a different confidence grade w_j in diagnosis. Fig. 2 schematically shows the network architecture of this integrated classifier. Unless specified, all the network links have unity weights.

The input nodes in layer 1 transmit the monitoring indices $\{x_1, x_2, \ldots, x_n\}$ or their forecast future values $\{x'_1, x'_2, \ldots, x'_n\}$ to the next layer. These two sets of monitoring indices are input to the network and processed separately.

Each node in layer 2 acts as a MF, which can be either a single node that performs a simple activation function or multilayer nodes that perform a complex function. The nodes in layer 3 perform the fuzzy T-norm operations. If a product operator is used, the firing strength of rule $\Re_j$ is

$$\eta_j = \prod_{i=1}^{n} A_{ij}(x_i) \quad (3)$$

$$\eta'_j = \prod_{i=1}^{n} A_{ij}(x'_i) \quad (4)$$

where $A_{ij}(\bullet)$ denote MF grades.

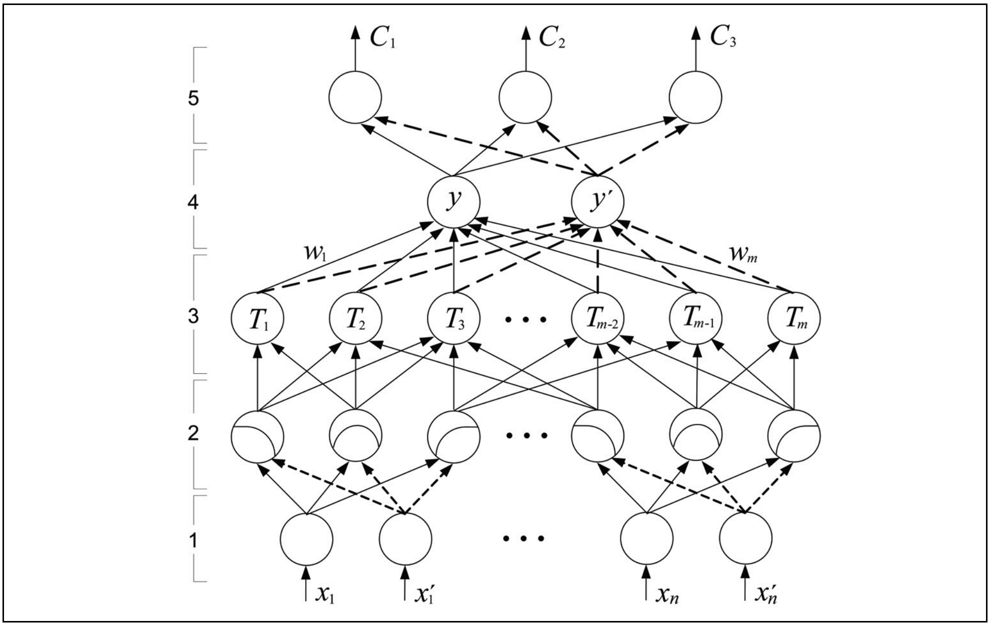


Fig. 2. The network architecture of the proposed integrated classifier.

Defuzzification is undertaken in layer 4. By normalization, the faulty diagnostic indicator will be

$$y = \frac{\sum_m \eta_j w_j}{\sum_m \eta_j} \tag{5}$$

Similarly, the fault diagnostic indicator based on forecast inputs will be

$$y' = \frac{\sum_m \eta'_j w_j}{\sum_m \eta'_j} \tag{6}$$

The states of the diagnostic indicator y (or y') are further classified into three categories:

$$\begin{cases} \text{If } 0 \le y \le 0.33 & \Rightarrow \textit{Healthy}\,(C_1) \\ \text{If } 0.33 < y \le 0.66 & \Rightarrow \textit{Possibly damaged}\,(C_2) \\ \text{If } 0.66 < y \le 1 & \Rightarrow \textit{Damaged}\,(C_3) \end{cases}.$$

The final decision regarding the health condition of the object of interest is made by:

a) If ($y \subset C_1$ and $y' \subset C_1$) or ($y \subset C_2$ and $y' \subset C_1$) then (the object is healthy C_1)
b) If ($y \subset C_3$ and $y' \subset C_3$) or ($y \subset C_2$ and $y' \subset C_3$) then (the object is damaged C_3) (7)
c) Otherwise, (the object is possibly damaged C_2).

3. Prediction of monitoring indices

3.1 Monitoring indices

In general, most machinery defects are related to transmission systems, mainly for gears and bearings. In this work, gears are used as an example to illustrate how to apply the proposed integrated classifier for machinery condition monitoring. In operations, the fault diagnosis of a gear train is conducted gear by gear. Because the measured vibration is an overall signal contributed from various vibratory sources, the primary step is to differentiate the signal specific to each gear of interest by using a synchronous average filter (Wang et al., 2001). By this filtering process, the signals which are non-synchronous to the rotation of the gear of interest (e.g., those from bearings, shafts and other gears) are filtered out. As a result, each gear signal is computed and represented in one full revolution, called the *signal average* which will be used for advanced analysis by other signal processing techniques.

Several techniques have been proposed in the literature for gear fault detection. However, because of the complexity in the machinery structures and operating conditions, each fault detection technique has its own advantages and limitations, and is efficient for some specific application only (Wang et al., 2001). Consequently, the selected features for fault diagnostics should be robust, that is, sensitive to component defects but insensitive to noise (i.e., the signal not carrying information of interest). In this case, three features from the information domains of energy, amplitude, and phase are employed for the diagnosis operation:

1. Wavelet energy function, using the overall residual signal which is obtained by bandstop filtering out the gear mesh frequency $f_R N$ and its harmonics, where f_R is the rotation frequency (in Hz) of the gear of interest and N is the number of teeth of the gear;
2. Phase demodulation (McFadden, 1986), using the signal average;
3. Beta kurtosis, using the overall residual signal.

The details of these reference functions are listed in Appendix A.

Based on the derived reference functions, the monitoring indices are determined to quantify the feature characteristics. Each index is a function of two variables, magnitude and position. The magnitude of an index is determined as the normalized relative maximum amplitude value of the corresponding reference function; the position is where the maximum amplitude is located. Usually, the maximum amplitude positions in these reference functions do not coincide exactly due to the phase lags in signal processing. Based on simulation and test observations, an *influence window* is defined as a period of four tooth periods in this case. Correspondingly, if all indices are located within one influence window, one set of inputs $\{x_1, x_2, x_3\}$ is given to the classifier. Otherwise, if three indices are not within one influence window, the object has no fault or has more than one defect; more than one set of inputs should be provided to the classifier. For example, if x_3 does not fall within the influence window determined by x_1 and x_2, two sets of inputs will be given to the monitoring classifier: The first input vector is $\{x_1, x_2, x_3\}$, where x_3 is computed over the influence window determined by both x_1 and x_2; The second input vector is $\{x_1, x_2, x_3\}$, where x_1 and x_2 are determined over the influence window around x_3.

Fig. 3 illustrates an example of the reference functions corresponding to a healthy gear with 41 teeth. Fig. 3a shows part of the original vibration signal measured from the experimental setup to be illustrated in Section 5. Fig. 3b represents the signal average of the gear of interest, which is obtained by synchronous average filtering; each wave represents a tooth period. Figs. 3c to 3e represent the resulting reference functions of the wavelet energy, beta kurtosis, and phase modulation, respectively. It is seen that no specific irregularities can be found from these reference functions for this healthy gear.

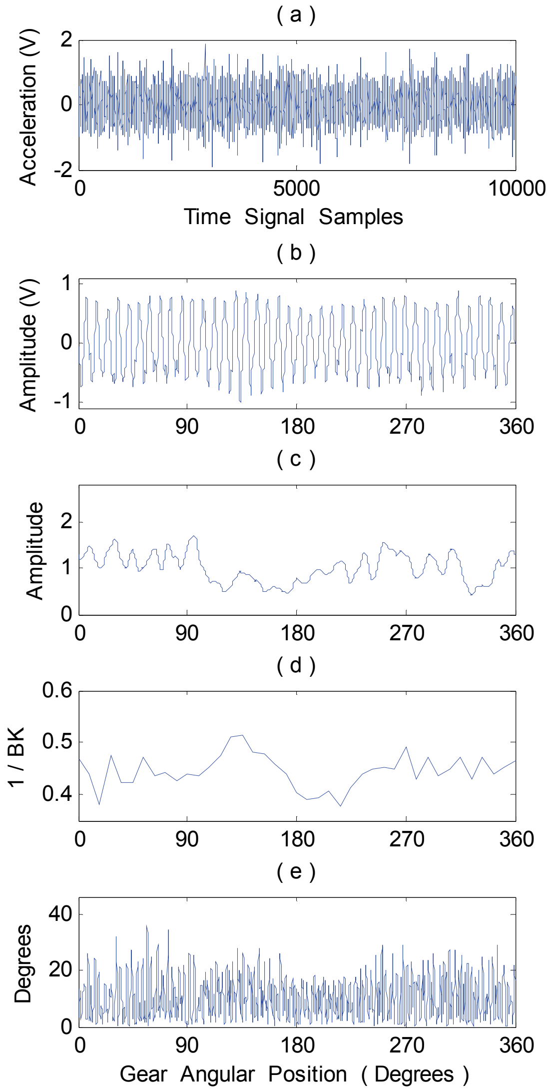


Fig. 3. Processing results for a healthy gear: (a) Part of the original vibration signal; (b) Signal average; (c) Wavelet reference function; (d) Beta kurtosis reference function; (e) Phase modulation reference function.

Fig. 4 shows the processing results corresponding to a cracked gear with 41 teeth. It is impossible to recognize the gear damage from the original signal (Fig. 4a). A little signature irregularity can be recognized around 200° in the signal average graph (Fig. 4b). However, this gear damage can be identified clearly from the proposed reference functions (Figs. 4c to 4e). Although the maximum peak positions are little different from one graph to another, these peaks occur within one influence window (four tooth periods in this case).

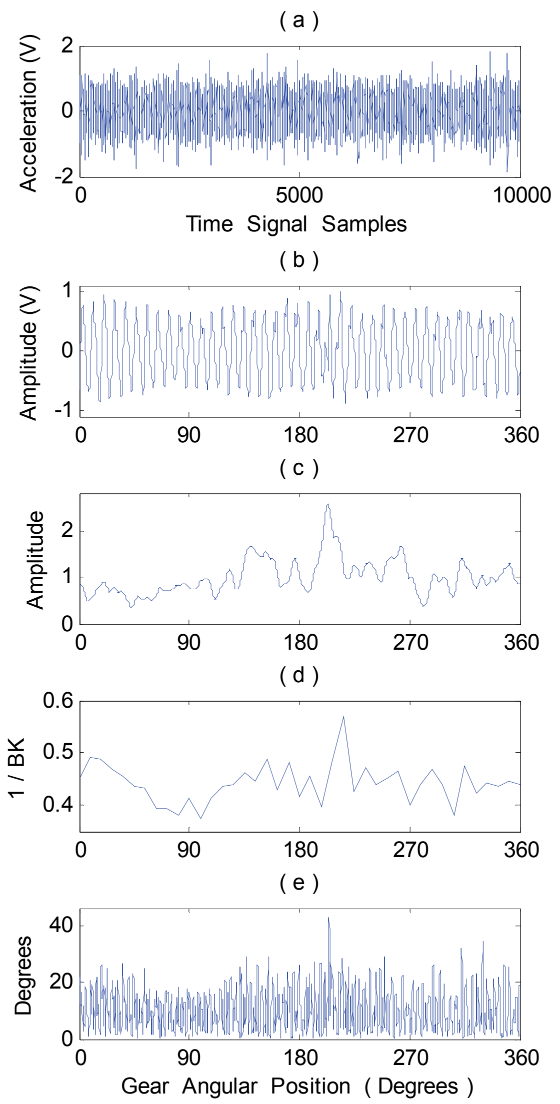


Fig. 4. Processing results for a cracked gear: (a) Part of the original vibration signal; (b) Signal average; (c) Wavelet reference function; (d) Beta kurtosis reference function; (e) Phase modulation reference function.

Fig. 5 illustrates the processing results for a chipped gear (with 41 teeth). Some signature irregularity can be recognized around 200° in the signal average graph (Fig. 5b) due to this gear tooth damage. However, this defect can be clearly identified from other three reference functions (Figs. 5c to 5e), and the monitoring indices are located within one influence window (four tooth periods).

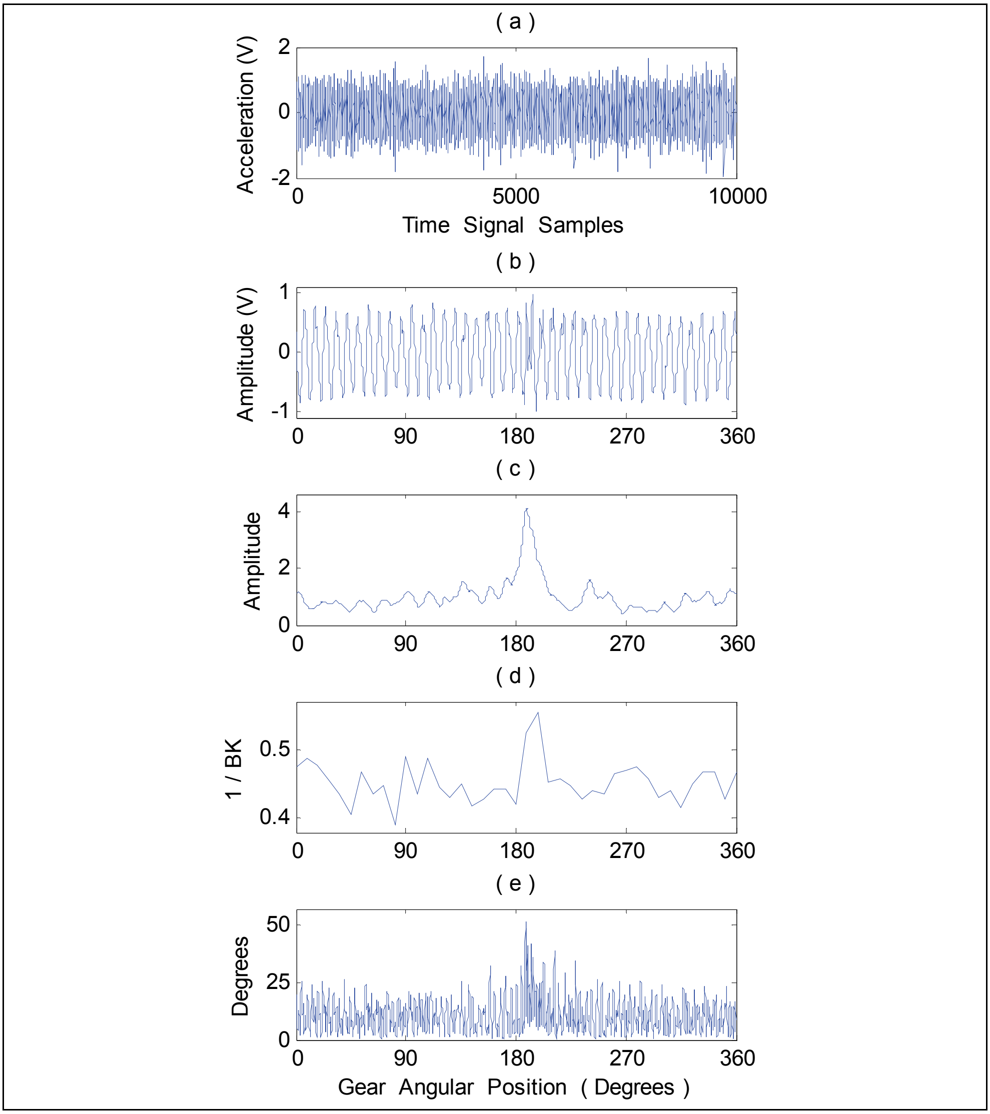


Fig. 5. Processing results for a chipped gear: (a) Part of the original vibration signal; (b) Signal average; (c) Wavelet reference function; (d) Beta kurtosis reference function; (e) Phase modulation reference function.

3.2 Forecasting of the monitoring indices

System state forecasting is the process to predict the future states in a dynamic system based on available observations. Several techniques have been suggested in the literature for time series forecasting. The classical methods are the use of stochastic models (Chelidze & Cusumano, 2004), which are usually difficult to derive for mechanical systems with complex structures. More recent research on time series forecasting has focused on the use of data-driven paradigms, such as neural networks and neural fuzzy schemes (Tse & Atherton, 1999, Pourahmadi, 2001). In this work, the multi-step-ahead prediction of the input variables (indices) is performed by the use of a predictor as suggested in (Wang & Vrbanek, 2007), whose effectiveness has been verified: it can capture and track the system's dynamic characteristics quickly and accurately, and it outperforms to other related classical forecasting schemes.

Given a monitoring index x_1, or x_2, or x_3, if $\{v_0 \ v_{-r} \ v_{-2r} \ v_{-3r}\}$ represent its current and previous three states with an interval of *r* steps, the *r*-step-ahead state v'_{+r} is estimated by a TS-1 fuzzy formulation:

$$\begin{aligned} \Re_j: \quad & \text{If } (v_0 \text{ is } B_{0k}) \text{ and } (v_{-r} \text{ is } B_{1k}) \text{ and } (v_{-2r} \text{ is } B_{2k}) \text{ and } (v_{-3r} \text{ is } B_{3k}) \\ & \text{then } v'_{+r} = c_0^j v_0 + c_1^j v_{-r} + c_2^j v_{-2r} + c_3^j v_{-3r} + c_4^j \end{aligned} \tag{8}$$

where $B_{\bullet}$ are MFs, c_i^j are constants, i = 0, 1, ..., 3; j = 1, 2, . . ., 16; k = 1, 2. Fig. 6 illustrates its fuzzy reasoning architecture.

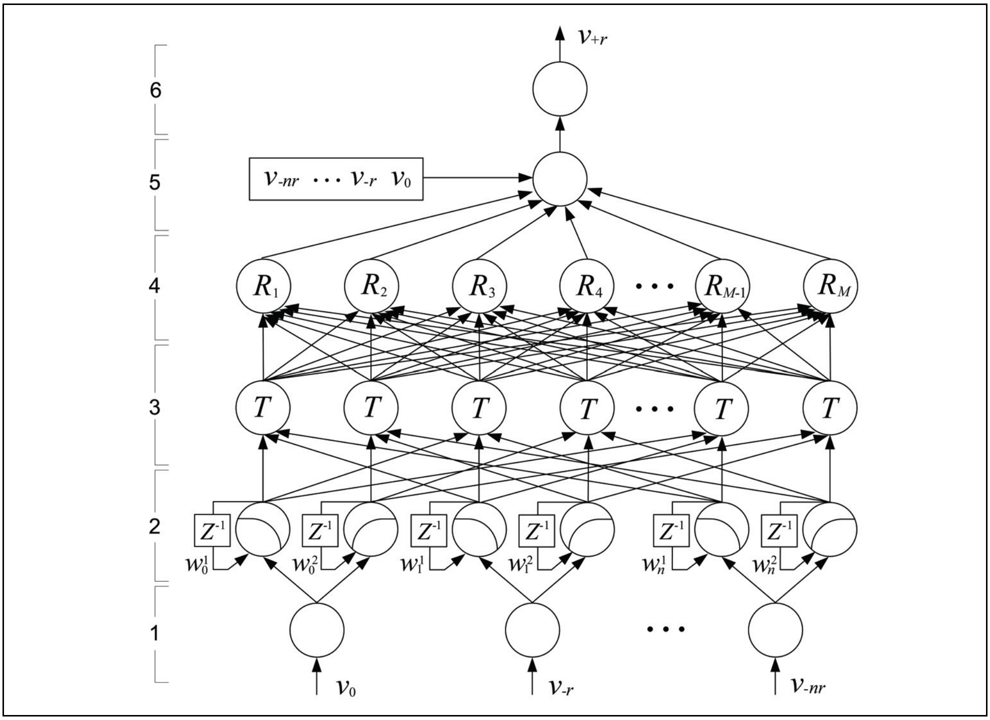


Fig. 6. The network architecture of the multi-step predictor.

This NF predictor has a weighted feedback link to each node in layer 2 to deal with time explicitly as opposed to representing temporal information spatially. The context units copy the activations of output nodes from the previous time step, and allow the network to memorize clues from the past, which forms a context for current processing. This function of recurrent networks is valuable for predictors with limited and step inputs (i.e., $r > 1$), to provide more information to the network so as to improve forecasting accuracy. If two sigmoid MFs are assigned to each input variable, the node output at the *k*th process step will be

$$\mu_{B_i^j}(V_{-ir}) = \frac{1}{1+\exp[-a_i^j(V_{-ir}-b_i^j)]} \tag{9}$$

$$V_{-is} = v_{-ir}^{(k)} + w_i^m \mu_{B_i^j}(v_{-ir}^{(k-1)}) = v_{-is}^{(k)} + \frac{w_i^m}{1+\exp[-a_i^j(v_{-ir}^{(k-1)}-b_i^j)]} \tag{10}$$

where m = 1, 2; i = 0, 1, . . ., n. $v_{-ir}^{(k)}$ and $v_{-ir}^{(k-1)}$ are, respectively, the input v_{-ir} at the *k*th and (*k*-1)th time steps, where k = 1, 2, . . ., K, K is the total number of time steps (or training data sets). If a max-product operator is applied in layer 3, and a centroid method is used for defuzzification in layer 5, by some related fuzzy operations, the predicted output v'_{+r} can be determined by

$$v'_{+r} = \sum_{j=1}^{16} \overline{\mu}_j (c_0^j v_0 + c_1^j v_{-r} + c_2^j v_{-2r} + c_3^j v_{-3r} + c_4^j) \tag{11}$$

where $\overline{\mu}_j = \frac{\mu_j}{\sum_{j=1}^{16}\mu_j}$ denotes the normalized rule firing strength, and μ_j is the firing strength of the *j*th rule.

The fuzzy system parameters are trained by using a hybrid algorithm: that is, the premise parameters in the MFs $B_\bullet$ are trained by a real-time recurrent training algorithm whereas the consequent parameters c_i^j in (8) are updated by least squares estimate (LSE). Details about the training algorithm can be found in (Wang, 2008).

4. Online training of the diagnostic classifier

The developed diagnostic classifier should be optimized in order to achieve the desired input-output mapping. Several training algorithms have been proposed in the literature for NF-based classification schemes (Figueiredo et al., 2004, Castellano et al., 2004). In offline training, representative data should cover all of the possible application conditions (Korbicz et al., 2004); such a requirement is usually difficult to achieve in real-world machinery applications because most machinery operates in noisy and uncertain environments. Furthermore, machinery dynamic characteristics may change suddenly, for instance, just after repair or regular maintenance. Therefore, an adaptive training algorithm is preferred in time-varying systems to accommodate different machinery conditions (Wang & Lee, 2002). In this case, a hybrid method based on recursive Levenberg-Marquet (LM) and LSE will be adopted to train the integrated classifier. Such a training approach possesses randomness that may help to escape certain local minima.

4.1 Training the premise MF parameters

The nonlinear premise MF parameters will be trained by adopting the recursive LM method. The general LM algorithm possesses quadratic convergence close to a minimum. Its convergence property is still reasonable, even if the initial estimates are poor. In addition, the LM algorithm has been proven globally convergent in many applications by properly choosing the step factors.

For a training data pair $\left\{\boldsymbol{x}^{(p)}, \boldsymbol{d}^{(p)}\right\}$, the inputs are $\boldsymbol{x}^{(p)} = \left\{x_1^{(p)}, x_2^{(p)}, x_3^{(p)}\right\}$, $p = 1, 2, \ldots, P$; $\boldsymbol{d}^{(p)}$ are the desired outputs {0, 0.5, 1} as $\boldsymbol{x}^{(p)}$ belongs to C_1, C_2 and C_3, respectively. The error function with respect to adjustable MF parameters $\boldsymbol{\theta}_p$ at the current time instant, p, is

$$E(\boldsymbol{\theta}_p) = \frac{1}{2}\sum_{p=1}^{P}\left[y_p(\boldsymbol{\theta}_p) - d_p\right]^2 = \frac{1}{2}\sum_{p=1}^{P} r_p^2(\boldsymbol{\theta}_p) = \boldsymbol{r}_p^T(\boldsymbol{\theta}_p)\boldsymbol{r}_p(\boldsymbol{\theta}_p) \tag{12}$$

where $y_p(\boldsymbol{\theta}_p)$ is the pth output determined by Eq. (5). $p = 1, 2, \ldots, P$; d_p is the desired output. To simplify expressions, the variable $\boldsymbol{\theta}_p$ is dropped in the related terms in this section. $\boldsymbol{r}_p$ is the error vector that can be either linear or nonlinear. By taking the Taylor series expansion and neglecting higher order terms,

$$\boldsymbol{\theta}_{p+1} \approx \boldsymbol{\theta}_p + \lambda(\boldsymbol{J}_p^T\boldsymbol{J}_p + \eta\boldsymbol{I})^{-1}\boldsymbol{J}_p^T\boldsymbol{r}_p = \boldsymbol{\theta}_p + (1-\alpha)\boldsymbol{H}_p^{-1}\boldsymbol{J}_p^T\boldsymbol{r}_p \tag{13}$$

where $\boldsymbol{J}_p \in \mathrm{R}^{N\times Z}$ denotes the Jacobian matrix; Z is the dimension (or the number of adjustable parameters) of $\boldsymbol{\theta}_p$; $\boldsymbol{H}_p \in \mathrm{R}^{Z\times Z}$ is the modified Hessian matrix; $\boldsymbol{I} \in \mathrm{R}^{Z\times Z}$ is an identity matrix; $\lambda = 1-\alpha$ is the learning rate, and α is the forgetting factor.

The Hessian matrix can be expressed as

$$\boldsymbol{H}_p = \alpha\boldsymbol{H}_{p-1} - (1-\alpha)(\boldsymbol{J}_p^T\boldsymbol{J}_p + \eta\boldsymbol{I}) \tag{14}$$

In implementation, instead of computing the $Z\times Z$ matrix $\eta\boldsymbol{I}$ at each time step, a diagonal element is added at each time step

$$\boldsymbol{H}_p = \alpha\boldsymbol{H}_{p-1} - (1-\alpha)(\boldsymbol{J}_p^T\boldsymbol{J}_p + Z\eta\boldsymbol{\Lambda}) \tag{15}$$

where $\boldsymbol{\Lambda} \in \mathrm{R}^{Z\times Z}$ has only one nonzero element located at $\{p \bmod(Z)+1\}$ diagonal position:

$$\Lambda_{ii} = \begin{cases} 1, & if\ \ i = \{p \ \bmod(Z)+1\} \\ 0, & otherwise \end{cases} \tag{16}$$

Correspondingly, (15) can be rewritten as

$$\boldsymbol{H}_p = \alpha\boldsymbol{H}_{p-1} - (1-\alpha)[\boldsymbol{U}\boldsymbol{V}^{-1}\boldsymbol{U}^T] \tag{17}$$

where $\boldsymbol{U}$ is a $Z\times 2$ matrix whose first column is $\boldsymbol{J}_p$ and second column consists of a $Z\times 1$ vector with one element of 1 at the position of $\{p \bmod(Z)+1\}$

$$\boldsymbol{U}^T(\boldsymbol{\theta}_p) = \begin{bmatrix} \boldsymbol{J}_p^T \\ 0 \cdots 0\ 1\ 0 \cdots 0 \end{bmatrix}, \text{ and } \boldsymbol{V}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & Z\eta \end{bmatrix}.$$

The computation of $\boldsymbol{H}_p^{-1}$ in (13) is time consuming, and is not suitable for real-time applications. To solve this problem, Eq. (13) is rewritten as

$$\begin{aligned}\boldsymbol{\theta}_{p+1} = \boldsymbol{\theta}_p + (1-\alpha)\boldsymbol{H}_p^{-1}\boldsymbol{J}_p^T\boldsymbol{r}_p = \boldsymbol{\theta}_p + (1-\alpha)\Big\{(\alpha\boldsymbol{H}_{p-1})^{-1} - (\alpha\boldsymbol{H}_{p-1})^{-1}(1-\alpha)\boldsymbol{U} \times \\ \left[\boldsymbol{V} + \boldsymbol{U}^T(\alpha\boldsymbol{H}_{p-1})^{-1}(1-\alpha)\boldsymbol{U}\right]^{-1}\boldsymbol{U}^T(\alpha\boldsymbol{H}_{p-1})^{-1}\Big\}\boldsymbol{J}_p^T\boldsymbol{r}_p\end{aligned} \tag{18}$$

Based on the matrix inversion formula and by some manipulations, Eq. (18) becomes

$$\boldsymbol{\theta}_{p+1} = \boldsymbol{\theta}_p + (1-\alpha)\left\{\alpha\boldsymbol{H}_{p-1} + (1-\alpha)\boldsymbol{U}\boldsymbol{V}^{-1}\boldsymbol{U}^T\right\}^{-1}\boldsymbol{J}_p^T\boldsymbol{r}_p \tag{19}$$

The recursive LM algorithm can be represented by

$$\boldsymbol{\theta}_{p+1} = \boldsymbol{\theta}_p + \boldsymbol{\Phi}_p\boldsymbol{J}_p\boldsymbol{r}_p \tag{20}$$

$$\boldsymbol{\Phi}_p = \frac{1}{\alpha}\left[\boldsymbol{\Phi}_{p-1} - \frac{\boldsymbol{\Phi}_{p-1}\boldsymbol{U}\boldsymbol{U}^T\boldsymbol{\Phi}_{p-1}}{\alpha\boldsymbol{V} + \boldsymbol{U}^T\boldsymbol{\Phi}_{p-1}\boldsymbol{U}}\right] \tag{21}$$

The denominator $\alpha\boldsymbol{V} + \boldsymbol{U}^T\boldsymbol{\Phi}_{p-1}\boldsymbol{U}$ is a matrix with dimension 2×2; its inverse computation is simple, and can be implemented for real-time applications. $\boldsymbol{\theta}_0 = \boldsymbol{0}$. $\boldsymbol{\Phi}_p$ is a covariance matrix with initial condition $\boldsymbol{\Phi}_0 = \rho\boldsymbol{I}$, where ρ is a positive quantity and $\boldsymbol{I}$ is an identity matrix.
By simulation tests with the requirements of the recognition rate ≥ 80%, reasonable training speed and accuracy, the following initial values are given to the related parameters in this study: $\eta = 0.01$ with tested range of $\eta \in [0.001, 10]$; $\alpha = 0.995$ with tested range of $\alpha \in [0.95, 1]$; $\rho = 10^3$ with tested range of $\rho \in [10^2, 10^5]$.

4.2 Implementation of the hybrid training method

In implementation, inside each training epoch, the nonlinear MF parameters in the classifier are optimized in the backward pass by using a recursive LM method, whereas consequent linear rule weights are updated by LSE in the forward pass. On the other hand, after training or real applications over some time period, if the updated rule weights w_j are sufficiently small (e.g., w_j < 0.01), the contribution of the related rule to the final classification operation can be neglected, and that rule can be removed from the rule base.

5. Performance evaluation

5.1. Experimental setup

Fig. 7 shows the experimental setup used in this study to verify the performance of the proposed integrated classifier.
The apparatus is anchored onto a massive concrete block. It consists of a 3-HP AC drive motor and a gearbox. The motor rotation is controlled by a speed controller which allows tested gears operating in the range of 20 to 4200 rpm. An optical sensor provides a one-pulse-per-revolution signal which is used as the reference for the time synchronous average

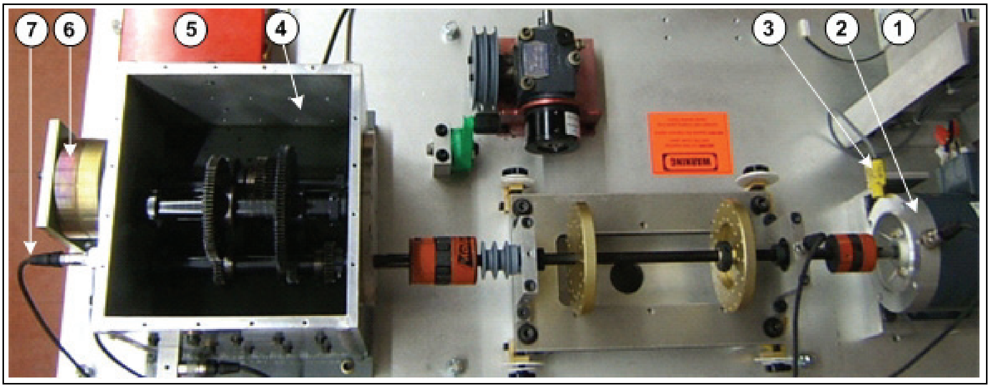


Fig. 7. The experimental setup: 1-speed controller, 2-motor, 3-optical sensor, 4-gearbox, 5-load controller, 6-loading system, 7-sensors.

filtering. The gearbox consists of two pairs of spur or helical gears. The shafts in the gearbox are mounted to the housing by rolling element bearings. The load is provided by a magnetic loading system which is connected to the output shaft. The speed of the drive motor and the load are adjusted to simulate different speed/load operating conditions. The vibration is measured using ICP accelerometers mounted on the gearbox housing along different orientations. After being properly preconditioned, the collected signals are fed to a computer for further processing.

5.2 Performance evaluation

To verify the viability of the proposed classifier, five gear cases are tested in this study as represented in Fig. 8:

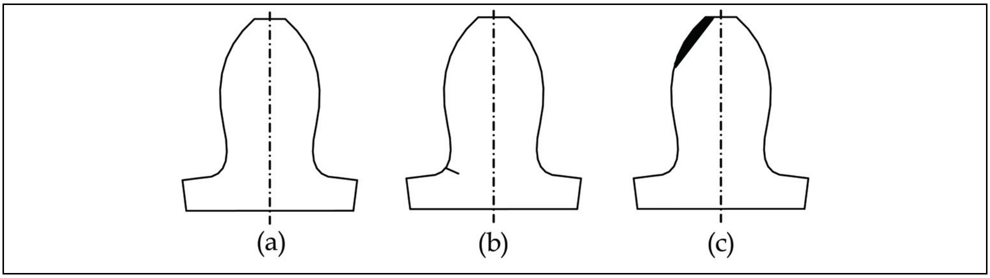


Fig. 8. Gear conditions tested: (a) healthy gear, (b) cracked gear; (c) chipped gear.

a. healthy gears (C_1);
b. gears having a tooth crack with 15% (C_2) and 50% (C_3) tooth root thickness;
c. gears having a chipped tooth with 10% (C_2) and 40% (C_3) tooth surface area removed.

These demonstrated faults belong to localized gear defects. From the signal property standpoint, when a localized fault occurs, some high-amplitude pulses will be generated due to impacts, which are relatively easier for a signal processing technique to recognize. When a localized fault propagates towards a distributed defect, the overall energy of the fault will increase, but it often becomes more wideband in nature and difficult to detect in the presence of the other vibratory components of the machine. This example identifies a characteristic of currently used fault detection techniques: It is usually easier to detect a distinct low-level narrowband tone than a high-level wideband signal in the presence of other signals or noises. Even though a distributed defect, such as pitting and wear, is initiated from a localized fault which is detectable as an incipient defect, most currently available vibration-based signal processing techniques cannot effectively detect an advanced distributed fault which, however, can be diagnosed based on other information carriers, such as acoustic signals.

To make a comparison, the diagnostic results from the following three classifiers are also listed:

1. A pure fuzzy system with a similar reasoning architecture as in Fig. 2 but without the use of predictors. The rule weight factors are chosen as those in the integrated classifier after initial training.
2. Classifier-1: An NF classifier with a similar reasoning architecture as in Fig. 2 but without predictors. Its MF parameters are trained by a gradient-LSE algorithm.
3. Classifier-2: Same as Classifier-1, but trained by the hybrid algorithm of the recursive LM and LSE.

Given the network architectures, the initial parameters of three adaptive classifiers can be primarily trained by using some data sets collected in previous tests on the same test apparatus, or be initialized by experience. Then these classifier parameters are optimized in the following online training processes.

During online tests, motor speed and load levels are randomly changed to simulate general and unusual machinery operating conditions. The tests are conducted under load levels from 0.5 to 3 hp, and motor speeds from 50 to 3600 rpm.

In online monitoring, based on test schedule and load/speed change frequency, the monitoring time-interval is set at 15 minutes; that is, all the monitoring schemes are applied automatically every 15 minutes for condition monitoring operations. Three-steps-ahead predictors (i.e., $r = 3$) are used in the integrated classifier. The selection of data size depends on noise reduction requirement; usually the data for the gear with the lowest speed should cover more than 100 revolutions. For example, if the slowest gear speed in the gearbox is 1200 rpm, the data acquisition process takes at least 5 seconds (15 seconds in this case). The monitoring is performed gear by gear. Three examples corresponding to healthy, cracked and chipped gears (all having 41 teeth) have been illustrated in Figs. 3 to 5, respectively.

Each healthy gear condition is tested over 24 hours whereas each faulty gear condition is tested over 50 hours. In total, 386 data pairs are recorded for testing purpose. Table 1 summarizes the classification performance by different diagnostic schemes.

Diagnostic Schemes	Healthy Gears		Cracked Gears		Chipped Gears		Overall Accuracy
	M.A.	F.A.	M.A.	F.A.	M.A.	F.A.	
Fuzzy System	0	13	12	16	3	8	85.3 %
Classifier-1	0	7	6	9	1	5	92.5 %
Classifier-2	0	5	7	8	0	4	93.6 %
New Classifier	0	2	3	3	0	1	97.6 %

Table 1. Comparison of the diagnostic results from different diagnostic schemes. M.A.- Missed Alarms, F.A.- False Alarms.

The fuzzy classifier records 15 missed alarms and 37 false alarms, with an overall reliability of 85.3%. Its relatively poor diagnostic performance is mainly due to the lack of learning capability. In addition, fixed or human-determined system parameters are subject to variations and are rarely optimal in terms of reproducing the desired classification outputs, which results in the fuzzy classifier not being optimized under different operating conditions.

Classifier-1 records 7 missed alarms and 21 false alarms, with an overall reliability of 92.5%. One difference between this NF system and the fuzzy classifier is related to the rule weight factors. Each signal processing technique (and the resulting feature) has a limited capability in fault detection. Even if the firing strengths of two fuzzy if-then rules are identical, their diagnostic reliabilities may be different under different machinery conditions. Therefore, rule weights play an important role in the diagnostic classification operations.

Classifier-2 records 7 missed alarms and 17 false alarms, with an overall reliability of 93.6%. The main difference between Classifier-2 and Classifier-1 is related to training algorithms. It is seen that the recursive LM algorithm is superior to the gradient method in convergence, and has the randomness to reduce the chance of possible trapping due to local minima. In addition, each rule has its own decision (mapping) space, whereas the MFs and the rule weights are directly associated with the characteristics of the decision space. The efficient optimization of classifiers can adjust the boundary characteristics of the decision space so as to reduce misclassifications. This property is especially important for classifier with coarse fuzzy partitions.

The developed integrated classifier generates 3 missed alarms and 7 false alarms, with an overall reliability of 97.6%. Compared with Classifier-2, the integrated classifier can enhance the classification accuracy by properly implementing the future states of the classifier. It follows that adaptively fine-tuning the fuzzy parameters is necessary to enhance the approximation of the mapping from the observed symptoms to the underlying faults. In addition, the fault severity can be recognized because, to some extent, the greater the fault, the more pronounced the feature modulation, and the larger the monitoring indices will become.

The developed integrated diagnostic classifier provides a robust problem solving framework. Machinery conditions vary dramatically in real-world applications, and new system conditions may occur under different circumstances. With the help of an adequate learning algorithm, new information can be extracted from online training, and the diagnostic knowledge base can be expanded automatically to accommodate different machinery conditions.

In general, deterioration history of most machinery components follows a "*U curve*" as illustrated in Fig. 9. It consists of four periods: the run-in stage (I), the normal operation period (II), initial (III) and advanced (IV) failure stages, respectively. Such a trend characteristic is easy for a powerful NF predictor to catch up. If a false alarm is generated during the healthy period II, the false alarm is induced due to noise instead of real defect. Based on the forecast result, the diagnostic state should lie in period III (or initial defect). However random noise will disappear in the following processing steps, and the diagnostic indicator should return to period II (or healthy). Correspondingly, this misclassification can be prevented by the integrated classification /forecasting information. On the other hand, if an object is damaged, its diagnostic indicator should lie in period III (or IV). If a misclassification occurs, or the diagnostic indicator falls in period II, the forecast information will be contradictory to that from the classifier. Comprehensive analysis in Eq. (7) can avoid this possible missed alarm so as to improve fault diagnostic reliability. In both aforementioned examples, classifier will be updated to accommodate such a noise in the following monitoring applications.

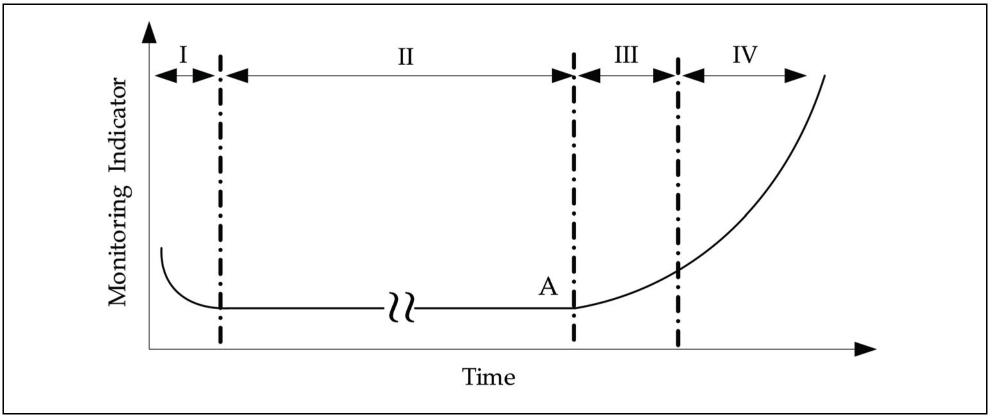


Fig. 9. The deterioration trend of a machinery component.

5. Conclusions

In this paper, an integrated classifier is developed for gear fault diagnostics. The purpose is to provide industries with a more reliable monitoring tool to prevent machinery system performance degradation, malfunction, and sudden failure. The classifier can integrate different features for a more positive assessment of the object's health condition. The diagnostic reliability is improved by properly integrating the future states of the gear, which are forecast by multi-step predictors. An online hybrid training technique based on a recursive LM and LSE is adopted to improve the classifier's convergence and adaptive capability to accommodate different machinery conditions. The viability of the new integrated classifier has been verified by experimental tests corresponding to different gear conditions.

On the other hand, it should be stated that although satisfactory results have been achieved based on the developed integrated classifier, its network architecture is relatively complex which may not be easy for implementation for some real-world applications. Future research is to develop novel evolving fuzzy or neuro-fuzzy classification schemes for more effective diagnostic operations. New training algorithms will be proposed to further improve the training convergence. The proposed techniques will also be employed for real-world industrial applications in vehicles, wind turbines, and manufacturing facilities.

6. Acknowledgement

This work was partly supported by MC Technologies Inc. and Materials and Manufacturing Ontario in Canada.

7. Appendix A: monitoring indices

1. Wavelet energy reference function $R_w(t)$

If $u_r(\tau)$ is the overall residual of the signal average $u(\tau)$, the wavelet energy function is proposed as

$$R_w(t) = \int_{f_1}^{f_2} \left| \int_0^{+\infty} u_r(\tau)\sqrt{s}\, w(t-\tau)\, d\tau \right| ds \tag{A1}$$

where s and t are the scale (frequency) and time variables, respectively; f_1 and f_2 are the frequency limits of interest. For the gear system in this work, $f_1 = 0.5 f_R N$ and $f_2 = 4.5 f_R N$; $w(t)$ is the mother wavelet, which is a modified Morlet function:

$$w(t) = \exp\left(-\frac{9\ln 2}{\pi^2} f_R^2 t^2 \right) \exp(2\pi k f_R t) \tag{A2}$$

where $k = 1, 2, \ldots, 5N$.

2. Beta kurtosis reference function $R_b(t)$

The beta kurtosis is the normalized fourth moment of a signal, in terms of the beta function instead of a generally used Gaussian function. If m_t and σ_t^2 represent the mean and variance of a tooth data block, T_d, centered at t, then $R_b(t)$ is defined as the reciprocal of the kurtosis

$$R_b(t) = \frac{\alpha\beta(\alpha+\beta+2)(\alpha+\beta+3)}{3(\alpha+\beta+1)(2\alpha^2 - 2\alpha\beta + \alpha^2\beta + \alpha\beta^2 + 2\beta^2)} \tag{A3}$$

where $\alpha = \frac{m_t}{\sigma_t^2}(m_t - m_t^2 - \sigma_t^2)$ and $\beta = \frac{1-m_t}{\sigma_t^2}(m_t - m_t^2 - \sigma_t^2)$. The derivation of (A3) can be found in [13].

3. Phase modulation reference function $R_p(t)$

For a pair of healthy gears with sound installation and ideal operating conditions, the meshing vibration $u(t)$ can be approximately expressed as,

$$u(t) = \sum_{h=0}^{H} A_h \cos(2\pi h N f_R t + \theta_h) \tag{A4}$$

where H is the total number of mesh frequency harmonics considered. If a fault occurs in one tooth, because of a change in tooth stiffness, the amplitude and phase functions of the gear meshing vibration will be modulated:

$$u(t) = \sum_{h=0}^{H} (A_h + a_h)\cos[2\pi h N f_R t + \theta_h + \phi(t)] \tag{A5}$$

The phase modulation $\phi(t)$ can be obtained from the analytical signal of (A5), which is computed by taking the Hilbert transform to $u(t)$. The phase reference function $R_p(t)$ is derived as the maximum phase difference over a tooth period T_d centered at t,

$$R_p(t) = \phi_{\max}(\tau) - \phi_{\min}(\tau),\ \tau \in [\, t - 0.5\,T_d,\ t + 0.5\,T_d\,]. \tag{A6}$$

8. References

Chelidze, D. & Cusumano, J. (2004). A dynamical systems approach to failure prognosis, *Journal of Vibration and Acoustics*, Vol. 126, 1-7.

Castellano G; Fanelli, A. & Mencar, C. (2004). An empirical risk functional to improve learning in a neuro-fuzzy classifier, *IEEE Transactions on Systems, Man, Cybernetics, Part B*, Vol. 34, 725-731.

Figueiredo, M.; Ballini, R.; Soares, S.; Andrade, M. & Gomide F. (2004). Learning algorithms for a class of neurofuzzy network and applications, *IEEE Transactions on Systems, Man, Cybernetics, Part C*, Vol. 34, 293-301.

Gusumano, J.; Chelidze, D. & Chatterjee, A. (2002). Dynamical systems approach to damage evolution tracking, part 2: Model-based validation and physical interpretation, *Journal of Vibration and Acoustics* , Vol. 124, 258-264.

Ishibuchi, H. & Yamamoto, Y. (2005). Rule weight specification in fuzzy rule-based classification systems, *IEEE Transactions on Fuzzy Systems*, Vol. 13, 428-435.

Isermann, R. (1998). On fuzzy logic applications for automatic control, supervision, and fault diagnosis, *IEEE Transactions on Systems, Man, Cybernetics, Part A*, Vol. 28, 221-235.

Jang, J. (1993). ANFIS: adaptive-network-based fuzzy inference system, *IEEE Transactions on Systems, Man, Cybernetics*, Vol. 23, 665-685.

Korbicz, J.; Koscielny, J.; Kowalczuk, Z. & Cholewa, W. (2004). *Fault Diagnosis: Models, Artificial Intelligence, Applications*, Springer.

Li, L. & Lee, H. (2005). Gear fatigue crack prognosis using embedded model gear dynamic model and fracture mechanics, *Mechanical Systems and Signal Processing*, Vol. 20, 836-846.

Mansoori, E.; Zolghadri, M. & Katebi, S. (2007). A weighting function for improving fuzzy classification systems performance, *Fuzzy Sets and Systems*, Vol. 158, 583-591.

McFadden, P. (1986). Detecting fatigue cracks in gears by amplitude and phase demodulation of the meshing vibration, *Journal of Vibration, Acoustics, Stress, and Reliability in Design*, Vol. 108, 165-170.

Pourahmadi, M. (2001). *Foundation of Time Series Analysis and Prediction Theory*, John & Sons.

Rish, I; Brodie, M.; Ma, S.; Odintsova, N.; Beygelzimer, A.; Grabarnik, G. & Hernandez, K. (2005). Adaptive diagnosis in distributed systems, *IEEE Transactions on Neural Networks*, Vol. 16, 1088-1109.

Tse, P. & Atherton, D. (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks, *Journal of Vibration and Acoustics*, Vol. 121, 355-362.

Uluyol, O.; Kim, K. & Nwadiognu, E. (2006). Synergistic use of soft computing technologies for fault detection in gas turbine engines, *IEEE Transactions on Systems, Man, Cybernetics, Part C*, Vol. 36, 476-484.

Wang, J & Lee, L. (2002). Self-adaptive neuro-fuzzy inference systems for classification applications, *IEEE Transactions on Fuzzy Systems*, Vol. 10, 790-802.

Wang, W. (2008). An intelligent system for machinery condition monitoring, *IEEE Transactions on Fuzzy Systems*, Vol. 16, No. 1, 110-122.

Wang, W.; Ismail, F. & Golnaraghi, F. (2001). Assessment of gear damage monitoring techniques using vibration measurements, *Mechanical Systems and Signal Processing*, Vol. 15, 905-922.

Wang, W. & Vrbanek, J. (2007). A multi-step predictor for dynamic system property forecasting, *Measurement Science and Technology*, Vol. 18, 3673-3681.

7

Fuzzy Filtering: A Mathematical Theory and Applications in Life Science

Mohit Kumar, Kerstin Thurow, Norbert Stoll, and Regina Stoll
University of Rostock
Germany

1. Introduction

A life science process is typically characterized by a large number of variables whose interrelations are uncertain and not completely known. The development of a computational paradigm, implementing an "intelligent" behavior in the sense of handling uncertainties related to the modeling of the interrelations among process variables, is an interesting research topic. A large number of studies apply computational intelligence techniques in the life science e.g.

- in modeling the environmental behavior of chemicals (Eldred & Jurs, 1999; Kaiser & Niculescu, 1999; Gini et al., 1999; L. Sztandera et al., 2003; Sztandera et al., 2003; Vracko, 1997; Benfenati & Gini, 1997; Gini, 2000; Mazzatorta et al., 2003),
- in medicine (Wilson & Russell, 2003b; Fukuda et al., 2001; Wilson & Russell, 2003a; Mandryk & Atkins, 2007; Lin et al., 2006; Rani et al., 2002; Adlassnig, 1986; Adlassnig et al., 1985; Bellazzi et al., 2001; 1998; Belmonte et al., 1994; Binaghi et al., 1993; Brai et al., 1994; Daniels et al., 1997; Fathitorbaghan & Meyer, 1994; Garibaldi & Ifeachor, 1999; Kuncheva & Steimann, 1999; Roy & Biswas, 1992; Steimann, 1996;Watanabe et al., 1994; Wong et al., 1990),
- in chemistry and drug design, see e.g. (Manallack & Livingstone, 1999; Winkler, 2004; Duch et al., 2007) and references therein.

The fuzzy systems based on fuzzy set theory (Zadeh, 1973; 1983) are considered suitable tools for dealing with the uncertainties. The use of fuzzy systems in data driven modeling is a topic that is widely studied by the researchers (Wang & Mendel, 1992; Nozaki et al., 1997; Shan & Fu, 1995; Nauck & Kruse, 1998; Jang, 1993; Thrift, 1991; Liska & Melsheimer, 1994; Herrera et al., 1994; González & Pérez, 1998; Babuška & Verbruggen, 1997; Babuška, 1998; Abonyi et al., 2002; Simon, 2000; 2002; Jang et al., 1997; Wang & Vrbanek, 2008; Lughofer, 2008; Kumar, Stoll & Stoll, 2009b; Lin et al., 2008; Kumar, Stoll & Stoll, 2009a) due to the successful applications of fuzzy techniques in data mining, prediction, control, classification, simulation, and pattern recognition.

It is assumed that input variables ($x_1, x_2, \dots, x_n$) are related to the output variable y through a mapping:

$$y = f(x)$$

where $x = [x_1\ x_2\ \ldots\ x_n] \in R^n$ is the input vector and the modeling aim is to identify the unknown function f . The fuzzy modeling is based on the assumption that there exists an ideal set of model parameters w^* such that model output $M(x;w^*)$ to input x is an approximation of the output value y. However, it may not be possible, for a given type and structure of the model M, to identify perfectly the inputs-output relationships. The part of the input-output mappings that can't be modeled, for a given type and structure of the model, is what we refer to as the uncertainty. Mathematically, we have

$$y = M(x;w^*) + n, \tag{1}$$

where n is termed as disturbance or noise in system identification literature. However, we refer n, in context to real-world modeling applications, to as uncertainty to emphasize that the uncertainties regarding optimal choices of the model and errors in output data resulted in the additive disturbance in (1). For an illustration, the authors in (Kumar et al., 2008), in context to subjective workload score modeling, explain the reasons giving rise to the uncertainty.

A robust (towards uncertainty n) identification of model parameters w^* using available inputs-output data pairs $\{\ x(j),y(j)\ \}_{j=0,1,\ldots}$ is obviously a straightforward approach to handle the uncertainty. Several robust methods of fuzzy identification have been developed (Chen & Jain, 1994; Wang et al., 1997; Burger et al., 2002; Yu & Li, 2004; Johansen, 1996; Hong et al., 2004; Kim et al., 2006; Kumar et al., 2004b; 2003b; 2006c; 2004a; 2006a;b). It may be desired to estimate the parameters w^* in an on-line scenario using an adaptive filtering algorithm aiming at the filtering of uncertainty n from y. A classical application of adaptive filters is to remove noise and artifacts from the biomedical signals (Philips, 1996; Lee & Lee, 2005; Plataniotis et al., 1999; Mastorocostas et al., 2000; Li et al., 2008). The adaptive filtering algorithms applications are not only limited to the engineering problems but also e.g. to medicinal chemistry where it is required to predict the biological activity of a chemical compound before its synthesis in the lab (Kumar et al., 2007b). Once a compound is synthesized and tested experimentally for its activity, the experimental data can be used for an improvement of the prediction performance (i.e. online learning of the adaptive system). Adaptive filtering of uncertainties may be desired e.g. for an intelligent interpretation of medical data which are contaminated by the uncertainties arising from the individual variations due to a difference in age, gender and body conditions (Kumar et al., 2007).

2. The fuzzy filter

It is required to filter out the uncertainties from the data with applications to many real-world modeling problems (Kumar et al., 2007; Kumar et al., 2007; Kumar et al., 2007a;b; 2008; Kumar et al., 2009; Kumar et al., 2008). A filter, in the context of our study, simply maps an input vector x to the quantity $y - n$ (called filtered output $y_f = y - n$) and thus separates uncertainty n from the output value y.

2.1 A Takagi-Sugeno fuzzy filter

Consider a zero-order Takagi-Sugeno fuzzy model ($F_s : X \to Y$) that maps n–dimensional input space ($X = X_1 \times X_2 \times \ldots \times X_n$) to one dimensional real line. A rule of the model is represented as

$$If\ x_1\ is\ A_1\ and \cdots and\ x_n\ is\ A_n\ then\ y_f = s.$$

Here $(x_1, \ldots, x_n)$ are the model input variables, y_f is the filtered output variable, $(A_1, \ldots, A_n)$ are the linguistic terms which are represented by fuzzy sets, and s is a real scalar. Given a universe of discourse X_j, a fuzzy subset A_j of X_j is characterized by a mapping:

$$\mu_{A_j} : X_j \to [0,1],$$

where for $x_j \in X_j$, $\mu_{A_j}(x_j)$ can be interpreted as the degree or grade to which x_j belongs to A_j. This mapping is called as membership function of the fuzzy set. Let us define, for j^{th} input, P_j non-empty fuzzy subsets of X_j (represented by $A_{1j}, A_{2j}, \ldots, A_{P_j j}$). Let the i^{th} rule of the rule-base is represented as

$$R_i \quad : \quad If\ x_1\ is\ A_{i1}\ and \cdots and\ x_n\ is\ A_{in}\ then\ y_f = s_i,$$

where $A_{i1} \in \{A_{11}, \cdots, A_{P_1 1}\}, A_{i2} \in \{A_{12}, \cdots, A_{P_2 2}\}$ and so on. Now, the different choices of $A_{i1}, A_{i2}, \ldots, A_{in}$ leads to the $K = \prod_{j=1}^{n} P_j$ number of fuzzy rules. For a given input x, the *degree of fulfillment* of the i^{th} rule, by modelling the logic operator 'and' using product, is given by

$$g_i(x) = \prod_{j=1}^{n} \mu_{A_{ij}}(x_j).$$

The output of the fuzzy model to input vector $x \in X$ is computed by taking the weighted average of the output provided by each rule:

$$y_f = \frac{\sum_{i=1}^{K} s_i g_i(x)}{\sum_{i=1}^{K} g_i(x)} = \frac{\sum_{i=1}^{K} s_i \prod_{j=1}^{n} \mu_{A_{ij}}(x_j)}{\sum_{i=1}^{K} \prod_{j=1}^{n} \mu_{A_{ij}}(x_j)}. \tag{2}$$

Let us define a real vector θ such that the membership functions of any type (e.g. trapezoidal, triangular, etc) can be constructed from the elements of vector θ. To illustrate the construction of membership functions based on knot vector (θ), consider the following examples:

2.1.1 Trapezoidal membership functions:

Let

$$\theta = (a_1, t_1^1, \cdots, t_1^{2P_1-2}, b_1, \cdots, a_n, t_n^1, \cdots, t_n^{2P_n-2}, b_n)$$

such that for i^{th} input ($x_i \in [a_i, b_i]$), $a_i < t_i^1 < \cdots < t_i^{2P_i-2} < b_i$ holds $\forall\ i = 1, \ldots, n$. Now, P_i trapezoidal membership functions for i^{th} input ($\mu_{A_{1i}}, \mu_{A_{2i}}, \cdots, \mu_{A_{P_i i}}$) can be defined as:

$$\mu_{A_{1i}}(x_i,\theta) = \begin{cases} 1 & \text{if } x_i \in [a_i\ ,\ t_i^1] \\ \dfrac{-x_i + t_i^2}{t_i^2 - t_i^1} & \text{if } x_i \in [t_i^1\ ,\ t_i^2] \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{A_{ij}}(x_i,\theta) = \begin{cases} \dfrac{x_i - t_i^{2j-3}}{t_i^{2j-2} - t_i^{2j-3}} & \text{if } x_i \in [t_i^{2j-3}\ ,\ t_i^{2j-2}] \\ 1 & \text{if } x_i \in [t_i^{2j-2}\ ,\ t_i^{2j-1}] \\ \dfrac{-x_i + t_i^{2j}}{t_i^{2j} - t_i^{2j-1}} & \text{if } x_i \in [t_i^{2j-1}\ ,\ t_i^{2j}] \\ 0 & \text{otherwise} \end{cases}$$

$$\vdots$$

$$\mu_{A_{P_i i}}(x_i,\theta) = \begin{cases} \dfrac{x_i - t_i^{2P_i-3}}{t_i^{2P_i-2} - t_i^{2P_i-3}} & \text{if } x_i \in [t_i^{2P_i-3}, t_i^{2P_i-2}] \\ 1 & \text{if } x_i \in [t_i^{2P_i-2}, b_i] \\ 0 & \text{otherwise} \end{cases}$$

2.1.2 One-dimensional clustering criterion based membership functions:

Let

$$\theta = (a_1, t_1^1, \cdots, t_1^{P_1-2}, b_1, \cdots, a_n, t_n^1, \cdots, t_n^{P_n-2}, b_n)$$

such that for i^{th} input, $a_i < t_i^1 < \cdots < t_i^{P_i-2} < b_i$ holds for all i = 1, …,n. Now, consider the problem of assigning two different memberships (say $\mu_{A_{1i}}$ and $\mu_{A_{2i}}$) to a point x_i such that $a_i < x_i < t_i^1$, based on following clustering criterion:

$$[\mu_{A_{1i}}(x_i), \mu_{A_{2i}}(x_i)] = \arg \min_{[u_1,u_2]} \left[u_1^2 (x_i - a_i)^2 + u_2^2 (x_i - t_i^1)^2, u_1 + u_2 = 1 \right].$$

This results in

$$\mu_{A_{1i}}(x_i) = \frac{(x_i - t_i^1)^2}{(x_i - a_i)^2 + (x_i - t_i^1)^2}, \mu_{A_{2i}}(x_i) = \frac{(x_i - a_i)^2}{(x_i - a_i)^2 + (x_i - t_i^1)^2}.$$

Thus, for i^{th} input, P_i membership functions can be defined as:

$$\mu_{A_{1i}}(x_i,\theta) = \begin{cases} 1 & x_i \le a_i \\ \dfrac{(x_i - t_i^1)^2}{(x_i - a_i)^2 + (x_i - t_i^1)^2} & a_i \le x_i \le t_i^1 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{A_{2i}}(x_i,\theta)=\begin{cases}\dfrac{(x_i-a_i)^2}{(x_i-a_i)^2+(x_i-t_i^1)^2} & a_i\le x_i\le t_i^1\\ \dfrac{(x_j-t_i^2)^2}{(x_i-t_i^1)^2+(x_j-t_i^2)^2} & t_i^1\le x_i\le t_i^2\\ 0 & \text{otherwise}\end{cases}$$

$$\vdots$$

$$\mu_{A_{P_i i}}(x_i,\theta)=\begin{cases}1 & x_i\ge b_i\\ \dfrac{(x_i-t_i^{P_i-2})^2}{(x_i-t_i^{P_i-2})^2+(x_i-b_i)^2} & t_i^{P_i-2}\le x_i\le b_i\\ 0 & \text{otherwise}\end{cases}$$

For any choice of membership functions (which can be constructed from a vector θ), (2) can be rewritten as function of θ:

$$y_f=\sum_{i=1}^{K}s_iG_i(x_1,x_2,\ldots,x_n,\theta),\; G_i(x_1,x_2,\ldots,x_n,\theta)=\frac{\prod_{j=1}^{n}\mu_{A_{ij}}(x_j,\theta)}{\sum_{i=1}^{K}\prod_{j=1}^{n}\mu_{A_{ij}}(x_j,\theta)}.$$

Let us introduce the following notation: $\alpha=[s_1\cdots s_K]\in R^K$, $x=[x_1\cdots x_n]\in R^n$, $G(x,\theta)=[G_1(x,\theta)\cdots G_K(x,\theta)]\in R^K$. Now, (2) becomes

$$y_f=G^T(x,\theta)\alpha.$$

In this expression, θ is not allowed to be any arbitrary vector, since the elements of θ must ensure

1. in case of trapezoidal membership functions,

$$a_i<t_i^1<\cdots<t_i^{2P_i-2}<b_i,\ \forall i=1,\cdots,n, \tag{3}$$

2. in case of one-dimensional clustering criterion based membership functions

$$a_i<t_i^1<\cdots<t_i^{P_i-2}<b_i,\ \forall i=1,\cdots,n, \tag{4}$$

to preserve the linguistic interpretation of fuzzy rule base (Lindskog, 1997). In other words, there must exists some $\epsilon_i>0$ for all $i = 1, \ldots, n$ such that for trapezoidal membership functions,

$$\begin{aligned} t_i^1-a_i &\ge \epsilon_i,\\ t_i^{j+1}-t_i^j &\ge \epsilon_i, \quad \text{for all } j=1,2,\ldots,(2P_i-3)\\ b_i-t_i^{2P_i-2} &\ge \epsilon_i.\end{aligned}$$

These inequalities and any other membership functions related constraints (designed for incorporating a priori knowledge) can be written in the form of a matrix inequality $c\theta\ge h$

(Burger et al., 2002; Kumar et al., 2003b;a; 2004b;a; 2006c;a). Hence, a Sugeno type fuzzy filter can be represented as

$$y_f = G^T(x,\theta)\alpha,\ c\theta \geq h. \tag{5}$$

2.2 A clustering based fuzzy filter

The fuzzy filter of (Kumar et al., 2007; Kumar et al., 2007; Kumar et al., 2007a;b; 2008; Kumar et al., 2009; Kumar et al., 2008) has K number of fuzzy rules of following type:

$$\textit{If } x \textit{ belongs to a cluster having centre } c_1 \textit{ then } y_f = s_1$$
$$\vdots$$
$$\textit{If } x \textit{ belongs to a cluster having centre } c_K \textit{ then } y_f = s_K$$

where $c_i \in R^n$ is the centre of i^{th} cluster, and the values $s_1, \ldots, s_K$ are real numbers. Based on a clustering criterion, it was shown in e.g. (Kumar et al., 2008) that

$$y_f = \sum_{i=1}^{K} s_i G_i(x, c_1, \cdots, c_K),$$

$$G_i(x,c_1,\cdots,c_K) = \frac{A_i(x,c_1,\cdots,c_K)}{\sum_{i=1}^{K} A_i(x,c_1,\cdots,c_K)},\ A_i(x,c_1,\cdots,c_K) = \frac{A_{1i}^{\tilde{m}}}{2} + \frac{A_{2i}}{2},\ \tilde{m} > 1,$$

where A_{1i}, A_{2i} are given as

$$A_{1i} = \begin{cases} \dfrac{1}{\sum_{j=1}^{K} \left(\dfrac{\| x - c_i \|^2}{\| x - c_j \|^2} \right)^{\frac{1}{\tilde{m}-1}}} & x \in X \setminus \{c_j\}_{j=1,\cdots,K}, \\ 1 & x = c_i, \\ 0 & x \in \{c_j\}_{j=1,\cdots,K} \setminus \{c_i\} \end{cases}$$

$$A_{2i} = \exp\left(-\frac{\| x - c_i \|^2}{\delta_i}\right),\ \delta_i = \min_{j,\, j \neq i} \| c_j - c_i \|^2 .$$

With the notations:

$$\alpha = [s_1 \cdots s_K] \in R^K,\ \theta = [c_1^T \cdots c_K^T]^T \in R^{Kn},\ G(x,\theta) = [G_1(x,\theta) \cdots G_K(x,\theta)] \in R^K,$$

the output of fuzzy filter for an input x can be expressed as

$$y_f = G^T(x,\theta)\alpha. \tag{6}$$

3. Estimation of fuzzy model parameters

The fuzzy filter parameters (α, θ) need to be estimated using given inputs-output data pairs $\{x(j),y(j)\}_{j=0,1,\ldots,N}$. This section outlines some of our results on the topic.

Result 1 (The result of (Kumar et al., 2009b)) *A class of algorithms for estimating the parameters of Takagi-Sugeno type fuzzy filter recursively using input-output data pairs* $\{x(j),y(j)\}_{j=0,1,\ldots}$ *is given by the following recursions:*

$$\theta_j = \arg\min_{\theta} \left[\Psi_j(\theta), c\theta \geq h \right] \tag{7}$$

$$\alpha_j = \alpha_{j-1} + \frac{P_j G(x(j),\theta_j)[y(j) - G^T(x(j),\theta_j)\alpha_{j-1}]}{1 + G^T(x(j),\theta_j)P_j G(x(j),\theta_j)}, \tag{8}$$

$$\Psi_j(\theta) = \frac{|y(j) - G^T(x(j),\theta)\alpha_{j-1}|^2}{1 + G^T(x(j),\theta)P_j G(x(j),\theta)} + \mu_\theta^{-1} \| \theta - \theta_{j-1} \|^2 \tag{9}$$

for all $j = 0, 1, \ldots$ *with* $\alpha_{-1} = 0$, $P_0 = \mu I$, *and* θ_{-1} *is an initial guess about antecedents. The positive constants* (μ,μ_θ) *are the learning rates for* (α, θ) *respectively. Here,* $\gamma \geq -1$ *is a scalar whose different choices solve the following different filtering problems:*

- $\gamma = -1$ *solves a* H^∞*-optimal like filtering problem,*
- $-1 \leq \gamma < 0$ *solves a risk-averse like filtering problem,*
- $\gamma > 0$ *solves a risk-seeking like filtering problem.*

The positive constants μ_θ in (9) is the learning rate for θ. The elements of vector θ, if assumed as random variables, may have different variances depending upon the distribution functions of different inputs. Therefore, estimating the elements of $\theta \in R^L$ with different learning rates makes a sense. To do this, define a diagonal matrix Σ (with positive entries on its main diagonal):

$$\Sigma = \begin{bmatrix} \mu_{\theta(1)} & 0 & \cdots & 0 \\ 0 & \mu_{\theta(2)} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & \cdots & \mu_{\theta(L)} \end{bmatrix},$$

to reformulate (9) as

$$\Psi_j(\theta) = \frac{|y(j) - G^T(x(j),\theta)\alpha_{j-1}|^2}{1 + G^T(x(j),\theta)P_j G(x(j),\theta)} + \| \Sigma^{-1/2}(\theta - \theta_{j-1}) \|^2 . \tag{10}$$

Result 2 (The result of (Kumar, Stoll & Stoll, 2009a)) *The adaptive p–norm algorithms for estimating the parameters of Takagi-Sugeno type fuzzy filter recursively using input-output data pairs* $\{x(j),y(j)\}_{j=0,1,\ldots}$ *take a general form of*

$$\theta_j = \arg\min_{\theta} [E_j(\widehat{\alpha}(\theta),\theta) + \mu_{\theta,j}^{-1} d_q(\theta,\theta_{j-1}); c\theta \geq h] \tag{11}$$

$$\alpha_j = f^{-1}\Big(f(\alpha_{j-1}) + \mu_j \phi\big(y(j) - G^T(x(j),\theta_j)\alpha_{j-1} \big) G(x(j),\theta_j) \Big) \tag{12}$$

Here,

$$E_j(\alpha,\theta) = L_j(\alpha,\theta) + \mu_j^{-1} d_q(\alpha,\alpha_{j-1}),$$

$$\hat{\alpha}(\theta) = f^{-1}\left(f(\alpha_{j-1}) + \mu_j\phi\left(y(j) - G^T(x(j),\theta)\alpha_{j-1}\right)G(x(j),\theta)\right),$$

$$d_q(u,w) = \frac{1}{2}\| u \|_q^2 - \frac{1}{2}\| w \|_q^2 - (u-w)^T f(w),$$

where ($\mu_j,\mu_{\theta,j}$) *are the learning rates for* (α, θ) *respectively,* f (*a* p *indexing for* f *is understood*), *as defined in* (*Gentile, 2003*), *is the bijective mapping* $f: \mathrm{R}^K \rightarrow R^K$ *such that*

$$f = [f_1 \cdots f_K]^T,\ f_i(w) = \frac{sign(w_i)\,|\,w_i\,|^{q-1}}{\| w \|_q^{q-2}},$$

where $w = [w_1 \cdots w_K]^T \in R^K$, *q is dual to p* (*i.e.* $1/p + 1/q = 1$), *and* $\|\cdot\|_q$ *denotes the q-norm.*

The different choices of loss term $L_j(\alpha, \theta)$ *lead to the different functional form of* ϕ *and thus different types of fuzzy filtering algorithms for any* p ($2 \le p \le \infty$). *A few examples of fuzzy filtering algorithms are listed in the following:*

- *algorithm* $A_{1,p}$:

$$L_j(\alpha,\theta) = \ln(\cosh(y(j) - G^T(x(j),\theta)\alpha))$$

$$\phi(e) = \tanh(e)$$

$$P_\phi(y,\overline{y}) = \ln(\cosh(\overline{y})) - \ln(\cosh(y)) - (\overline{y} - y)\tanh(y)$$

- *algorithm* $A_{2,p}$:

$$L_j(\alpha,\theta) = \frac{1}{2}|\,y(j) - G^T(x(j),\theta)\alpha\,|^2$$

$$\phi(e) = e$$

$$P_\phi(y,\overline{y}) = \frac{1}{2}|\,y - \overline{y}\,|^2$$

- *algorithm* $A_{3,p}$:

$$L_j(\alpha,\theta) = \frac{1}{4}|\,y(j) - G^T(x(j),\theta)\alpha\,|^4$$

$$\phi(e) = e^3$$

$$P_\phi(y,\overline{y}) = \frac{\overline{y}^4}{4} - \frac{y^4}{4} - (\overline{y} - y)y^3$$

- *algorithm* $A_{4,p}$:

$$L_j(\alpha,\theta) = \frac{a}{2}|\,y(j) - G^T(x(j),\theta)\alpha\,|^2 + \frac{b}{4}|\,y(j) - G^T(x(j),\theta)\alpha\,|^4$$

$$\phi(e) = ae + be^3$$

$$P_\phi(y,\bar{y}) = \frac{a}{2}|y-\bar{y}|^2 + b\left[\frac{\bar{y}^4}{4} - \frac{y^4}{4} - (\bar{y}-y)y^3\right]$$

- *algorithm* $A_{5,p}$:

$$L_j(\alpha,\theta) = \cosh(y(j) - G^T(x(j),\theta)\alpha))$$

$$\phi(e) = \sinh(e)$$

$$P_\phi(y,\bar{y}) = \cosh(\bar{y}) - \cosh(y) - (\bar{y}-y)\sinh(y)$$

The filtering algorithms, with a learning rate of

$$\mu_j = \frac{2P_\phi\left(y(j), G^T(x(j),\theta_j)\alpha_{j-1}\right)}{den}, \tag{13}$$

$$den = \phi\left(y(j) - G^T(x(j),\theta_j)\alpha_{j-1}\right)(p-1)[\phi(y(j)) - \phi(G^T(x(j),\theta_j)\alpha_{j-1})]\,\|G(x(j),\theta_j)\|_p^2,$$

$$P_\phi(y,\bar{y}) = \int_y^{\bar{y}} (\phi(r) - \phi(y))dr,$$

achieves a stability and robustness against disturbances in some sense.
For a standard algorithm for computing θ_j numerically based on (11), define

$$k^q_{\theta,\theta_{j-1}} = \begin{cases} \dfrac{2d_q(\theta,\theta_{j-1})}{\|\theta-\theta_{j-1}\|^2}, & \text{if } \theta \neq \theta_{j-1} \\ 1, & \text{if } \theta = \theta_{j-1} \end{cases}$$

to express (11) as

$$\theta_j = \arg\min_\theta [E_j(\hat{\alpha}(\theta),\theta) + \frac{\mu^{-1}_{\theta,j} k^q_{\theta,\theta_{j-1}}}{2}\|\theta-\theta_{j-1}\|^2]. \tag{14}$$

Choosing a time-invariant learning rate for θ in (14), i.e. $\mu_{\theta j} = \mu_\theta$, and estimating the elements of vector θ with different learning rates as in (10), (14) finally becomes

$$\theta_j = \arg\min_\theta [E_j(\hat{\alpha}(\theta),\theta) + \frac{k^q_{\theta,\theta_{j-1}}}{2}\|\Sigma^{-1/2}(\theta-\theta_{j-1})\|^2]. \tag{15}$$

Define vectors $r(\theta)$ and $r_q(\theta)$ as

$$r(\theta) = \begin{bmatrix} \left(\dfrac{[y(j) - G^T(x(j),\theta)\alpha_{j-1}]^2}{1 + G^T(x(j),\theta)P_j G(x(j),\theta)}\right)^{1/2} \\ \Sigma^{-1/2}(\theta-\theta_{j-1}) \end{bmatrix} \in R^{L+1}, \tag{16}$$

Algorithm 1 Fuzzy filtering

Require: Data pairs $\{x(j), y(j)\}_{j=0,1,\cdots,N}$.

Choose the number of rules K; initial guess about the antecedents θ_{-1}; positive learning rate μ for α; positive learning rates $\left(\mu_{\theta(1)}, \mu_{\theta(2)}, \cdots, \mu_{\theta(L)}\right)$ for different elements of θ; number of maximum epochs E_{max}.

Set data index $j = 0$; epoch count $EC = 0$; initial guess about consequents $\alpha_{-1} = 0$.

Choose a filtering criterion: either of result 1 or of result 2.

if filtering criterion of result 1 **then**

  Choose a value of $\gamma \geq -1$.

else {filtering criterion of result 2}

  Choose an algorithm out of the 5 different algorithms $(A_{1,p}, A_{2,p}, A_{3,p}, A_{4,p}, A_{5,p})$ and a value of p ($2 \leq p \leq \infty$).

  Define $L_j(\alpha, \theta), \phi(e), P_\phi(y, \bar{y})$ depending upon the choice of algorithm.

end if

while $EC < EC_{max}$ **do**

  while $j \leq N$ **do**

    if filtering criterion of result 1 **then**

      Define $r(\theta)$ as (16). Let $s^*(\theta)$ be the unique solution, obtained by the algorithm suggested in (Lawson & Hanson, 1995), of the following constrained linear least-squares problem:

$$s^*(\theta) = \arg\min_s [\|r(\theta) + r'(\theta)s\|^2; cs \geq h - c\theta],$$

      where $r'(\theta)$ is the Jacobian matrix of vector r with respect to θ, determined by the method of finite-differences. The Jacobian $r'(\theta)$ is a full rank matrix, since the main diagonal of the diagonal matrix Σ has positive entries.

      Compute θ_j based on (18) using a Gauss-Newton like algorithm:

$$\theta_j = \theta_{j-1} + s^*(\theta_{j-1}).$$

      Compute α_j using (8).

    else {filtering criterion of result 2}

      Define $r_q(\theta)$ as (17) and let $s_q^*(\theta)$ be the unique solution of following constrained linear least-squares problem:

$$s_q^*(\theta) = \arg\min_s [\|r_q(\theta) + r_q'(\theta)s\|^2; cs \geq h - c\theta],$$

      where $r_q'(\theta)$ is the Jacobian matrix of vector r_q with respect to θ.

      Compute θ_j based on (18) using a Gauss-Newton like algorithm:

$$\theta_j = \theta_{j-1} + s_q^*(\theta_{j-1}).$$

      Compute α_j using (12) with the learning rate provided by (13).

    end if

    $j \leftarrow j + 1$

  end while

  $EC \leftarrow EC + 1$; $\alpha_{-1} = \alpha_N$; $\theta_{-1} = \theta_N$; $j = 0$.

end while

return identified fuzzy filter parameters $\alpha^I = \alpha_N$ and $\theta^I = \theta_N$.

$$r_q(\theta) = \begin{bmatrix} \sqrt{E_j(\widehat{\alpha}(\theta),\theta)} \\ \left(\frac{k^q_{\theta,\theta_{j-1}}}{2}\right)^{1/2} \Sigma^{-1/2}(\theta - \theta_{j-1}) \end{bmatrix} \in R^{L+1}, \tag{17}$$

so that (7) and (11) can be formulated as

$$\theta_j = \begin{cases} \arg\min_{\theta} \left[\| r(\theta) \|^2 ; c\theta \geq h\right], & \textit{as per result} \\ \arg\min_{\theta} \left[\| r_q(\theta) \|^2 ; c\theta \geq h\right], & \textit{as per result} \end{cases} \tag{18}$$

Algorithm 1 presents an algorithm to estimate fuzzy filter parameters based on the filtering criteria of either result 1 or result 2. The constrained linear least-squares problem is solved by transforming first it to a least distance programming (Lawson & Hanson, 1995).
Remark 1 *Algorithm 1 estimates the parameters of the fuzzy filter of type (5). In the case of fuzzy filter of type (6), there are no matrix inequality constraints and thus linear least-squares problem will be solved at step 13 or 17 of algorithm 1.*

4. Applications in life science

The efforts have been made by the authors to develop fuzzy filtering based methods for a proper handling of the uncertainties involved in applications related to the life science (Kumar et al., 2007; Kumar et al., 2008; Kumar et al., 2007; Kumar et al., 2009; Kumar et al., 2007a; Kumar et al., 2007; Kumar et al., 2008; 2007b). This section provides a brief summary of some of the studies.

4.1 Quantitative Structure-Activity Relationship (QSAR)

4.1.1 Background

The QSAR methods developed by Hansch and Fujita (Hansch & Fujita, 1964) identify relationship between chemical structure of compounds and their activity and have been applied to chemistry and drug design (Guo, 1995; Kaiser, 1999; Jackson, 1995). The QSAR modeling is based on the principle that molecular properties like lipophilicity, shape, electronic properties modulate the biological activity of the molecule. Mathematically, biological activity is a function of molecular properties descriptors:

$$BA = f(d_1, d_2, \cdots),$$

where BA is a biological response (e.g. IC_{50}, ED_{50}, LD_{50}) and d_1,d_2, … are mathematical descriptors of molecular properties. During the last years, the applications of neural networks in chemistry and drug design has dramatically increased. A review of the field can be found e.g. in (Manallack & Livingstone, 1999; Winkler, 2004). While developing a QSAR model for the design and discovery of bioactive agents, we may come across the situation that descriptors don’t accurately capture the molecular properties relevant to the biological activity or the chosen model structure (i.e. number of adjustable model parameters) is not optimal. In such situations, there exist modeling errors. The common problems associated with QSAR modeling can be summarized as follows:

1. For the chosen structure of the model and descriptors, there may exist modeling errors. The commonly used nonlinear model training algorithms (e.g. gradient-descent based backpropagation techniques) are not robust toward modeling errors.
2. The model identification process may result in the overtraining. This leads to a loss of ability of the identified model to generalize. Although overtraining can be avoided by using validation data sets, but the computation effort to cross-validate identified models can result in large validation times for a large and diverse training data set.

4.1.2 A fuzzy filtering based method

An important issue in QSAR modeling is of robustness, i.e., model should not undergo overtraining and model performance should be least sensitive to the modeling errors associated with the chosen descriptors and structure of the model. The fuzzy filtering based method of (Kumar et al., 2007b) establishes a robust input-output mappings for QSAR studies based on fuzzy "if-then" rules. The identification of these mappings (i.e. the construction of fuzzy rules) is based on a robust criterion being referred to as "energy-gain bounding approach" (Kumar et al., 2006a). The method minimize the maximum possible value of energy-gain from modeling errors to the identification errors. The maximum value of energygain (that will be minimized) is calculated over all possible finite disturbances without making any statistical assumptions about the nature of signals. The authors in (Kumar et al., 2007b) compare their method with Bayesian regularized neural networks through the QSAR modeling examples of 1) carboquinones data set, 2) benzodiazepine data set, and 3) predicting the rate constant for hydroxyl radical tropospheric degradation of 460 heterogeneous organic compounds.

4.2 Fuzzy filtering for environmental behavior of chemicals

4.2.1 Toxicity modeling

A fundamental concern in the Quantitative Structure-Activity Relationship approach to toxicity evaluation is the generalization of the model over a wide range of compounds. The data driven modeling of toxicity, due to the complex and ill-defined nature of eco-toxicological systems, is an uncertain process. The development of a toxicity predicting model without considering uncertainties may produce a model with a low generalization performance. The work of (Kumar et al., 2007) presents a novel approach to toxicity modeling that handles the involved uncertainties using a fuzzy filter, and thus improves the generalization capability of the model. The method is illustrated by considering a data set built up by U.S. Environmental Protection Agency referring to acute toxicity 96-h LC_{50} in the fathead minnow fish (*Pimephales promelas*) (Russom et al., 1997; Pintore et al., 2003; Mazzatorta et al., 2003; Gini et al., 2004). The data set contains 568 compounds representing several chemical classes and modes of action.

4.2.2 Bioconcentration factor modeling

This work of (Kumar et al., 2009) presents a fuzzy filtering based technique for rendering robustness to the modeling methods. A case study, dealing with the development of a model for predicting the bioconcentration factor (BCF) of chemicals, was considered. The conventional neural/fuzzy BCF models, due to the involved uncertainties, may have a poor generalization performance (i.e. poor prediction performance for new chemicals). The approach of (Kumar et al., 2009) to improve the generalization performance of neural/fuzzy BCF models consists of

1. exploiting a fuzzy filter to filter out the uncertainties from the modeling problem,
2. utilizing the information about uncertainties, being provided by the fuzzy filter, for the identification of robust BCF models with an increased generalization performance.

The approach was illustrated with a data set of 511 chemicals (Dimitrov et al., 2005) taking different types of neural/fuzzy modeling techniques.

4.3 Mental stress assessment

The work presented in (Kumar et al., 2007) used fuzzy filtering for mental stress assessment via evaluating the heart rate signals. The approach consists of

1. online monitoring of heart rate signal,
2. signal processing (e.g. using the continuous wavelet transform to extract the local features of heart rate variability in time-frequency domain),
3. exploiting fuzzy clustering and fuzzy identification techniques to render robustness in heart rate variability analysis against uncertainties due to individual variations,
4. monitoring the functioning of autonomic nervous system under different stress conditions.

The experiments involved 38 physically fit subjects (26 male, 12 female, aged 18-29 years) in air traffic control task simulations.

4.4 Subjective workload score modeling

A fuzzy filtering based tool was developed in (Kumar et al., 2008) to predict the subjective workload score of the operators working in the chemistry laboratories with different levels of automation. The work proposed a fuzzy-based modeling technique that first filters out the uncertainties from the modeling problem, analyzes the uncertainties statistically using finite-mixture modeling, and, finally, utilizes the information about uncertainties for adapting the workload model to an individual's physiological conditions. The method of (Kumar et al., 2008) was demonstrated with the real-world medical data of 11 subjects who conducted an enzymatic inhibition assay in the chemistry laboratories under different workload situations.

5. References

Abonyi, J., Babuška, R. & Szeifert, F. (2002). Modified Gath-Geva Fuzzy Clustering for Identification of Takagi-Sugeno Fuzzy Models, *IEEE Trans. on System, Man and Cybernetics,Part B* pp. 612–621.

Adlassnig, K. P. (1986). Fuzzy set theory in medical diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics* 16(2): 260–265.

Adlassnig, K. P., Kolarz, G., Sheithauer, W., Effenberger, H. & Grabner, G. (1985). CADIAG: Approaches to computer-assisted medical diagnosis, *Comput. Biol. Med.* 15(5): 315–335.

Babuška, R. (1998). *Fuzzy Modeling for Control*, Kluwer Academic Publishers, Boston.

Babuška, R. & Verbruggen, H. (1997). Constructing fuzzy models by product space clustering, *in* H. Hellendoorn & D. Driankov (eds), *Fuzzy Model Identification: Selected Approaches*, Springer, Berlin, Germany, pp. 53–90.

Bellazzi, R., Guglielmann, R. & Ironi, L. (2001). Learning from biomedical time series through the integration of qualitative models and fuzzy systems, *Artificial Intelligence in Medicine* 21: 215–220.

Bellazzi, R., Ironi, L., Guglielmann, R. & Stefanelli, M. (1998). Qualitative models and fuzzy systems: an integrated approach for learning from data, *Artificial Intelligence in Medicine* 14: 5–28.

Belmonte, M., Sierra, C. & de M´antaras, R. L. (1994). RENOIR: an expert system using fuzzy logic for rheumatology diagnosis, *International Journal of Intelligent Systems* 9(11): 985– 1000.

Benfenati, E. & Gini, G. (1997). Computational predictive programs (expert systems) in toxicology, *Toxicology* 119: 213–225.

Binaghi, E., Giorgi, O. D., Maggi, G., Motta, T. & Rampini, A. (1993). Computer-assisted diagnosis of postmenopausal osteoporosis using a fuzzy expert system shell, *Computers and Biomedical Research* 26(6): 498–516.

Brai, A., Vibert, J.-F.&Koutlidis, R. (1994). An expert system for the analysis and interpretation of evoked potentials based on fuzzy classification: application to brainstem auditory evoked potentials, *Computers and Biomedical Research* 27(5): 351–366.

Burger, M., Engl, H., J.Haslinger & U.Bodenhofer (2002). Regularized data-driven construction of fuzzy controllers, *J. Inverse and Ill-posed Problems* 10: 319–344.

Chen, D. S. & Jain, R. C. (1994). A robust back propagation learning algorithm for function approximation, *IEEE Trans. Neural Networks* 5: 467–479.

Daniels, J. E., Cayton, R. M., Chappel, M. J. & Tjahjadi, T. (1997). CADOSA: A fuzzy expert system for differential daignosis of obstructive sleep apnoea and related conditions, *Expert Systems with Applications* 12(2): 163–177.

Dimitrov, S., Dimitrova, N., Parkerton, T., Comber, M., Bonnell, M. & Mekenyan, O. (2005). Base-line model for identifying the bioaccumulation potential of chemicals, *SAR and QSAR in Environmental Research* 16(6): 531–554.

Duch, W., Swaminathan, K. & Meller, J. (2007). Artificial intelligence approaches for rational drug design and discovery, *Current Pharmaceutical Design* 13: 1497–1508.

Eldred, D. V. & Jurs, P. C. (1999). Prediction of Acute Mammalian Toxicity of Organophosphorus Pesticide Compounds from Molecular Structure, *SAR and QSAR in environmental research* 10: 75–99.

Fathitorbaghan, M.&Meyer, D. (1994). MEDUSA: a fuzzy expert system for medical diagnosis of acute abdominal pain, *Methods of Information in Medicine* 33(5): 522–529.

Fukuda, O., Nagata, Y., Homma, K. & Tsuji, T. (2001). Evaluation of heart rate variability by using wavelet transformation and a recurrent neural network, *23rd Annual International Conference of the IEEE Engineering in Medicine and Biology*, Vol. 2, Istanbul, Turkey, pp. 1769–1772.

Garibaldi, J. M. & Ifeachor, E. C. (1999). Application of simulated annealing fuzzy model tuning to umbilical cord acid-base interpretation, *IEEE Transactions on Fuzzy systems* 7(1): 72–84.

Gentile, C. (2003). The robustness of the p-norm algorithms, *Machine Learning* 53(3): 265–299.

Gini, G. (2000). Predictive Toxicology of Chemicals: Experience and Impact of AI tools, *AI Magazine* 21: 81–84.

Gini, G., Cracium, M. V. & K¨onig, C. (2004). Combining Unsupervised and Supervised Artificial Neural Networks to Predict Aquatic Toxicity, *J. Chem. Inf. Comput. Sci.* 44: 1897–1902.

Gini, G., Lorenzini, M., Benfenati, E., Grasso, P. & Bruschi, M. (1999). Predictive Carcinogenicity: A Model for Aromatic Compounds, with Nitrogen-Containing Substituents, Based on Molecular Descriptors Using an Artificial Neural Network, *J. Chem. Inf. Comput. Sci.* 39: 1076–1080.

González, A. & Pérez, R. (1998). Completeness and consistency conditions for learning fuzzy rules, *Fuzzy Sets and Systems* 96: 37–51.

Guo, Z. (1995). Structure-activity relationships in medicinal chemistry: development of drug candidates from lead compounds, *Pharmacochem. Libr.* 23: 299–320.

Hansch, C. & Fujita, T. (1964). $\rho - \sigma - \pi$ Analysis. A method for the correlation of biological activity and chemical structure, *J. Am. Chem. Soc.* 86: 1616–1626.

Herrera, F., Lozano, M. & Verdegay, J. (1994). Generating fuzzy rules from examples using genetic algorithms, *Proc. 5th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'94)*, Paris, France, pp. 675–680.

Hong, X., Harris, C. J. & Chen, S. (2004). Robust neurofuzzy rule base knowledge extraction and estimation using subspace decomposition combined with regularization and doptimality, *IEEE Trans. Syst., Man., Cybern. B* 34(1): 598–608.

Jackson, R. C. (1995). Update on computer-aided drug design, *Curr. Opin. Biotechnol.* 6: 646–651.

Jang, J.-S. R. (1993). ANFIS: Adaptive-network-based fuzzy inference systems, *IEEE Transactions on Systems, Man, and Cybernetics* 23: 665–685.

Jang, J. S. R., Sun, C. T. & Mizutani, E. (1997). Neuro-Fuzzy and Soft Computing; a Computational Approach to Learning and Machine Intelligence, Prentice-Hall, Upper Saddle River.

Johansen, T. (1996). Robust identification of takagi-sugeno-kang fuzzy models using regularization, *Proc. IEEE conf. Fuzzy Systems*, New Orleans, USA, pp. 180–186.

Kaiser, K. L. E. (1999). Quantitative structure-activity relationships in chemistry, *Can. Chem. News* 51.

Kaiser, K. L. E. & Niculescu, S. P. (1999). Using Probabilistic Neural Networks to Model the Toxicity of Chemicals to the Fathead Minnow (*Pimephales promelas*): A Study Based on 865 Compounds, *Chemosphere* 38: 3237–3245.

Kim, J., Suga, Y. & Won, S. (2006). A new approach to fuzzy modeling of nonlinear dynamic systems with noise: Relevance vector learning mechanism, *IEEE Trans. on Fuzzy Systems* 14(2): 222–231.

Kumar, M., Arndt, D., Kreuzfeld, S., Thurow, K., Stoll, N. & Stoll, R. (2008). Fuzzy techniques for subjective workload score modelling under uncertainties, *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics* 38(6): 1449–1464.

Kumar, M., Stoll, N., Kaber, D., Thurow, K. & Stoll, R. (2007). Fuzzy filtering for an intelligent interpretation of medical data, *Proc. IEEE International Conference on Automation Science and Engineering (CASE 2007)*, Scottsdale, Arizona USA, pp. 225–230.

Kumar, M., Stoll, N. & Stoll, R. (2006a). An energy-gain bounding approach to robust fuzzy identification, *Automatica* 42(5): 711–721.

Kumar, M., Stoll, N. & Stoll, R. (2009a). Adaptive fuzzy filtering in a deterministic setting, *IEEE Transactions on Fuzzy Systems* 17(4): 763–776.

Kumar, M., Stoll, N. & Stoll, R. (2009b). On the estimation of parameters of takagi-sugeno fuzzy filters, *IEEE Transactions on Fuzzy Systems* 17(1): 150–166.

Kumar, M., Stoll, R. & Stoll, N. (2003a). Regularized adaptation of fuzzy inference systems. Modelling the opinion of a medical expert about physical fitness: An application, *Fuzzy Optimization and Decision Making* 2(4): 317–336.

Kumar, M., Stoll, R. & Stoll, N. (2003b). Robust adaptive fuzzy identification of time-varying processes with uncertain data. Handling uncertainties in the physical fitness fuzzy approximation with real world medical data: An application, *Fuzzy Optimization and Decision Making* 2(3): 243–259.

Kumar, M., Stoll, R. & Stoll, N. (2004a). Robust adaptive identification of fuzzy systems with uncertain data, *Fuzzy Optimization and Decision Making* 3(3): 195–216.

Kumar, M., Stoll, R. & Stoll, N. (2004b). Robust solution to fuzzy identification problem with uncertain data by regularization. Fuzzy approximation to physical fitness with real world medical data: An application, *Fuzzy Optimization and Decision Making* 3(1): 63– 82.

Kumar, M., Stoll, R. & Stoll, N. (2006b). Deterministic approach to robust adaptive learning of fuzzy models, *IEEE Trans. Syst., Man., Cybern. B* 36(4): 767–780.

Kumar, M., Stoll, R. & Stoll, N. (2006c). A robust design criterion for interpretable fuzzy models with uncertain data, *IEEE Trans. on Fuzzy Systems* 14(2): 314–328.

Kumar, M., Thurow, K., Stoll, N. & Stoll, R. (2007a). Fuzzy handling of uncertainties in modeling the inhibition of glycogen synthase kinase-3 by paullones, *Proc. IEEE International Conference on Automation Science and Engineering (CASE 2007)*, Scottsdale, Arizona USA, pp. 237–242.

Kumar, M., Thurow, K., Stoll, N. & Stoll, R. (2007b). Robust fuzzy mappings for QSAR studies, *European Journal of Medicinal Chemistry* 42: 675–685.

Kumar, M., Thurow, K., Stoll, N. & Stoll, R. (2008). A fuzzy system for modeling the structure-activity relationships in presence of uncertainties, *Proc. IEEE International Conference on Automation Science and Engineering (CASE 2008)*, Washington DC, USA, pp. 1025– 1030.

Kumar, M., Weippert, M., Vilbrandt, R., Kreuzfeld, S. & Stoll, R. (2007). Fuzzy evaluation of heart rate signals for mental stress assessment, *IEEE Transactions on Fuzzy Systems* 15(5): 791–808.

Kumar, S., Kumar, M., Stoll, R.&Kragl, U. (2007). Handling uncertainties in toxicity modeling using a fuzzy filter, *SAR and QSAR in Environmental Research* 18(7-8): 645–662.

Kumar, S., Kumar, M., Thurow, K., Stoll, R. & Kragl, U. (2009). Fuzzy filtering for robust bioconcentration factor modelling, *Environmental Modelling & Software* 24(1): 44–53.

Kuncheva, L. & Steimann, F. (1999). Fuzzy diagnosis, *Artificial Intelligence in Medicine* 16(2): 121–128.

L. Sztandera and M. Trachtman and C. Bock and J. Velga and A. Garg (2003). Soft computing and density functional theory in the design of safe textile chemicals, *in* L. M. Sztandera and C. Pastore (ed.), *Soft computing in textile sciences*, Physica-Verlag GmbH, Heidelberg, Germany.

Lawson, C. L. & Hanson, R. J. (1995). *Solving Least squares Problems*, SIAM, Philadelphia, USA.

Lee, J.W. & Lee, G. K. (2005). Design of an Adaptive Filter with a Dynamic Structure for ECG Signal Processing, *International Journal of Control, Automation, and Systems* 3(1): 137– 142.

Li, Q., Mark, R. G. & Clifford, G. D. (2008). Robust heart rate estimation from multiple asynchronous noisy sources using signal quality indices and a Kalman filter, *Physiological Measurement* 29(1): 15–32.

Lin, C.-J., Chen, C.-H. & Lin, C.-T. (2008). Efficient self-evolving evolutionary learning for neuro-fuzzy inference systems, *IEEE Trans. on Fuzzy Systems* 16(6): 1476–1490.

Lin, C. T., Ko, L. W., Chung, I. F., Huang, T. Y., Chen, Y. C., Jung, T. P. & Liang, S. F. (2006). Adaptive EEG-Based Alertness Estimation System by Using ICA-Based Fuzzy Neural Networks, *IEEE Transactions on Circuits and Systems-I: Regular Papers* 53(11): 2469– 2476.

Lindskog, P. (1997). Fuzzy identification from a grey box modeling point of view, *in* H. Hellendoorn & D. Driankov (eds), *Fuzzy Model Identification: Selected Approaches*, Springer, Berlin, Germany.

Liska, J. & Melsheimer, S. S. (1994). Complete design of fuzzy logic systems using genetic algorithms, *Proc. of the 3rd IEEE Int. Conf. on Fuzzy Systems*, pp. 1377–1382.

Lughofer, E. (2008). FLEXFIS: A Robust Incremental Learning Approach for Evolving TS Fuzzy Models, *IEEE Trans. on Fuzzy Systems* 16(6): 1393–1410.

Manallack, D. T. & Livingstone, D. J. (1999). Neural networks in drug discovery: have they lived up to their promise?, *Eur. J. Med. Chem.* 34: 195–208.

Mandryk, R. L. & Atkins, M. S. (2007). A fuzzy physiological approach for continuously modeling emotion during interaction with play technologies, *Int. J. Human-Computer Studies* 65(4): 329–347.

Mastorocostas, P. A., Tolias, Y. A., Theocharis, J. B., Hadjileontiadis, L. J. & Panas, S. M. (2000). An Orthogonal Least Squares-Based Fuzzy Filter for Real-Time Analysis of Lung Sounds, *IEEE Transactions on Biomedical Engineering* 47(9): 1165–1176.

Mazzatorta, P., Benfenati, E., Neagu, C.-D. & Gini, G. (2003). Tuning Neural and Fuzzy-Neural Networks for Toxicity Modeling, *J. Chem. inf. Comput. Sci.* 43: 513–518.

Nauck, D. & Kruse, R. (1998). A neuro-fuzzy approach to obtain interpretable fuzzy systems for function approximation, *Proc. IEEE International Conference on Fuzzy Systems 1998 (FUZZ-IEEE'98)*, Anchorage, AK, pp. 1106–1111.

Nozaki, K., Ishibuchi, H. & Tanaka, H. (1997). A simple but powerful heuristic method for generating fuzzy rules from numerical data, *Fuzzy Sets and Systems* 86: 251–270.

Philips, W. (1996). Adaptive noise removal from biomedical signals using warped polynomials, *IEEE Transactions on Biomedical Engineering* 43(5): 480–492.

Pintore, M., Piclin, N., Benfenati, E., Gini, G. & Chr´etien, J. R. (2003). Predicting Toxicity against the fathead Minnow by Adaptive Fuzzy Partition, *QSAR Comb. Sci.* 22: 210–219.

Plataniotis, K. N., Androutsos, D. & Venetsanopoulos, A. N. (1999). Adaptive fuzzy systems for multichannel signal processing, *Proceedings of the IEEE* 87(9): 1601–1622.

Rani, P., Sims, J., Brackin, R. & Sarkar, N. (2002). Online stress detection using psychophysiological signal for implicit human-robot cooperation, *Robotica* 20(6): 673–686.

Roy, M. K. & Biswas, R. (1992). I-v fuzzy relations and sanchez's approach for medical diagnosis, *Fuzzy Sets and Systems* 42: 35–38.

Russom, C. L., Bradbury, S. P. & Broderius, S. J. (1997). Predicting modes of toxic action from chemical structure: Acute toxicity in the fathead minnow (Pimephales promelas), *Environmental Toxicology and Chemistry* 16(5): 948–967.

Shan, J. J. & Fu, H. C. (1995). A fuzzy neural network for rule acquiring on fuzzy control systems, *Fuzzy Sets and Systems* 71: 345–357.

Simon, D. (2000). Design and rule base reduction of a fuzzy filter for the estimation of motor currents, *International Journal of Approximate Reasoning* 25: 145–167.

Simon, D. (2002). Training fuzzy systems with the extended kalman filter, *Fuzzy Sets and Systems* 132: 189–199.

Steimann, F. (1996). The interpretation of time-varying data with DIAMON-1, *Artificial Intelligence in Medicine* 8(4): 343–357.

Sztandera, L., Trachtman, M., Bock, C., Velga, J. & Garg, A. (2003). Soft Computing in the Design of Nontoxic Chemicals, *J. Chem. Inf. Comput. Sci.* 43: 189–198.

Thrift, P. (1991). Fuzzy logic synthesis with genetic algorithms, *Proc. of the 4th Int. Conf. on Genetic Algorithms*, pp. 509–513.

Vracko, M. (1997). A Study of Structure-Carcinogenic Potency Relationship with Artificial Neural Networks. The Using of Descriptors Related to Geometrical and Electronic Structures, *J. Chem. Inf. Comput. Sci.* 37: 1037–1043.

Wang, L. X. & Mendel, J. M. (1992). Generating fuzzy rules by learning from examples, *IEEE Trans. on Systems, Man, and Cybernetics* 22(6): 1414–1427.

Wang, W. & Vrbanek, J. (2008). An evolving fuzzy predictor for industrial applications, *IEEE Trans. on Fuzzy Systems* 16(6): 1439–1449.

Wang, W. Y., Lee, T. T., Liu, C. L. & Wang, C. H. (1997). Function approximation using fuzzy neural networks with robust learning algorithm, *IEEE Trans. Syst., Man., Cybern. B* 27: 740–747.

Watanabe, H., Yakowenko,W. J., Kim, Y. M., Anbe, J. & Tobi, T. (1994). Application of a fuzzy discrimination analysis for diagnosis of valvular heart disease, *IEEE Transactions on Fuzzy systems* 2(4): 267–276.

Wilson, G. F. & Russell, C. A. (2003a). Operator functional state classification using multiple psychophysiological features in an air traffic control task, *Human Factors* 45(3): 381–389.

Wilson, G. F. & Russell, C. A. (2003b). Real-time assessment of mental workload using psychophysiological measures and artificial neural networks, *Human Factors* 45(4): 635–643.

Winkler, D. A. (2004). Neural networks as robust tools in drug lead discovery and development, *Molecular Biotechnology* 27: 139–167.

Wong, W. S. F., Leung, K. S. & So, Y. T. (1990). The recent development and evaluation of a medical expert system (ABVAB), *International Journal of Bio-Medical Computing* 25:2-3: 223–229.

Yu, W. & Li, X. (2004). Fuzzy identification using fuzzy neural networks with stable learning algorithms, *IEEE Trans. on Fuzzy Systems* 12(3): 411–420.

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. on Systems, Man, and Cybernetics* 3: 28–44.

Zadeh, L. A. (1983). The role of fuzzy logic in the management of uncertainty in expert systems, *Fuzzy Sets Systems* 11: 199–227.

8

Information Extraction from Text – Dealing with Imprecise Data

Turksen, I.Burhan[1,2] and Celikyilmaz, Asli[3]
[1]*Department of Industrial Engineering TOBB-Economics and Technology University,*
[2]*University of Toronto,*
[3]*Computer Sciences Department University of California, Berkeley,*
[1]*Turkey*
[2]*Canada*
[3]*USA*

1. Introduction

Information Extraction from text is a special case of Data Mining where one extracts valuable information from unstructured documents. On the other hand, soft computing approaches, e.g., neural networks, fuzzy systems, deal with information processing. An architecture that can combine these processes into a complete system has been the top research field in computer and information sciences for the last decade. In this paper we will present novel methods for information processing, which can model imprecision in a given database that classical bivalent methods cannot handle. Specifically we will present novel approaches on developing soft models via function representations in place of rule based methods. We will present examples on more intelligent applications of information extraction from text and compare the performance of the novel approaches to the state-of-the-art learning methods on this field.
There have been vast amount of work on information processing, which keeps us listing them all in here. Since the aim of this chapter is to present novel approaches on information processing via fuzzy functions and their extensions, we will start with the related work on functional analysis on information processing. Later in section 3, we introduce the framework of fuzzy system modelling with fuzzy functions followed by extensions of fuzzy functions under uncertainties in section 4. Specifically, we present various fuzzy system modelling approaches via higher order fuzzy sets, e.g., interval-valued type-2 and full type-2 fuzzy modelling. Section 5 presents possible applications of the latter novel approaches on information extraction from text. In section 6 we present the results of this study and discussions for future research. Finally, in section 7 we draw conclusions.

2. Related wok on information processing with functional representations

Let us first briefly review the literature to expose a historical account of "fuzzy function?" in a variety of approaches by several authors.

Originally, "Fuzzy Functions" were defined in (Bandler & Grinder, 1976) as a connecting or overlapping of our sensory representational systems. Technically, Bandler and Grinder define "fuzzy functions" as:

"...Any modeling involving a representational system and either an input channel or an output channel in which the input or output channel involved is a different modality from the representational system with which it is being used. In traditional psychophysics, this term, 'fuzzy function', is most closely translated by the term 'synesthesia'..."

Later we find certain articles in the literature, for example, (Sasaki, 1993) and (Demirci, 1999), etc…

Turksen (2006) first introduced "Fuzzy Functions" unaware of the publications stated above and published "Fuzzy Functions with LSE" (Turksen, 2008) which is quite different in structure and intent from Sasaki and Demirci expositions. Later "Fuzzy Functions" were further developed in a variety of directions in (Celikyilmaz & Turksen, 2007; 2008a-g; 2009a-d; Turksen & Celikyilmaz, 2006).

With this perspective, Fuzzy Functions, for short, FF, are proposed for the structure identification of system models and reasoning with them. These fuzzy functions can be determined by any function identification method such as least squares' estimates, LSE, maximum likelihood estimates, MLE, support vector machine estimates, SVM (Gunn, 1998) etc. Furthermore, our work extends to Type 2 Fuzzy Functions which incorporates the parameter uncertainties in system modelling.

3. Building fuzzy system models with fuzzy functions

3.1 Background of fuzzy rule bases

Traditional FIS structure is based on the fuzzy rule base (FRB) (*if-then* rules) structures,

$$R_i: \textit{IF antecedent}_i \textit{ THEN consequent}_i \quad (1)$$

In (1) each R_i, $i=1\ldots c$, represents one fuzzy rule. Based on the representation of the consequents structure, FISs get the name; *Linguistic* FIS when the consequents are represented with fuzzy sets as in (Zadeh, 1965), *Mizumoto* FIS (Mizumoto, 1989) when the consequents are represented with a scalar value, *Takagi-Sugeno FRB* (Takagi & Sugeno, 1985) when the consequents are represented with linear or non-linear equations of input variables. For illustration, *Takagi-Sugeno* FIS structure is defined as;

$$R_i: \textit{IF } AND_i^{NV} (x_j \in X_j \text{ is } A_{ij}) \textit{ THEN } y_i = a_i x^T + b_i \quad (2)$$

In (2) A_{ij} is the type-1 fuzzy set characterized by a type-1 membership function, $\mu_A(x_j) \rightarrow [0,1]$, where $x_j \in X_j$ is the jth input variable. $a_i=(a_{i,1}\ldots a_{i,NV})$ and b_i are regression coefficients of ith rule. A type-1 fuzzy set is identified for each input variable, assuming they are independent from one another, viz. non-interactivity assumption. Fuzzy connectives such as t-norm are used to combine antecedent fuzzy sets to calculate the degree of fire of each rule.

The traditional FIS structures presented above have various challenges that should not be neglected (Turksen & Celikyilmaz, 2006). Among some of these challenges are identification of the; types of antecedent and consequent membership functions, and their varying parameters, most suitable combination operators (t-norm, t-conorm, etc.), conjunction operators during aggregation of antecedents, and consequents, implication operator types to capture uncertainty associated with the linguistic "AND", "OR", "IMP" for the representation of the rules, and reasoning with them, type of defuzzification method, etc.

The literature indicates that a given FIS model performance can be slightly affected by the change in t-norm values. Nevertheless, one still needs to decide the type of t-norm and t-conorm operators. Over the course of many years these challenges have been investigated to reduce the fuzzy operations (Babuska & Verbruggen, 1997), and expert intervention and many different methods are proposed such as building hybrid fuzzy systems using other soft computing methods via genetic algorithms or neural networks, etc.
Some extensions of traditional FISs e.g., (Uncu et.al., 2004), assume that antecedent fuzzy sets are dependent on each other (interactive), so in these systems an entire antecedent part of a given rule is represented with a single type-1 fuzzy set. Such FIS structures are expressed as follows:

$$R_i\text{: } IF\ x \in X \text{ is } A_i\ THEN\ y_i = a_i x^T + b_i \quad (3)$$

In (3) the fuzzy set A_i is characterized by a type-1 membership function $\mu_i(x) \rightarrow [0,1]$ where $x \in X$ is an input vector.
Later, the performance of latter systems is improved with the implementation of improved fuzzy functions algorithm (Celikyilmaz & Turksen, 2008a-g). Next subsection briefly reviews such systems, which forms the basis of the Type-2 Fuzzy Functions.

3.2 Enhanced FIS with improved fuzzy functions

Although FSM approach based on Fuzzy Functions in Fig. 1 and traditional FSM approaches based on FRB structures (Takagi & Sugeno, 1985; Emami et.al. 1998; Bodur et al., etc.) share similar system design steps, they differ in structure identification, namely in finding the fuzzy models (rules) for each pattern identified. The new FFs approach first clusters a given data into several overlapping fuzzy clusters, each of which is used to define a separate decision rule. Fuzzy c-means clustering (FCM) (Bezdek, 1984) has been the main clustering algorithm utilized in these methods to find fuzzy partitions so far. The novelty of the FFs approaches are that, during structure identification, similarity of the objects are enhanced with additional fuzzy identifiers viz. membership values, by utilizing them as additional predictors of the system model along with the original input variables to estimate the local relations of the input-output data. Thus, membership values and their list of possible (user-defined) transformations are augmented to original dataset as new dimensions to structure different representations for each cluster.

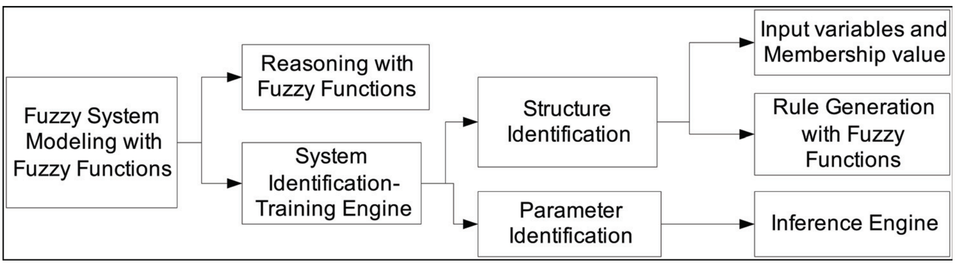


Fig. 1. Framework of fuzzy system models with fuzzy functions approach.

In (Celikyilmaz & Turksen, 2008b) a new fuzzy clustering algorithm is proposed, namely Improved Fuzzy Clustering (IFC) algorithm, which carries out two objectives: (i) to find good representation of the partition matrix, which captures the multiple model structure of the given system by identifying the hidden patterns, (ii) to find the membership values, which are good predictors of the regression models of each cluster. Therefore the objective function of the new IFC is designed based on these two objectives. The novelty of the presented fuzzy clustering approach, which aparts itself from the earlier improved fuzzy clustering approaches by (Chen et al. 1998; Höppner & Klawonn, 2003; Menard, 2001) is that, during IFC optimization, regression models, to be build for each cluster, will use only membership values measured at a particular iteration and their user defined transformations, but not the original input variables. Alienating original input variables and building regression models with membership values will shape the memberships into candidate inputs to explain the output variable for each local model. As a result of this improvement, the new IFC introduces a new membership function. In the proposed IFC, we hypothesize to find membership values that can increase the prediction power of the system modeling with FFs. In this sense, the resulting fuzzy functions are referred as "improved fuzzy functions (IFF)".

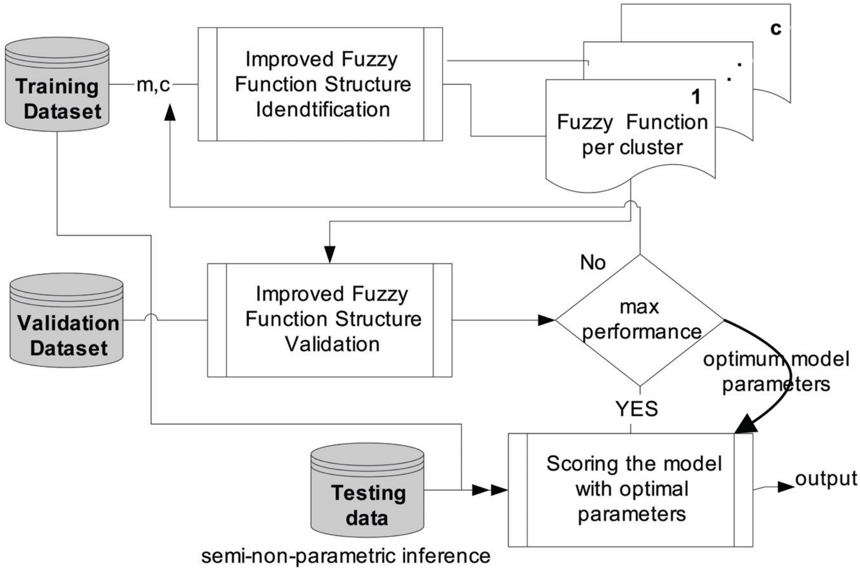


Fig. 2. Flow chart of Fuzzy System Models with Fuzzy Functions Approach.

Structure identification of FIS with Fuzzy Functions Systems is based on Improved Fuzzy Clustering (IFC) algorithm to identify the hidden structures in a given dataset. The learning algorithm is sketched in Fig.2.

The type-1 FIS with Improved Fuzzy Functions (Celikyilmaz & Turksen, 2007, 2008b) is designed to eliminate most of the aforementioned fuzzy operations of traditional type-1 FIS. In somewhat simplified view, such fuzzy systems work as follows:

- The domain $X \subseteq \Re^{nv}$ with nv dimensional input space is partitioned into c overlapping clusters using IFC, and each cluster is represented with cluster centers, V_i, i=1,..,c, and membership value matrix, U_i.
- To each of these regions a local fuzzy model f_i: $V_i \rightarrow \Re$ is assigned by using membership values as additional predictors to given input vector, $x \in X$. The system then identifies

one fuzzy output from each fuzzy model and then weights these outputs based on the membership values of the given input vector in each cluster.

Let (x_k,y_k) denote each training data point, where $x_k(x_{1,k}\ldots x_{nv,k})$, is the kth input vector of nv dimensions, y_k, is their output value, $\mu_{ik}\in[0,1]$ represent the membership value of kth vector to cluster $i=1\ldots c$, c be the total number of clusters, m, be the level of fuzziness parameter. The learning algorithm of type-1 FIS with the Improved Fuzzy Functions approach (Celikyilmaz & Turksen, 2007; 2008b;c) is processed as follows:

Step 1: IFC is a dual-structure clustering method combining FCM (Bezdek, 1984) and fuzzy c-regression algorithms (Höppner & Klawonn, 2003) within one clustering schema and has the following objective function:

$$\min J_m^{IFC} = \sum\nolimits_{i=1}^{c}\sum\nolimits_{k=1}^{n}\mu_{ik}^m d_{ik}^2 + \sum\nolimits_{i=1}^{c}\sum\nolimits_{k=1}^{n}\mu_{ik}^m E_{ik} \tag{4}$$

In (4), $d_{ik}=||x_k-v_i||$, represents the Euclidean distance of each x_k to each cluster center, v_i. The error $E_{ik}=(y_k-g_i(\tau_{ik}))^2$ is the total squared deviation between of the approximated fuzzy models, namely the *interim* fuzzy functions, $g_i(\tau_i)$ of cluster i and the actual output. The novelty of each $g_i(\tau_i)$ is that corresponding membership values and their possible transformations are the only predictors of interim fuzzy functions, while excluding original variables. The aim is to calculate the membership values that can be candidate input variables when used to estimate the local models. An example interim fuzzy function can be formed using:

$$g_i(\tau_i,\hat{w}_i) = \hat{w}_{0i} + \hat{w}_{1i}\mu_i + \hat{w}_{2i}(1+\exp(-\mu_{ik}^m)) \tag{5}$$

In (5), $\hat{w}_i$ represents the vector of regression coefficients. IFC minimizes the objective function, J_m^{IFC}. The second term of the objective function can be minimized if optimum functions can be found. Thus, the algorithm searches for the best interim fuzzy functions, $g_i(\tau_i)$.

From the Lagrange transformation of the objective function in (4) the membership values are calculated with a new membership value update equation as follows,

$$\mu_{ik} = \left(\sum_{j=1}^{c}\left[\left(d_{ik}^2+E_{ik}^2\right)\Big/\left(d_{jk}^2+E_{jk}^2\right)\right]^{1/(m-1)}\right)^{-1};\sum_{i=1}^{c}\mu_{ik}=1 \tag{6}$$

, $i=1\ldots c$, $k=1\ldots n$. Punishing the objective function with an additional error, forces to capture the membership values that would help to improve the local models, but at the same time identify the clusters. Thus, the new membership function yields a matrix of "*improved*" membership values, $\mu_{ik}^*\in U^*\subset\Re^{n\times c}$. It has been proven that the improved membership values obtained from the IFC can predict the local relations better than the membership values obtained from the FCM clustering algorithm.

Proposed IFC optimization method searches for optimum membership values, which are to be used later as additional predictors to estimate parameters of Fuzzy Functions of a given system model. The structures of functions to be approximated depend on distribution of membership values with an output variable. One should choose appropriate membership value transformations to approximate output variable. For any given fuzzifier m and number of clusters c the outputs of the IFC algorithm are as follows:

- optimum parameters of fuzzy functions f(τi) of each cluster ŵi, i=1…c, that are captured from the last iteration step,
- structure of the input matrix, **τ**i, viz. the list of different types of membership value transformations that are used to approximate each *f*(**τ**i) during IFC,
- optimized membership matrix, U*(x,y), the cluster centers v*(x,y)

(*) indicates the optimum results from the new IFC algorithm.

Step 2: One fuzzy function is approximated for each cluster to identify the input-output relations in local model for each cluster *i*. The dataset of each cluster is comprised of the original input variables, *x,* improved membership values of particular cluster *i* obtained from IFC, and their user defined transformations. This is same as mapping the input space, $\Re^{nv}$, of each individual cluster *i* onto a higher dimensional feature space $\Re^{nv+nm}$, i.e., $x \rightarrow \Phi_i(x,\mu_i^*)$, where *nm* is the total number of membership value transformations used to structure a system of *principle fuzzy functions*. Parameters of an optimum regression function are sought in this new space. The principle fuzzy functions, $\hat{f}_i(\Phi_i)$, to determine the local relations of each cluster are structured in *(nv+nm)* space.

The interim fuzzy functions, $g_i(\tau_i)$ are different from principle fuzzy functions $\hat{f}_i(\Phi_i)$, since $g_i(\tau_i)$ is used only for shaping the membership functions during IFC algorithm and only use membership values and their transformations only as input variables. A prominent feature of the principle fuzzy function approximation of such forms is that, if the relations between input and output variables cannot be defined in the original space, we can use proposed fuzzy functions approach to explain their relationship in the $\Re^{nv+nm}$ space.

Step 3: An approximate optimum number of clusters, *c**, of IFC algorithm is determined with the cluster validity index, *cviFF* (Celikyilmaz & Turksen, 2009a;2008c), designed to evaluate the IFC algorithm with:

$$cviFF = \frac{vc^*}{(c \cdot vs^*)+1}, \quad vc^* = \max \frac{1}{n}\sum_{k=1}^{n} \mu_{ik}^{*m}\left(d_{ik}^{\ 2} + (y_k - \hat{f}_{ik}(\Phi_i))^2\right)$$
$$vs^* = \begin{cases} \min\limits_{i,j\neq i}\left(\left\|v_i - v_j\right\|^2 / \left|\langle \alpha_i, \alpha_j \rangle\right|\right), & if \left|\langle \alpha_i, \alpha_j \rangle\right| \neq 0, \\ \min\limits_{i,j\neq i}\left\|v_i - v_j\right\|^2, & otherwise \end{cases} \tag{7}$$

In (7) vc^* represents the *compactness* and vs^* represents the *separability*. vc^* combines within-cluster distances and errors between actual and estimated output obtained from *c* number of principle fuzzy functions. The v_i and v_j i,j=1,..,c, $i \neq j$ represent the cluster center vectors of two separate clusters of an IFC model. vs^* determines the structure of clusters by measuring the ratio of cluster center distances to the angle between their regression functions. The a_i in the $|\langle a_i,a_j \rangle| \in [0,1]$, $i,j=1,\ldots,c$, is the unit normal vector of each principle fuzzy function *i*, $\hat{f}_i(\Phi_i)$, $a_i=[n_i]/||n_i||$. The absolute value of inner product of unit vectors of two fuzzy functions of two different clusters, $|\langle a_i,a_j \rangle| \in [0,1]$, $i,j=1,\ldots,c$, $i \neq j$, equals to the value of *cosine* of the angle between them: $cos\theta_{i,j} = \langle n_i,n_j \rangle / |n_i| * |n_j| = \langle a_i,a_j \rangle$. When two cluster centers are too close to each other due to oversized number of clusters, the distance between them becomes almost (≅0) invisible, then validity measure goes to infinity. To prevent this, the denominator of *cviFF* in (7) is increased by 1.

Any regression approximation method can be employed to identify the parameters of local functions, e.g. LSE or soft computing approaches such as neural networks or support vector machines (SVM) (Gunn, 1998). For instance, when LSE is used to identify the local models of a cluster i, the principle fuzzy function is formed with function as:

$$\hat{y}_i = \hat{f}_i(x, \Phi_i) = \beta_{0,i} + \beta_{1,i}\mu_i^* + \beta_{2,i}x \tag{8}$$

Step 4: Finally, one crisp output is obtained by taking the average weight of the outputs from each principle function i, with corresponding membership values as follows:

$$\hat{y} = \sum_{i=1}^{c^*} \mu_i^* \hat{f}_i(\Phi_i) \tag{9}$$

The experiments indicate that the FIS system based on Fuzzy Functions (Turksen, 2008; Celikyilmaz & Turksen, 2008a) outperform traditional type-1 FIS as well as other soft computing approaches. One of the issues of this approach is that since type-1 fuzzy sets are implemented, it may not be possible to handle uncertainties. In particular, there is also the uncertainty in determining the system parameters such as; type of membership value transformations (τ_i) used during IFC algorithm (such as in (5)) and during shaping principle fuzzy functions, $\hat{f}_i(\Phi_i)$ (such as in (8)). Hence, we implement interval type-2 fuzzy sets into fuzzy functions system. Using the type-2 FIS instead of type-1 FIS in Fuzzy Function systems has many advantages, which are summarized as follows:

- The type-2 fuzzy sets can handle the numerical uncertainties in inputs and outputs of fuzzy functions,
- The uncertainty in determining the type, and parameters of membership value extraction functions are managed,
- The type-2 fuzzy sets are discretisized into a large number of embedded type-1 fuzzy sets, which enable a wealthy environment to describe the local input-output relations.

The new type-2 FIS based on Fuzzy Functions is designed that can characterize structure of optimum membership value transformations $\Omega=\{\tau_i,\Phi_i\}$ of given fuzzy function, the shape of membership values, the number and type of fuzzy function structures, and number of local structures. In summary, the proposed approach searches for the optimum uncertainty interval of membership functions and optimum list of the fuzzy function structures for each local model using soft computing approaches such as genetic algorithms.

4. Modelling uncertainty with fuzzy functions

4.1 Review of type-2 fuzzy inference systems and variations

Before we present the new type-2 FIS based on Fuzzy Functions, we briefly review the traditional type-2 FISs. For the generalized type-2 case, where the secondary membership functions, the third dimension, are of *any* type, there is a significant computational complexity that has delayed their development (Coupland & John, 2007). Thus, in most type-2 fuzzy logic research, the interval type-2 fuzzy sets are. Nonetheless, recent investigations on full type-2 fuzzy logic systems such as (Coupland & John, 2007) or (Celikyilmaz & Turksen, 2008c) present promising results.

A type-2 fuzzy set $\tilde{A}$ is characterized by a type-2 membership function $\mu_{\tilde{A}}(x,u)$, where $x \in X$ and $u \in J_x \subseteq [0,1]$, i.e.,

$$\tilde{A} = \left\{ \left(x, \left(\mu_{\tilde{A}}, u \right) \right) \mid x \in X, \mu_{\tilde{A}} \in J_x \subseteq [0,1] \right\}. \tag{10}$$

The elements of the domain of $\mu_{\tilde{A}}(x)$ are called the *primary memberships* of x in $\tilde{A}$, and the membership functions of the primary memberships in $\mu_{\tilde{A}}(x)$ are called the *secondary memberships* of x in $\tilde{A}$.

The interval fuzzy logic systems are embedded type-1 fuzzy inference systems, which implement fuzzy sets, $\tilde{A}$. In (10) J_x is a set of real values with finite elements. A special case of interval-valued type-2 FIS is formalized with the fuzzy sets of discrete domain as follows:

$$\tilde{A}^i = \{ (x, (\mu, 1)) \mid x \in X, \mu \in \{\mu_i\}, \mu_i \in [0,1] \} \tag{11}$$

In (11), the membership functions are discretisized and are used to form a collection of embedded type-1 FIS. Hence, ith rule in a type-2 system having nv inputs $x_1 \in X_1 \ldots x_{nv} \in X_{nv}$ and one output $y \in Y$ is represented with;

$$\tilde{R}_i : \; IF \; AND_{j=1}^{nv} \; (x_j \in X \; is \; \tilde{A}_{ji}) \; THEN \; y_i \; is \; \tilde{B}_i \tag{12}$$

The uncertainty in primary membership functions of a type-2 fuzzy set $\tilde{A}$, is represented with a bounded region that is called the *foot-print of uncertainty* (FOU). It is the union of all the primary membership functions. With the implementation of type-2 fuzzy sets, determining the optimum type-1 membership function reduces its significance.

In order to extract crisp output, the type of the set is first reduced with *a type reduction* process, which is an extension of defuzzification method. Then type reduced set is defuzzified to obtain a zero order (crisp) output. The foundations of type-2 fuzzy logic system are explained in (Mendel, 2001) in more detail.

The type-2 fuzzy set parameters associated with each variable in each rule are identified mostly using supervised learning methods. In (Uncu et.al., 2004) the FCM (Bezdek, 1984) clustering is used to identify the hidden structures. They use uncertainty in selection of *level of fuzziness* parameter, m, of FCM as the source of uncertainty of the values of inference parameters and identify embedded type-1 FIS for each m to represent discrete interval type-2 FIS (DIT2FIS). Let m^r be the r^{th} level of fuzziness, $m^r \in \{m^1 .. m^{NM}\}$, where NM is the number of disjoint m values. Thus, they find r^{th} embedded type-1 fuzzy rule for each different m^r. $\mu_A{}^r$ represents the membership values associated with r^{th} embedded type-1 fuzzy set A. Their Tagaki-Sugeno FIS is as follows:

$$\tilde{R}_i^r : \text{IF } x \in X \text{ is } A_i{}^r \text{ THEN } y_i{}^r = a_i{}^r x^T + b_i{}^r \tag{13}$$

In (13) $r=1 \ldots NM$, and $a_i{}^r \, x^T + b_i{}^r$ are regression coefficients associated with i^{th} rule of r^{th} embedded type-1 fuzzy rule. Thus, the problem of building type-2 FIS in DIT2FIS is reduced to finding traditional embedded type-1 FISs.

Type-2 FIS based on Fuzzy functions (Celikyilmaz & Turksen, 2009c;2008a) is a different approach to uncertainty modeling which extends inference strategy of (Uncu et.al., 2004) by introducing two separate uncertainty parameters, the level of fuzziness and the fuzzy function structures to form interval type-2 fuzzy sets. In the next we will briefly present type-2 fuzzy functions methods.

4.2 Type-2 fuzzy functions

4.2.1 Interval valued type-2 fuzzy functions

The interval Valued Type-2 Fuzzy Functions, IVT2FF in short, evidently differs from the other type-2 FIS of the previous sections in many ways. For instance, instead of the traditional FIS such as Tagaki-Sugeno structures, the algorithm is based on the Fuzzy Functions structures (Turksen, 2008), which do not require fuzzy connectives (aggregation, implication, defuzzification) and introduce a new fuzzy clustering algorithm. In addition, the uncertainty interval of membership values are identified based on two different sources of imprecision: (i) selection of the level of fuzziness parameter, *m,* of IFC by identifying an *m-bound* (ii) determination of the list of optimum structures of fuzzy functions by identifying optimum forms of membership values.

IVT2FF is an iterative hybrid system, in which, the structure is learnt and parameters are tuned by a genetic learning algorithm, to determine the hidden structures viz. information points, which is the fundamental concept of the system identification. The ET2FF has three fundamental phases:

- **Phase 1**: Determination of the optimum uncertainty interval of the membership functions – FOU and optimum list of fuzzy functions and optimum values of other parameters with a soft computing algorithm. Here we use genetic learning process, although other optimization methods can be used as well.
- **Phase 2**: Type-2 FIS structure identification.
- **Phase 3**: Inference for testing dataset.

Phase 1: Genetic Learning Process (GLP). The idea is to create an optimization framework, using a soft computing method, e.g., Genetic Algorithms (GA) (Goldberg, 1989) to find the optimum system parameters and boundaries of the level fuzziness parameter to define boundaries for membership functions and the list of fuzzy functions that are most suitable for estimating local dependencies. Hence, the structure of each chromosome in GA framework encodes given type-2 FIS parameters, which are parameters of Improved Fuzzy Clustering (IFC) (Celikyilmaz & Turksen, 2008b) algorithm and fuzzy function structures. The parameter genes, in sequence, are composed of: two of the IFC clustering parameters, *m-lower* and *m-upper* $\in[1.01, \infty]$ and the type of the regression method, e.g. {*1='(linear regression) LSE', 2='(non-lienar regression) SVM', etc*}, The rest of the parameter genes depend on the type of regression method. If SVM is used to construct more complex non-linear fuzzy functions, three additional SVM parameters, *Creg*, *epsilon* and *kernel type*, are set up as additional alleles in the chromosome.

The rest of the *nm* different alleles represent the membership value transformations to be used to shape fuzzy functions. Among many different types, in our models we used power sets, exponential, sigmoid, logistic transformations, etc., of membership values as additional inputs. Each chromosome represents parameters of two separate models of type-1 FIS with Fuzzy Functions using two different *m* values, each of which has the same fuzzy function structure and regression parameters. Each individual in the population have different parameters and *m* boundaries so that population is diverse.

The optimum number of cluster, c^* is fixed based on *cviFF* validity index of Fuzzy Function systems before GLP is processed. At the start of the GLP a wide range is assigned for the boundary values of *m-interval,* e.g.. {*m-lower*=*1.2, m-upper*=*7*}. For each chromosome, two separate type-1 FIS are constructed using each *m-bound* and parameters of the rest of the alleles.

In Fig. 3, FOU of the membership functions and fuzzy functions before and after GLP is shown. Note that these membership functions are the idealized representations of the membership values obtained from the IFC method. We do not curve fit the membership values into membership function in the actual calculations.

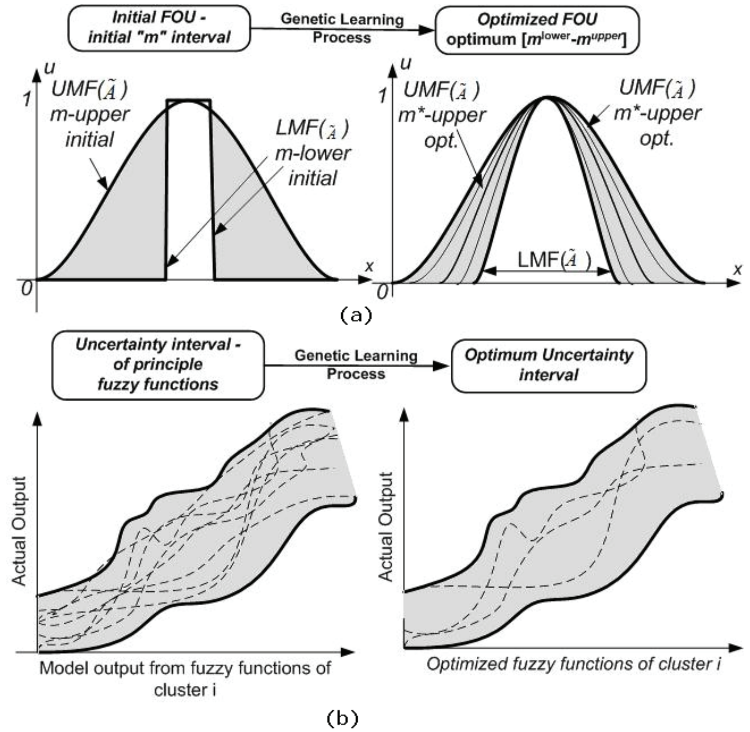


Fig. 3. Optimization using Genetic Learning Process. FOU of (a) idealized representation of the membership functions (MF), (b) output from principle fuzzy functions, UMF=Upper MF, LMF=Lower MF.

The membership functions, the top graphs, are predicted via IFC method. They are mainly based on two parameters, the level of fuzziness (m) and the structure of the interim fuzzy functions, $g_i(\tau_i)$, (as seen in (5) and (6)). The lower and upper membership functions-LMF($\tilde{A}$) and UMF($\tilde{A}$)- of the graph in Fig. 3.a on the left is formed using the initial *m-lower* and *m-upper* and the initial interim fuzzy function structures for the IFC method.

The interim fuzzy function parameters are randomly determined by the fuzzy function type and structure alleles (control genes) of each chromosome. They represent different forms of the membership values to be used to identify the interim fuzzy functions. In between the upper and lower boundaries of the shaded area- FOU any other type-1 membership value distribution can be formed using any value from [*m-lower, m-upper*] interval or any fuzzy function structure by combining different membership value transformations (Fig. 4). After IFC, two type-1 FIS are constructed using membership values and original input variables to build fuzzy functions to represent each local model.

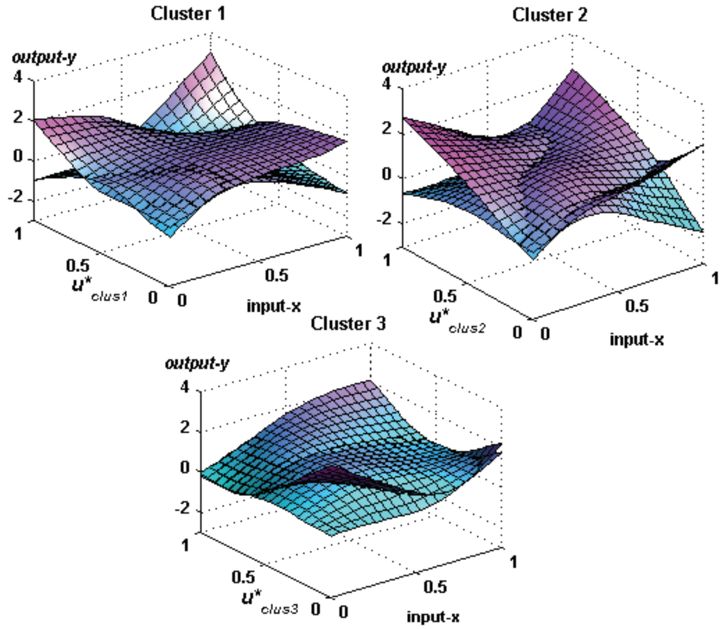


Fig. 4. Decision surfaces - $f(x,e^{\mu})$ obtained from GLP using parameters, SVM-Gaussian Kernel allele=0 {Non-linear} and $(m_{low},m_{up},Creg,\varepsilon)$={1.75,2.00,54.5,0.115}, c^*=3. u_{clusi} represents membership values of corresponding cluster *i*.

The algorithm starts with a larger interval of parameter values and optimizes the interval based on the fitness of each chromosome obtained from the combination of the boundary type-1 FISs. The fitness is evaluated as follows:

$$Fitness_p = -\sqrt{\frac{1}{n}\sum_{k=1}^{n}\left[(y_k - \hat{y}_{k,p}^{(m^{lower},\Omega)})^2 + (y_k - \hat{y}_{k,p}^{(m^{upper},\Omega)})^2\right]} \quad (14)$$

'*p*' is the *population-size*, Ω is the optimum parameter list. The algorithm searches for the optimum model parameters and the *m-bound* so that the two type-1 FIS models would have the minimum error. Hence, the algorithm starts with a larger *m-bound* and gradually shifts to where the $Fitness_p$ is maximized. To ensure that the fitness function increases monotonically, the best candidate solution in each generation enters the next generation directly.

Phase 2: Type-2 FIS Structure Identification. The optimum uncertainty intervals – FOU and the list of optimum fuzzy functions- determined in the previous step, are discretisized to find as many embedded type-1 FIS with fuzzy functions as feasible. The IVFF essentially is comprised of collection of embedded type-1 FISs.

Each embedded type-1 FIS defines a list of fuzzy functions for each cluster. These functions may or may not have the same input variables because each function of each cluster may be formed with a different membership value transformation used as additional inputs that best describes the local structure. Each fuzzy function would have a different membership value as a variable and its different possible transformations to approximate the fuzzy functions. The algorithm presented here captures the best model parameters in cluster level among the embedded fuzzy models, one for each training vector, and keeps them in a matrix (collection table) to be used for reasoning.

Using the optimum parameters, from the previoys step the following steps are processed:

***Step-1**:* The optimum *m* interval, [*m-low**,*m-up**] is discretisized into a list of disjoint *m* values. On the other hand, the optimum fuzzy function structures include information on different types of membership value transformations that can be used in formation of interim and principle fuzzy functions as additional inputs.

***Step-2**:* For each combination of discrete parameters, IFC clustering is applied to partition the data into c^* clusters and calculate improved membership values. Membership values of the input space are calculated using IFC membership function in (6). For each discrete point x', different membership values are obtained from the IFC model using the list of learning parameter set.

***Step-3**:* Fuzzy functions, $f_i^{r,s}$, $i=1,\ldots c^*$, of each embedded type-1 FIS model are determined using each set of discrete parameters and improved membership values using the functions such as in (8) depending on the model type.

For each cluster, only one of these approximated functions can explain the output better than rest of embedded functions. For instance, Fig. 5 depicts prediction performance of four different types of linear fuzzy functions of a single cluster using different *m* values based on root mean square error (RMSE). These four functions are formulized using different forms of membership value transformations shown in the label of in Fig.5. Every point corresponds to one function of a specific cluster. One specific model with a specific *m* value can reduce the error better than others. In another cluster, these results might be different and different fuzzy functions for different fuzziness levels could be more preferable. We need to determine the best functions obtained from different sets of parameters. This corresponds to finding the best embedded type-1 FIS model for each training vector using type-2 FIS system.

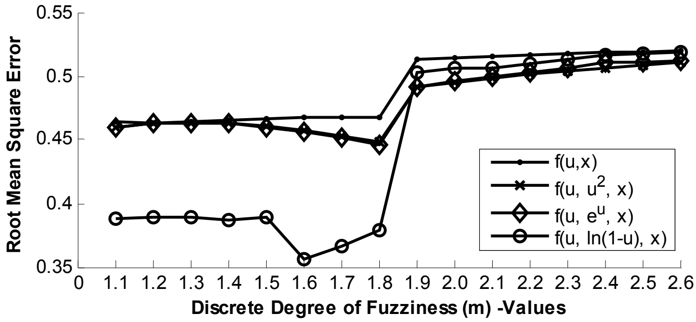


Fig. 5. The uncertainty in choosing the *m* values as a function of the error measure of the proposed type-2 FIS (ET2FF) - RMSE values as a function of degree of fuzziness (*m*) for four different fuzzy function structures. *u*: improved membership values.

***Step-4**:* We find the parameters of each cluster that would give the minimum local fuzzy function error.

4.2.2 Full type-2 fuzzy functions

Interval type-2 fuzzy sets (IT2FS) are simplified forms of full type-2 fuzzy sets (FT2FS), where the secondary MEMBERSHIP FUNCTIONs are unified, e.g., equal to 1. Interval IT2FS identify footprint-of-uncertainty (FOU) as depicted in Fig. 6.

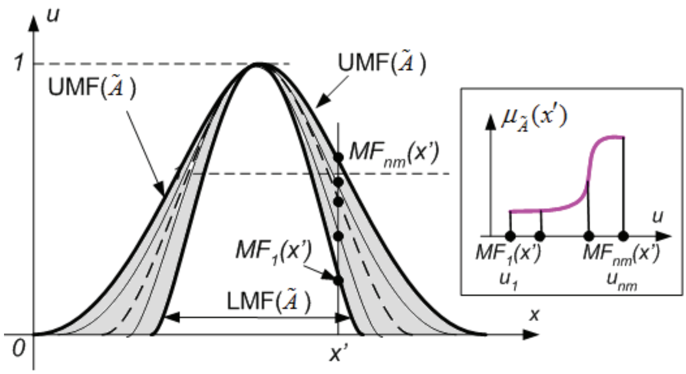


Fig. 6. Membership functions where base-end-points have uncertainty intervals. The insert represents secondary MEMBERSHIP FUNCTION of x'.

FOU of a FT2FS $\tilde{A}$ is the uncertainty region (2D-region) specified by lower and upper membership functions (membership functions), LMF($\tilde{A}$), UMF($\tilde{A}$). For each data point, x', there can be nm=2,..,∞ different membership functions within this interval. Hence, FT2FS have secondary grades, which sit on top of FOU to form the 3D region.

In different studies, e.g., (Celikyilmaz & Turksen, 2008e;f), uncertainties of parameters from imperfect information are investigated using fuzzy clustering algorithm. In particular, the FOU of the IT2FS are formed based on the level of fuzziness parameter of FCM clustering.

In fuzzy clustering methods, fuzziness is measured by the level of fuzziness parameter, m, which determines the degree of overlap between the clusters, viz. structures, granules, etc., identified in the given dataset. In many research, identification of the footprint_of_uncertainty of membership functions of FCM clustering algorithm, e.g., (Hwang & Rhee, 2007; Celikyilmaz & Turksen, 2008e), or hybrid clustering algorithms (Celikyilmaz & Turksen, 2008f) is based on the level of fuzziness parameter. One can investigate the level of fuzziness, m, of particularly fuzzy c-regression model (FCRM) clustering methods (Hathaway & Bezdek, 1993), instead of conventional clustering algorithms. In building fuzzy inference systems, separate functions are identified for each local input-output relation, which are defined with hyperplanes. Therefore, a better way is to construct hyperplane-shaped clusters.

Thus, we presented a new type-2 fuzzy inference method (Celikyilmaz & Turksen, 2008g), which can identify the optimum secondary membershp function grades, i.e., weights, of the primary MF grades using genetic algorithms. New data vectors adopt the secondary membership function grades obtained from the training samples in their neighborhood. During genetic learning process, each individual in the population encodes these weights for each training vector for each cluster, separately. This is quite cumbersome process when the number of training vectors are large therefore it is simplified in this paper by implementing transductive learning method. Instead of learning the secondary MF grades of the entire training dataset, for each new data point a new set of weights are learnt from fairly less training vectors, which are close to this new vector in distance. Experimental analysis demonstrates the performance of the new approach.

The distibution of secondary membership functions is demonstrated in Fig. 7 using an artificial dataset. The dataset ontains single input and single output with two local structures; therefore, the number of clusters is set to two. The primary MF grades, u(x) values, are obtained from FCRM model using list of levels of fuzziness parameter m={1.1,1.25,..,2.6} as shown in Fig. 7 top-right graph, also the base of the 3D graph , the bottom graph in Fig. 7. The bottom 3-D graph in Fig. 7 displays secondary membership function of a single point $x_k=0.5$. The secondary membership function values of nearest data points are optimized with genetic algorithms.

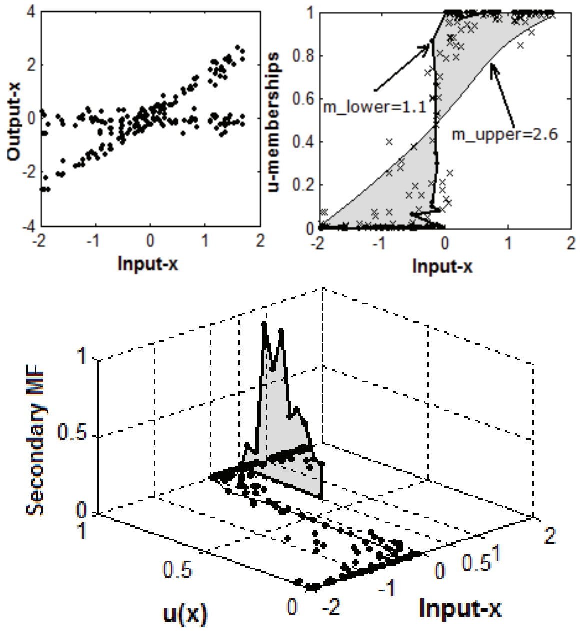


Fig. 7. (Top-left) Artificial Dataset, (Top-right) FOU by $m \in [1.1, 2.6]$, (Bottom) secondary MF of data point $x'=0.5$.

5. Experiments on text mining

In this paper we present various different fuzzy function approaches which is a summary of our research for the last five years. Our experiments have shown that as we introduce the uncertainty, we gain more performance from the models that we build to represent the real systems, i.e., variaous natual language processing applications on infomration retrieval and information extraction. Hence, the interval type-2 fuzzy system models based on fuzzy functions have shown better performance improvement compared to the type-2 fuzzy function models (Celikyilmaz & Turksen, 2008a). Later on we have developed the full type-2 fuzzy functions method with which we can introduce second-order uncertainties to the system model. The results have shown that the full type-2 fuzzy functions can improve the perforamnce of fuzzy system models when there is uncertainty. Since natural language appliations are imprecise in nature, we prefer to use full type-2 fuzzy functions when building language models. In addition the space limitations keep us presenting all the result from different our different system modeling approaches. Hence, in the next we will present the result of our experiments using Full Type-2 Fuzzy Functions, in other words, Type-2 Fuzzy Inference System (T2FIS) presented in this paper. We will build a Question and Answering (QA) system.

The aim of QA systems is to find precise answers to natural language questions from large document collections by processing several modules in sequence including question analysis, document retrieval, answer extraction and answer selection. In this paper we are particularly interested in answer selection part, in which retrieved candidate answers are ranked based on a *textual entailment* model[1]. An entailment relation between two text snippets (text-hypothesis pair) is produced when the meaning of the hypothesis meaning can be inferred from the meaning of text.

Inasmuch our QA system is designed to return candidate sentences from a corpus, instead of returning exact answer phrases, such as we return the sentence containing the answer-phrase but not extract the phrase[2]. Hence, we try to find binary entailment relationships between queries and candidate sentences with the hypothesis that the answer phrase is likely to be contained in them. Firstly, we convert a question into a regular sentence, which represents our hypothesis sentence to be entailed (hypothesis-*h*) and then use *textual entailment* module to identify if the candidate sentence (text-*t*) entails *h*. Thus, given a (*t*-*h*) pair, we try to recognize the relation between the meaning of the text and hypothesis as a *true entailment* if the meaning of the hypothesis is entailed from the meaning of the text such as given follows:

t: Harry was born in Iowa.
h: Harry's birthplace is Iowa.

t entails *h*, otherwise we recognize the relation between the meaning of the texts as *false entailment*. In this section, we demonstrate experiments conducted on Textual Entailment datasets (freely available from PASCAL recognizing textual entailment (RTE) challenge conference- http://pascallin.ecs.soton.ac.uk/Challenges/RTE/) using the proposed T2FIS method. The goal of RTE challenge is to recognize semantic inference that a textual entailment defines directional relation between two text fragments, called *text* (*T*) and *hypothesis* (*H*) so that a human being can infer that *H* is most likely true on the basis of the contents of *T*. As a further note, we use the entailment model to build a QA system.

Using the RTE datasets, we build a classifier model using proposed T2FIS method. This model is build to be implemented to our Question Answering (QA) system to rank the sentences retrieved from a search engine while matching each retrieved sentence (T) with the question query sentence (H). The question query is transformed into a sentence putting a placeholder to where the answer should be in the question sentence.

---

[1] Textual entailment models are first introduced in Pascal RTE conference (Dagan et.al. 2006).

[2] Answer-extraction is left out as a future research study on natural language processing applications.

Dataset. There are four different RTE challenges so far, each having different sets of T-H pairs. We combined the first three RTE datasets and only used the *T-H* pairs that are specifically designed for QA systems, i.e., there are different sets of pairs constructed for different applications such as summarization, information retrieval, etc. The hypothesis in T-H pairs are formed by converting a question sentence into a regular sentence and placing the question word (what, how, when, etc.) with the correct/false answer as shown in Table 4.

Example Pairs	***Entailment***
T: In February 2002, President George Bush visited China to mark the 30th anniversary of Nixon's historic trip. **H**: Nixon visited China in February 2002.	***FALSE***
T: The Chernobyl nuclear-power plant is in Ukraine, but the reactor that exploded during the night of April 26, 1986, is only 10 miles from the Belarusian border **H**: The Chernobyl disaster took place on the 26th of April, 1986.	***TRUE***
T: Microsoft was established in Italy in 1985. **H**: Microsoft was established in 1985.	***FALSE***

Table 4. Examples of Text-Hypothesis pairs from Recognizing Text Entailment Challenge.

Features: We extract different sets of attributes from the *T-H* pairs and to generate some of these features, we used different tools including Stanford Tagger (Klein and Manning, 2003), Named Entity Tagger (Finkel et.al., 2005), WordNet::Similarity Package (WordNet). Each (T-H) pair is analyzed to extract features (input variables), which depend on the relation between them, some of which is shown as follows:

Lexico-Syntactic Overlap-Alignment Features*:* These features range from the ratio of the consecutive word overlap between the T and H (*n-gram, i.e.,* $n \in \{1,2,3\}$), the lowest common subsequence which measures the similarity between text T with length m and hypothesis H with length n, by searching in-sequence matches that reflect sentence level word order. Other features in this category are skip-*n*gram, number of common pair of words in T and H in order with gaps.

Semantic Features: Noun, verb and adjective/adverb specific semantic overlap metric (similarity measure) using WordNet's hypernym, hyponym, negation match between T-H based on clue phrases such as 'no', 'not', 'neither', etc., which are some of the examples of the features extracted from T-H pairs.

Since the task is text entailment, we extracted two verb match statistics using WordNet's *cause to* and *entailment* relations. For each verb pair that groups a verb from the text v_T and one from the hypothesis v_H we tested either a caused by or entailment relation when;

- *verb entailment:* v_H entailment v_T
- *verb cause:* v_T cause to v_H

To generate separate features for each relation, we counted the number of verb pairs constructed in the above form.

We generate the train and testing datasets using the T-H pairs from RTE challenge and extract features as explained above to form the inputs. The binary output variable having the value '1' indicates "true entailment" and '0' "false entailment", and these are assigned manually (given by the RTE challenge datasets). We extract 29 features using different combinations of the lexico-syntactic and semantic features. We use 1670 T-H pairs for building the learning models--training and 2400 pairs for testing purposes. False and true entailments are evenly distributed. We used 10, 50, 100, 200 number of training vectors as the number of nearest neighbors to build four different T2FIS models, i.e., T2FIS_10, T2FIS_50, T2FIS_100, T2FIS_200, and analyzed the difference in the experiments. For the rest of the benchmark models, we randomly selected 750 training samples five times to build different models and analyzed their average testing performance on the same testing dataset and measured their error margins between five experiments, i.e., standard deviation of the accuracies.

Model Construction. The system model performance is measured with accuracy,

$$accuracy(\%) = \frac{(True\ Positives) + (True\ Negatives)}{number\ of\ data(nd)}$$

Since the classification model outputs are probabilities, different threshold values (to discern between two classes) values are varied to obtain the optimum True Positives (TPs) and True Negatives (TNs) during learning stage of each modeling approach. The threshold values that are identified by the structure identification are used during inference to estimate class labels of testing dataset. The same parameters that are used in the previous experiments is used in this experiments as well with the exception that the algorithms are designed to find classifier functions, e.g., SVM for classification, and FCCM of T2FIS methods are used. The feature extraction, explained above, is implemented as a part of entailment into our QA system using Java and the T2FIS is implemented using Matlab. The average accuracy results from the five repetitions of experiments and the model with the best average accuracy are shown in Table 5.

Model -- accuracy	*Testing Dataset Average Accuracy*	*significance-test between the best T2FIS model and the benchmarks* ($p<0.05$)
T2FIS-10	0.579	
T2FIS-50	0.585	
T2FIS-100	**0.598**	
T2FIS-200	0.582	
ANFIS	0.547	0.001
SVM-LIN	0.568	0.021
SVM-RBF	0.561	0.009
NN	0.550	0.006

Table 5. Accuracy results of the Text Entailment for QA tasks.

Based on the results of this experiment, the best testing accuracy is obtained when the T2FIS is executed for when 100 nearest training vectors are used (T2FIS_100). Compared to the benchmark methods, there is a [5-9%] improvement when the proposed T2FIS_100 is used. Fig. 8 shows accuracies along with their standard deviations across five separate experiments.

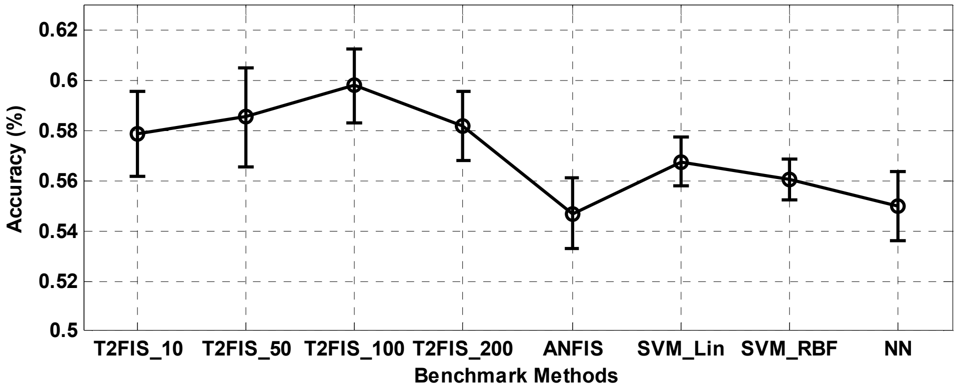


Fig. 8. Accuracy Comparison Graph between the models constructed based on five repetitions. The standard errors are also shown as error-bars.

We also measure the statistical significance of the proposed approach T2FIS on classification problems. The same two-sample left-tailed *t-test* with 95 percent confidence level is used to indicate the significance of the optimum models of each methodology. Our hypothesis is that cross validation errors of the best proposed method (T2FIS_100) and the rest of the models are same with 95% confidence. In Table 5 *significance probabilities* are shown. In all experiments the T2FIS_100 model is significantly better than the benchmark models ($p<0.05$). Thus, we can conclude that the proposed algorithm has comparable/significantly better results than other powerful well-known modeling tools.

The proposed approach helps us to quantify the uncertainty in the membership functions used in the fuzzy system models and variation in model parameters. It is shown that it is possible to capture the variations with a list of discrete membership functions and weighing them individually to incorporate their individual effects to the model. We quantify this uncertainty based on the imprecision in the level of fuzziness parameter of the fuzzy clusters we identify. The real problems can be modeled by using type-2 membership functions which can be derived from the changing values of the fuzziness of the clusters. Hence, when the expert is not present to identify the fuzzy sets, this method could provide better solutions. With the two experiments, we showed that the T2FIS is better compared to the rest of the fuzzy or non-fuzzy reasoning approaches based on the modeling error.

6. Conclusions

Fuzzy logic encompasses conceptual framework of sets and logic that is able to handle both precise and imprecise information and meaning. Although fuzzy systems still do not necessarily outperform human in dealing with uncertainty and imprecision, it helps to reduce the real world problems to a scale that is possible for computing solutions that was impossible before. The principal objective of the presented methods of this paper is to develop applications to enable information extraction under uncertainty, particularly on the conception and design of autonomous systems for natural language processing applications specifically on question and answering systems and textual entailment mechanism. A direct practical application of fuzzy logic to these fields does not seem to exist at present. Thus, higher order fuzzy system models based on fuzzy functions will have many uses in textual and semantic analysis, data mining, and search algorithms in the near future.

In this paper, a new approach to information extraction via fuzzy functions is presented. The presented type-2 fuzzy inference system is used for uncertainty quantification of real-world data. Partitioning a given set of data into granules is most fundamental problem in pattern recognition and data mining. With the presented fuzzy inference system, we define membership functions based on the given dataset and use fuzzy clustering methods. The approach of membership function elicitation of type-2 fuzzy inference system is a category of hybrid knowledge-data class of fuzzy set elicitation and enables employment of different membership functions and local dependency function structures for each cluster. The major benefit of this approach is that, it does not require definition of membership function by an expert. The primary membership functions are found from fuzzy clustering methods presented in the paper and the secondary membership grades are optimized with genetic algorithms. The algorithm adopts simple type-reduction and does not require defuzzification. Textual entailment task is a challenging problem and depends on careful analysis of the features between the question and candidate answer pairs and an efficient classifier model such as uncertainty modeling tool presented in this paper.

7. References

Bandler, R. and Grinder, J. (1996). *The Structure of Magic: A Book about Language and Therapy.* 198 pages, Science and Behaviour Books, ISSN 0831400447.

Babuska, R. and Verbruggen, H.B. (1997). Constructing fuzzy models by product space clustering. In *H. Hellendoorn and D Driankow(Eds.) Fuzzy Model identification: Selected Approaches*, 53-90, Springer, Berlin, Germany, 1997.

Bezdek, J.C., Ehrlich, R. and Full, W. (1984) FCM: Fuzzy C-Means Algorithm. *Computers and Geoscience*, 10, (1984) 191-203.

Bodur, M. and Acan, A., Unveren, A. (2006). Reduction of Generalization error in Fuzzy System Models. *2006 IEEE Int. Conf. on Fuzzy Systems, Vancouver, BC.*, 2184-2189. ISBN 0-7803-9488-7.

Celikyilmaz, A. and Turksen, I.B. (2009a). Increasing Accuracy of Two Class Pattern Recognition with Improved Fuzzy Functions. Expert Systems with Applications, 36, (March 2009) 1337-1354, ISSN 0957-4174.

Celikyilmaz, A. and Turksen, I.B. (2009b). Genetic Fuzzy System Based on Improved Fuzzy Functions. Journal of Computers, 4, 2, (February 2009) 135-146.

Celikyilmaz, A. and Turksen, I.B. (2009c). Spectral Learning with Type-2 Fuzzy Numbers for Question/Answering System. *IFSA World Congress, EUSFLAT Conference, Lisbon, Portugal*, July 2009.

Celikyilmaz, A. and Turksen, I.B. (2009d). Kernel Based Hybrid Fuzzy Clustering for Non-Linear Fuzzy Classifiers, *Proc. 27th Intern. Conf. of the North American Fuzzy Information Processing Society – NAFIPS- IEEE*, (June 2009).

Celikyilmaz, A. and Turksen, I.B. (2008a). Uncertainty modelling with evolutionary improved fuzzy functions approach. *IEEE Systems, Man, and Cybernetics- Part B*, 38, 4, (August 2008) 1098-1110, ISSN 1083-4419.

Celikyilmaz, A. and Turksen, I.B. (2008b). Enhanced Fuzzy System Models with Improved Fuzzy Clustering Algorithm. *IEEE Trans. on Fuzzy Systems*, 16, 3, (June 2008) 779-794, ISSN 1063-6706.

Celikyilmaz, A. and Turksen, I.B. (2008c). Validation Criteria for Enhanced Fuzzy Clustering. *Pattern Recognition Letters*, 29, 2, (January 2008) 97-108, ISSN 0167-8655.

Celikyilmaz, A. and Thint, M. (2008). Semantic Approach to Textual Entailment for Question Answering – A New Domain for Uncertainty Modeling. *7th IEEE Int. Conf. Cognitive Informatics (ICCI'08), Stanford University, CA, IEEE CS Press*, (August 2008) 481-487, ISBN 978-1-4244-2538-9.

Celikyilmaz, A. and Turksen, I.B. (2008d). Type-2 Fuzzy Classifier Ensembles for Text Entailment. *Joint Conference on Information Sciences (JCIS2008)– Fuzzy Theory and Technology Track, Shenzhen, China*, (December 2008) 1-7, ISSN 1951-6851.

Celikyilmaz, A. and Turksen, I.B. (2008e). Uncertainty Bounds of Fuzzy C-Regression Method. *IEEE Int. Conference on Fuzzy Systems - World Congress on Computational Intelligence (WCCI 2008), Hong Kong*, (June 2008) 1193-1198, ISSN 1098-7584.

Celikyilmaz, A. and Turksen, I.B. (2008f). Genetic Type-2 Fuzzy Classifier Functions. *Proc. 26th Intern. Conf. of the North American Fuzzy Information Processing Society – NAFIPS- IEEE*, (May 2008) 1-6, ISBN 978-1-4244-2351-4.

Celikyilmaz, A. and Turksen, I.B. (2008g). A Type-2 Fuzzy C-regression Method. *Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU-2008), Malaga, Spain*, (2008).

Celikyilmaz, A. and Turksen, I.B. (2007). Fuzzy Functions with Support Vector Machines. *Information Sciences*, 177, 23, (December 2007) 5163-5177, 2007.

Chen, J.-Q. and Xi, Y.-G., and Zhang, Z.-J. (1998) A clustering algorithm for fuzzy model identification. *Fuzzy Sets and Systems*, 98, 3, (Sept. 1998) 319-329, ISSN 0165-0114.

Coupland, C. and John, R. (2007) Geometric Type 1 and Type 2 Fuzzy Logic Systems. IEEE Trns. Fuzzy Systems, 15, (February 2007) 3-15.

Dogan, I. and Glickman, O. And Magnini, B. (2006) The PASCAL Recognising Textual Entailment Challenge. *In Quinonero-Candela, J. and Dogan, I., and Magnini, B.*

And D'Alche-Buc, F. (Eds), Machine Learning Challenges. LNCS 3944 (2006) 177-190.

Demirci, M. (1999). Fuzzy functions and their fundamental properties. *Fuzzy Sets and Systems*, 106, 2, (September 1999) 239-246, ISSN 0165-0114.

Emami, M.R. and Turksen, I.B., Goldenberg, A.A. (1998). Development of a Systematic Methodology of Fuzzy Logic Modelling. *IEEE Trans. on Fuzzy Systems*, 63, (August 1998) 346-361, ISSN 1063-6706.

Finkel, J.R. and Grenager, T. And Manning, C. (2005) Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proc. 43nd Annual Meeting of the Assiciation for Computational Linguistics (ACL 2005) (2005) 363-370.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Publishing.

Gunn, S. (1998). Support Vector Machines for Classification and Regression. *ISIS Technical Report*.

Hagras, H. (2004). A Hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. *IEEE Trans. Fuzzy Syst.*, 14, 4, (Aug. 2004) 524-539, ISSN 1063-6706.

Hathaway, R. and Bezdek, J. (1993) Switching regression model and fuzzy clustering. *IEEE Trans. Fuzzy Syst.* 1, 195-204.

Höppner, H. and Klawonn, F. (2003) Improved fuzzy partitions for fuzzy regression models. *Int. Jrnl. of Approximate Reasoning*, 32, (2003) 85-102, ISSN 0888-613X.

Hwang, C. and Rhee, F.C.-H. (2007) Uncertain fuzzy clustering: interval type-2 fuzzy approach to c-means. *IEEE Trans. Fuzzy Syst*. 15, 107-120.

Klein, D., and Manning, C.D. (2003) Fast Exact Inference with a Factored Model for Natural Language Parsing. *In Advances in Neural Information Processing Systems (NIPS 2002)*, 15 (2003) 3-10.

Menard, M. (2001). Fuzzy Clustering and Switching Regression Models Using Ambiguity and Distance Rejects. *Fuzzy Sets and Systems*, 122, (2001) 363-399.

Mendel, J.M. (2001). *Uncertain Rule-Based Fuzzy Logic Systems: Theory and Design*. Prentice, Upper Saddle River, 2001.

Mizumoto, M. (1989). Method of Fuzzy Inference suitable for Fuzzy Control. *J. Soc. Instrument Control Engineering*, 584, (1989) 959-963.

Turksen, I.B. (2008). Fuzzy Functions with LSE. *Applied Soft Computing*, 8,3, (June 2008) 1178-1188, ISSN 1568-4946.

Turksen, I.B. and Celikyilmaz, A. (2006). Comparison of Fuzzy functions with Fuzzy Rule Bases. *Int. Journal of Fuzzy Systems*, 8, 3, (2006) 137-148.

Sasaki, M. (1993) Fuzzy Functions. *Fuzzy Sets and Systems*, 55, 3, (May 1993) 295-301, ISSN 0165-0114.

Takagi, T. and Sugeno, M. (1985). Fuzzy Identification of Systems and Its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics,* 15, 1, (1985) 116-132, 1985.

Uncu, O. and Kilic, K. and Turksen, I.B. (2004). A new fuzzy inference approach based on Mamdani inference using discrete type-2 fuzzy sets. *IEEE Intrn. Conf. Syst.,Man, Cbyern., B, Cybern*., 3, (October 2004) 2272-2277.

WordNet, http://wordnet.princeton.edu/perl/webwn?s=word-you-want
Zadeh, L.A. (1965) Fuzzy Sets. *Inform.& Control,* 8 (1965) 338-353.

9

The Algorithms of the Body Signature Identification

Hnatiuc Mihaela
Constanta Maritime University
Romania

1. Introduction

The behaviour of subjects during the sleep gives information about their healthy. Medical staff has done several studies regarding the body position and its motion on the bed. At present the bed and the mattress are conceived in order to satisfy the comfort needs of the user. The mattresses have position, pressure, temperature and humidity sensors which inform the doctor about the behaviour of the subject on the bed (Agarwal R. & Gotman J., 2004).

The sleep depends by the many parameters as: the type of illness, the psychical state, the medical treatment, the alcohol and caffeine consumption, the external conditions etc. This can be classified into several states using information by the polysomnografic measurement that includes: electroencephalogram (EEG), electromyogram (EMG), electrooculogram (EOG).

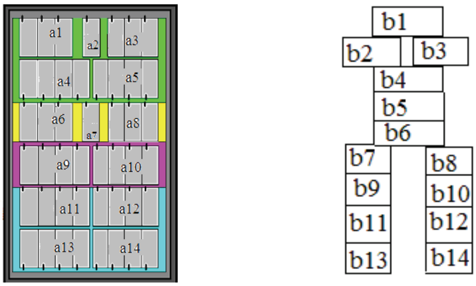
In this paper are presented the methods for the identification of the body position on the bed using the position sensors. The system of sensors has divided in areas of studies recorded in a matrix. The sensors information's are acquired with the dedicated card for this application. The card has a microcontroller core for: acquisition, conversion of analog to digital data and sending the data to the PC (Virone, 2003). The sampling values of the sensors give information to the medical staff about the subject presence on the bed, the movement and position of the body (Watanabe T. & Watanabe K., 2004), (Shochart & Oksenberg, 2003). The localization of the body position depends of the numbers of system areas. The data are represented as a matrix in which the row represents the values of sensors placed on the length of bed and column represents the sensors values placed on the transverse direction. The data are classified using fuzzy and neuro-fuzzy algorithms implemented in a MATLAB program. The aim of work is to develop an expert system which can diagnose the healthy of the subject using the data acquired of the different type of sensors placed on the mattress.

In the following sections are described: the sensors' system, the algorithms used to identify the position of the body and in the last are present the results of the experimental part.

2. The sensors system

The system of sensors for the identification of the body position is conceived by the use of the medical staff. Several methods to position the sensors are tested. The data are processed for two prototypes.

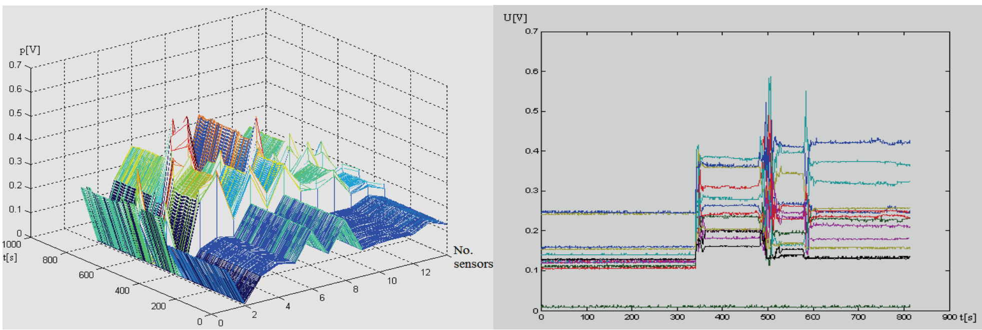
In the first case (Fig. 1 a), the sensors are equally distributed along of the bed and in the second mode (Fig. 1b), the sensors placed on the bed have a body form.



a) b)

Fig. 1. The representation of the zones where are placed the sensors on the bed

The data acquired from the sensors distributed as Fig. 1a, are represented in Fig. 2.
The 3D representation of the sampling signals, acquired during the 1000s period of time, are showed in Fig.2 a. The p[V] of the OZ axis represents the values of 14 sensors. The data are acquired with a subject on the bed. The peaks variations represent the body movement when the subject changes the position.
The 2D plot represents the sensors values during the 900s time period (Fig. 2b). The signals of the first 350s represent initial values, without the subject on the bed. Beginning with the 350s, the subject is on the bed. The peaks represent the body movement.



a) b)

Fig. 2. The signal variations of the sensors with the subject on the bed, 3D representation a) and 2D representation b)

The large values of signals represent the zone of body which presses harder on the bed. Using these results the medical bed can be adjusted according to the body position. The mattress can be build with many layers, to assure the comfort of the subject.
The matrices corresponding to the sensors values for the two models (Fig.1 a, b) are:

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & 0 & a_5 \\ a_6 & a_7 & a_8 \\ a_9 & 0 & a_{10} \\ a_{11} & 0 & a_{12} \\ a_{13} & 0 & a_{14} \end{bmatrix} \tag{1}$$

$$B = \begin{bmatrix} b_1 & 0 \\ b_2 & b_3 \\ b_4 & 0 \\ b_5 & 0 \\ b_6 & 0 \\ b_7 & b_8 \\ b_9 & b_{10} \\ b_{11} & b_{12} \\ b_{13} & b_{14} \end{bmatrix} \tag{2}$$

The elements of matrix A corresponds of the sensors values represented in Fig. 1a. The matrix have the 3 columns and 6 rows. The number of columns corresponds to the maximum number of sensors placed on the length. The null elements indicate the absence of the sensors. Matrix B has 2 columns and 9 rows. Its elements correspond of the sensors value placed according to Fig 1b. The maximum number of sensors placed on the length is 2.
The sampling period of acquisition is 2s.
Using the values acquired from the sensors and a simple algorithm one can know the hour of presence or absence of subject on the bed, the period of time spend in bed, the movement period and the body position (Table 1.).

Data	The hour of body movement	The sensor value variation (maxim)	The part of body movement	The movement period	The daily period	Hour absence on the bed
12.01.2005	12:52	0,572V	All	10s	Aftern oon	13:55
12.01.2005	18:50	1V	All	1h	Night	19:30
12.01.2005	19:40	1V	All	10min	Night	-
12.01.2005	19:40 20:12	0,22	Right leg	5s	Night	-
12.01.2005	19:40 20:50	0,57V	All	5s	Night	20:55
12.01.2005	21:30	1,2V	All	10s	Night	-

Table 1. The table diagram of the subject behaviour on the bed

The sensor value variation (Table 1), represents the maximum variation of the data, between all the sensors, which results after the body movement. The localisation of the body movement is achieved.
The data are analysed during a period of 24 hours and this are divided in the small periods corresponding to the day and the night.
The period of 24 hours is a sum of the absence (T_{abs}) and presence (T_{pres}) periods of the subject on the bed.

$$24h = T_{abs} + T_{pres} \tag{3}$$

The period of rest of the subject on the bed can be passive or active and this results from the movement period and the amplitude.
The system might be integrated in the intelligent apartment which has many sensors for the tracking of the subject (Ross, 2004), (Hirota & Tamaki, 2001), (Hnatiuc & Fontaine, 2006). Using the entire daily activity data one can identify the healthy of the subject taking into account his behaviour.

3. The data analysis algorithms. Short description

3.1 The K-means method. Silhouette parameter

A good cluster gives the qualities classes with similarities between the objects of a class and small similarities between external objects of the class. The quality of classification is measured of the abilities to discover the hidden features (Hans-Hermann, 2008).

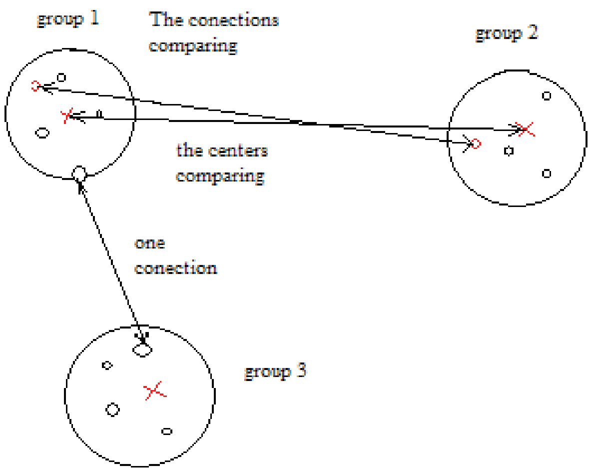


Fig. 3. The methods of data classification using the similarities between objects

The K-means algorithm is based on the relative similarities and defines parameter and probabilities. The idea is to find the K centers, one for each cluster. The centers might be placed at the distance one of the other. The next step is to compute the distances between objects and centers. The objects are associated to each center. The new centers are recomputed as centers resulted of the last step. When it is known the new K centers is applied one more time the algorithm describes above. It is generated a loop which is stopped when the centers don't change their place. The algorithm is achieved with minimization of the function of square error.

The "Silhouette" technique computes the silhouette distance for each sample as average for each cluster and the average of overlapping for all data sets.

The silhouette parameter represents a measure of similarity of a point with the points in the same cluster compared with points of the other cluster.

Let us consider a cluster of K points in K groups; x_i is associated of cluster A; C_k is a cluster different of A. Depending on the average value of the silhouette parameter there is evaluated the number of classes.

$$a(x_i) = \frac{1}{no\{A\} - 1} \sum_{j \in A, j \neq i} d(x_i, x_j) \tag{4},$$

where *a(i)* is the average of not similarly distance of x_i for all points of *A* and *d(i,C)* is the average of not similarly distance of x_i for all points of *C*.

$$d(x_i, C_k) = \frac{1}{no\{C_k\}} \sum_{x_j \in C_k} d(x_i, x_j) \tag{5}$$

One chose the smallest distance between them (6).

$$b(x_i) = \min_{C_k \neq A} \{d(x_i, C_k)\} \tag{6}$$

where *b(i)* is the neighbourhood of x_i and no{C_k} is number of objects included in cluster D.

The average of silhouette value *s(i)* of point x_i is:

$$s(x_i) = (b(x_i) - a(x_i)) * \frac{1}{\max(a(x_i), b(x_i))} \tag{7}$$

The coefficient silhouette interpretation, with respect to its value, according to Rousseeuw, 1987, follows:

for the general range $s(xi) \in [-1,1]$

- $s(x_i) \in [0.71, 1]$ – the points are in the good class, the structure is strong the distance of the other cluster is good;
- $s(x_i) \in [0.51, 0.70]$ – the structure is acceptable;
- $s(x_i) \in [0.26, 0.50]$ – the structure is poor, it might be artificial;
- $s(x_i)=0$ – the point is at the crossing of two class;
- $s(x_i)=-1$ – the points is not classified

This method is used to find the number of classes and their centres used as parameters for the fuzzy or neuro-fuzzy systems.

3.2 The fuzzy classification method

The expert system gives the advice, diagnoses and recommendation inspired from the real world. The organisation of such a system is based on the knowledge of a human expert, but it is difficult to precisely extract the logic which can be implemented. In general, it is developed a prototype based on the experts' information and then the system is tested in order to update the databases (Fukuda&Kubota, 1996). The knowledge base has the rules of type „IF-THEN”. Using a motor of inference are deduced the final solutions (Zemankova-Leech, 1983). These are represented by the fuzzy or neuro-fuzzy systems. (Fig.4.).

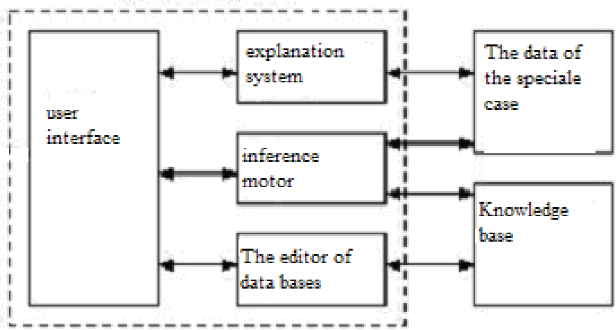


Fig. 4. The block diagram of an expert system

The input data of a system, in many cases, is associated to many classes with a membership degree. This is computed according to a membership function. All these are simulated with a fuzzy system which is divided in premises and consequences parts. The classification systems use the membership function in premises part and singleton in the consequences parts because these might be associated with the centres of classes.

If the numbers of classes and memberships functions of the inputs or outputs are established, one might compute the centre for each class and their limits. These parameters are adjusted in the learning system period.

The expert system has three stages of simulation: learning, testing and checking.

In the learning stage the system might test all the possible cases, using all values acquired in the experimental study. The number of learning stages might be very large excluding the transient period of the beginning. The error between the output resulted after the computing and the targeted output is defined by the user.

Fuzzy Logic. Definition.

If U is any group, one call the Fuzzy Logic, an application $f{:}U\rightarrow[0,1]$ characterised by the membership function $\mu_f{:}U\rightarrow[0,1]$. If, the input x is assigned to U, $\mu_{f(x)}(x)$ is defined as the membership degree.

The fuzzy system algorithm is based on the fuzzy rules. It uses the linguistic variable in inputs and the output. The linguistics expressions describe the relationship between the condition and the consequence, parts of the classification system. In the end, the value resulted after the deffuzification is a crisp value (Zimmermann, 2001) (Zadeh, 1965), (Zadeh, 1968).

The rules “If-Then”, using the simplified fuzzy inference method, have the form:

$$\text{If } x_1 \text{ is } A_{i,1} \text{ and } x_2 \text{ is } A_{i,2} \text{ and… and } x_n \text{ is } A_{i,n}, \text{ then } y \text{ is } w_i$$

where $A_{i,j}$ is a membership function for the j-th input of the i-rule, and w_i is a singleton for the output of the i-th rule.

There are two types of fuzzy systems: Mamdani and Sugeno. The classification system usually uses the Sugeno, which has the following stages:

1. Computing of the membership degrees of $\mu_{Ai,j}(x_j)$ and $\mu_{Ai,j+1}(x_{j+1})$ of the i-th rule ($i=1,\ldots,r$, $j=1,\ldots,n$).
2. Computing of the firing strength of the i-th rule using equation (2):

$$\mu_r = \prod_{j=1}^{n} \mu_{Ai,j}(x_j) \quad or \quad \mu_r = \min_{j=1}^{n}(x_j) \tag{8}$$

3. Computing of the resulting output by weighted average, based on the firing strength, with the equation (7)

$$y(x) = \frac{\sum_{i=1}^{r} \mu_i(x) \cdot w_i}{\sum_{i=1}^{r} \mu_i(x)} \tag{9}$$

Where $y(x)$ is the output after the defuzzyfication, w_i is the singleton of the consequence part.

The fuzzy parameters might be identified with a clustering method.

4. Application of fuzzy system. The identification of body position on the mattress

4.1 The pre-processing data

The body position on the bed is identified using the position sensors placed on the mattress as presented in Fig 1 a) and b) (Alametsa,Varri, et.al. 2004). The system might give a result about the position in the rest period, when the data acquired is constant for a period of time (Hnatiuc & Caranica, 2009).

The data from the analyses are recorded seven patients, women and men, being tested. The sensors system is divided in three zones of study: the head and shoulders, the abdomen and the legs.

A data pre-processing is done to check if the subject is on the bed and if he is completely outstretched. If the subject is not completed outstretched the data processing is stopped.

In the pre-processing stage is computed the maximum and the minimum values and their area of the sensors at an instant time. First identification is done according to these values. If the minimum value is equal with the initial value of the sensors, the subject is not completely outstretched on the bed. These values help to identify the place of the subject on the bed.

It is known that the information about the position is obtained after the subtraction operation between the sensor value with and without subject. The initial value of sensor is between [0, 0.02]V, when the subject is not present in that place. The value is larger then 0.09V when the subject presses the place where the sensor is located. The maximum amplitude of the sensor signal is 5V.

After the first data analysis it results: the presence of the person on the bed, the body position outstretched or not and the legs' position (Table 2). The final analysis is about the abdomen position identification. The algorithm for the position identification is not available during the body movement. The sample value is recorded at 2s time.

Test the person presence on the mattress;
1. *If* the maximum value sensor variation, from all the sensors, is bigger then 0.02V *Then* the person is on the mattress and *go to* 2.
Test the body movement on the mattress;
2. *If* the amplitude between the present value and the previous value of sensor is larger than 0.3V *Then* the body moves. So *go to* 1. *If not*, compute the maximum value of the sensors in the same time and save this value and sensor number (go to 3)
Test the time period between two body movements;
3. Compute the time period between two maximum values showed at stage 2. *If* the time period is bigger *then* 10 minutes, *go to* 4. *If* it is *not* go to 1.
Test the zone of the maximum variation. It is known that the basin of the person weighs the most of the body zone;
4. *If* the maximum value is at the head or legs area, the patient is not completely outstretched on the mattress and the analysis for the position identification is *STOPPED*. *If* this zone is abdomen or shoulders the person is completely outstretched and continue the analyses.

Table 2. The pseudo code of the pre-processing data

The data are processed before to be introduced in the classification system. If the subject is not completed outstretched on the bed, the analysis for the body position identification is not necessary.

After the data preprocessing can be studied the number of classes which can be identified with the sensors' values. Using the K-mean clustering one know which is the maximum number of classes and how are separated.

4.2 Cluster data

The acquisition of data is produced in the presence of a specialist. The body position of the subject is recorded after each experiment, using the specialist indication.

One verified, in the following paragraph, if the number of classes resulted after the cluster algorithm are equal with the numbers of classes recorded in experimental part.

A. The first test is done on the model 1a). The sensors covered, in this case, the entire bed surface. The area of the sensors is composed by three zones: the head and shoulders (1, 2, 3, 4, 5), the abdomen (6, 7, 8, 9, 10) and the legs (11, 12, 13, 14).

The algorithm presented in section 3.1 is applied on the data and the results are presented on the Table 3. In this test, the databases used don't contain the column with the body position class. One uses the Euclidian distance for the computing of distance between data. The silhouette parameter must be in range [0, 1].

The results of K-means algorithm prove as the number of classes is five because of this number the value of the silhouette parameter is maximum (Table 3). Looking on the graphic representation, the silhouette parameter has positives and negatives values (Fig. 5). So the maximum numbers of classes which can identify, using the silhouette values is four. The data are recorded using the sensors system represented in Fig. 1a.

The body classification with the K-means method, using silhouette parameter is equal with the predicted number, after the visual observation. Using the arrangements of sensors as in the first figure, one can identify the body position on the bed.

Number of classes	Silhouette Parameter
2	0.3201
3	0.6772
4	0.6586
5	0.7140
6	0.5775

Table 3. The results of the body position identification using silhouette parameter

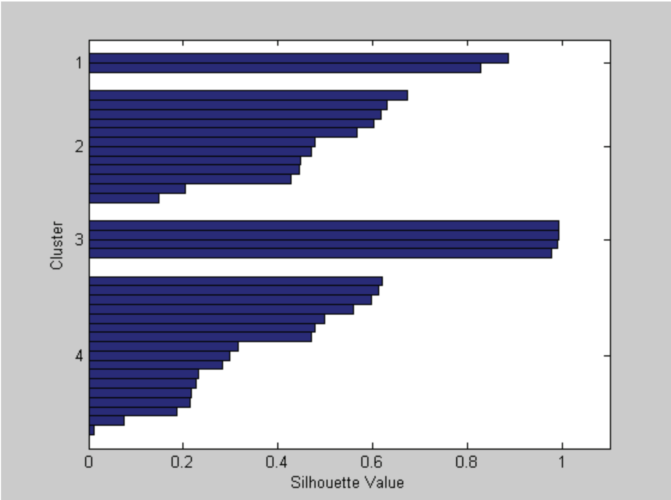


Fig. 5. The silhouette values resulted after cluster method applied of the sensors system represented in Fig.1 a)

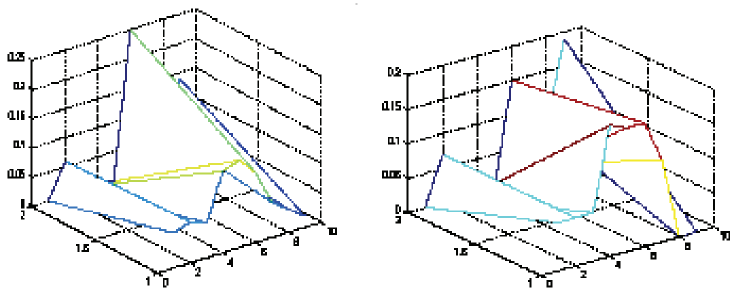
B. The tests of the second prototype are presented in the next paragraph.
In this case the sensors are placed, on the bed, under the most important position of the body zone. For each acquisition is marked the position of body, and there are used 12 values provided by the sensors (Table 4).

b1	**b2**	**B3**	**b4**	**b5**	**b6**	**b7**	**b8**	**b9**	**b10**	**b11**	**b12**	**class**
0.09	0.14	0.09	0.09	0.11	0.33	0.25	0.23	0.08	0.05	0.06	0.07	1
0.09	0.14	0.08	0.09	0.11	0.32	0.24	0.15	0.07	0.05	0.06	0.06	1
0.12	0.11	0.02	0.12	0.15	0.21	0.21	0.19	0.16	0.01	0.02	0.06	2
0.1	0.1	0.08	0.1	0.24	0.2	0.13	0.01	0.14	0.13	0	0.02	3
0.1	0.1	0.08	0.1	0.23	0.19	0.12	0.01	0.13	0.13	0	0.02	3
0.07	0.01	0.18	0.07	0.14	0.21	0.16	0.17	0.02	0.14	0.02	0.1	4
0.06	0.07	0.11	0.06	0.11	0.29	0.34	0.05	0.02	0.07	0.02	0.11	4
0.2	0.1	0.06	0.2	0.14	0.14	0.3	0.26	0.07	0.05	0.08	0.09	1

Table 4. The data acquisition representation

Note: In the 15 column the number represents the body position as following: 1 is the breech down, 2 is the lateral left, 3 is the face down and 4 is the right laterally position.

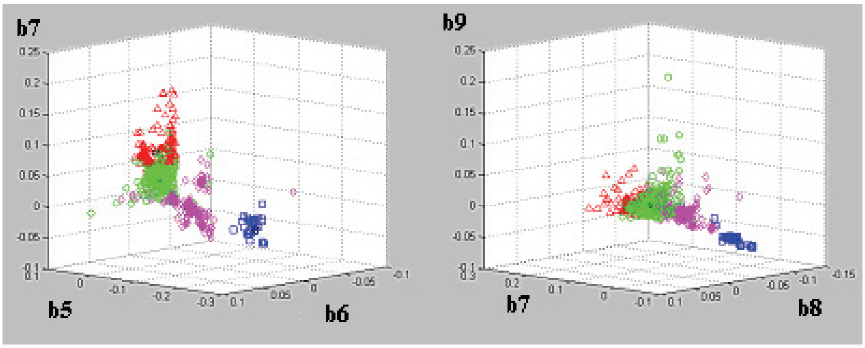
One tries to identify the body position using the visual observation using the data plots. The 3D representations of the matrices B for two position of the body are presented in Fig. 6. The OX axis represents the matrix columns, the OY axis represents the rows of the matrix and on the OZ axis is the sensors values. The Fig. 5a) represent the sensors values for the breech down position. The high values are recorded in the b6, b7, and b8 elements. The second figure (Fig.5b) represents the sample values of the face down position. The high values are recorded for the b1, b2, b3, b11, b12 elements.



a) b)

Fig. 6. The 3D representations of the samples values for two body position breech down a) and face down b)

The maximum density of the sensors values are in the range [0, 0.25] V. There is not very large difference between the two plots. The visual identification is not possible.
Is presented the classification of body position using the data value of the three sensors. The K-means algorithm is applied to the data provided by 6 sensors (b5, b7, b7 and b7, b8, b9) which are included in two groups. The number of classes for body identification, used in that test, is four (Fig. 7).



a) b)

Fig. 7. The cluster identification using the sensors values b5, b7, b7 (a) and b7, b8, b9 (b)

So, the last processing is done using the data of all of the second prototype. It has the sensors assigned to the following areas: the head and shoulders (1, 2, 3) , the abdomen (4, 5, 6, 7, 8) and the legs (9,10, 11, 12, 13, 14).

After cluster computing for 2, 3, 4, 5, 6 classes, the maxim value of silhouette parameter is 0.585 for 5 clusters. The graphics of 5 clusters, similar to the Silhouette parameter, show that the Silhouette average value has negatives values (Fig. 8a). The points of one class are included in the other classes. The representation of the silhouette parameter where there are positive values is for two clusters (Fig. 8b). The average value of the parameter silhouette is smaller then 0.8. The numbers of classes coincides with the classes recorded after visual observation.

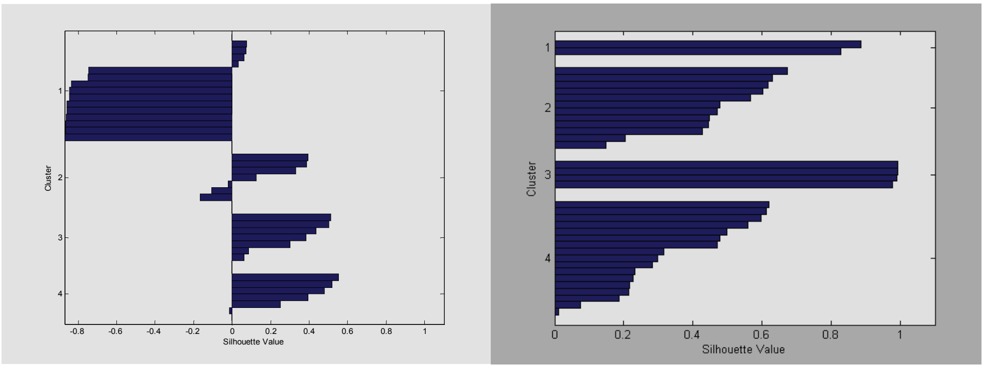


Fig. 8. The Silhouette average representation for 5 clusters a) and 4 clusters b) of body position

The algorithm described above shows the possibilities to identify the position. The centers of classes are too close one to the other and the classes are overlapped.
The classification system might be a fuzzy system in which the sensors' values can be the inputted. The relations between them are expressed in linguistics expressions.

4.3 Data classification

The cluster step offers information about the number of classes and their limits. Using these information a classification system can be created. The data are classified using a fuzzy system with *12 inputs* and *4 outputs*. The inputs represent the sensors values and the output the position of body. The results of cluster part using k-mean show as the maximum number of classes which represent the body positions is four. The outputs of fuzzy system represent the number of classes in which can be grouped the sampling values of sensors.
The fuzzy system proposed is Sugeno, type 0. The input membership functions are triangular type and the outputs are singleton. Each membership input function has three irregular triangles (10) which are overlapped.
The computing algorithm of the fuzzy system uses the set of stages defined in the section 3.2. The learning stage uses the back-propagation method.

$$\mu^T(i,rule)=\begin{cases}\dfrac{(T(i)-a_j(i))}{a_{j+1}(i)-a_j(i)};T(i)>=a_j(i)\\ \dfrac{a_{j+2}(i)-T(i)}{a_{j+2}(i)-a_{j+1}(i)};T(i)>=a_{j+1}(i)\end{cases} \tag{10}$$

where *i* is a sample number, *j* is the input number, *rule* represents the rule number. The μ_T is the membership degree of the time input. The coefficients $a_j(i)$ is the height of triangle, $a_{j+1}(i)$ and $a_{j-1}(i)$ are the values of the two sides of the triangle.
The membership functions for the samples amplitude in the input may be defined as: "Small", "Average", "Large". The membership functions of the outputs have the linguistic descriptions as: "Breech Down" - 1, "Left" - 2, "Right" - 3 and "Face Down" - 4.
The rules are created in function of the pressing force on the sensors areas. The most important areas for position identification are the shoulders and basin. Using the matrices elements of the second prototype one creates the following general rules of the fuzzy system.

Rules No.	**Rules Description**
R1i	***IF*** *the values of the shoulders area are Average* ***AND IF*** *the values of abdomen area are Large* ***AND IF*** *the values of the legs area are Small* ***THAN*** *The position is Face Down*
R2i	***IF*** *the values of the shoulders area are Average* ***AND IF*** *the values of abdomen area are Average* ***AND IF*** *the values of the legs area are Average* ***THAN*** *The position is breech down*
R3i	***IF*** *the values of the shoulders (left side) area are Big* ***AND IF*** *the values of the shoulders (right side) are Small* ***AND IF*** *the values of abdomen (right side) are Big* ***AND IF*** *the values of the abdomen (left side) are Small* ***AND IF*** *the values of the legs area are Small* ***THAN*** *The position is Right*
R4i	***IF*** *the values of the shoulders area (right side) are Big* ***AND IF*** *the values of the shoulders (left side) are Small* ***AND IF*** *the values of abdomen (left side) area are Large* ***AND IF*** *the values of the abdomen (right side) are Small* ***AND IF*** *the values of the legs area are Small* ***THAN*** *The position is left*
R5i	***IF*** *the values of the shoulders area (right side) are Average* ***AND IF*** *the values of the shoulders (left side) are Average* ***AND*** *the values of abdomen (left side) area are Large* ***AND*** *the values of the abdomen (right side) are Small* ***AND*** *the values of the legs area are Small* ***THAN*** *The position is left*
R6i	***IF*** *the values of the shoulders area (right side) are Average* ***AND*** *the values of the shoulders (left side) are Average* ***AND*** *the values of abdomen (left side) area are Small* ***AND*** *the values of the abdomen (right side) are Large* ***AND*** *the values of the legs area are Small* ***THAN*** *The position is right*

Table 5. The general rules of the fuzzy system

Note: $i = \overline{1,r}$ where *r* is the number of particularly rules of generals rules presented above.
The custom of the rule R1i is exemplified in the next table. The elements of the matrix B are used as inputs and the positions as output.
The rules are conceived using the results of the cluster algorithm. The knowledge base and the ranges of the variation might be updated with respect to the stature of the subject.
The fuzzy system is simulated in FIS structure of MATLAB. To perform the input/output map the system map inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs. The parameters associated with the membership functions changes through the learning process. The adjustment of system parameters is facilitated by a gradient vector. This gradient vector provides a measure of how well the fuzzy inference system is modeling

the input/output data for a given set of parameters. When the gradient vector is obtained, any of several optimization routines can be applied in order to adjust the parameters to reduce some error measurements. This measurements error is usually defined by the sum of the squared difference between actual and desired outputs. The back-propagation method can be used to estimate this parameter.

Rules No.	**Rules Description**
R1i (i=1÷4)	**The rules for the breech down position**
R11:	If b1 is Average AND IF the b2 is Average and b3 is Average AND IF b4 is Small AND IF b5 is Small AND IF b6 is Average AND IF b7 is Big AND IF b8 is *Large* AND IF b9 is Average AND IF b10 is Average AND IF b11 is Small AND IF b12 is Small Then the position is breech down
R12	If b1 is *Large* AND IF the b2 is Average and b3 is Average AND IF b4 is Small AND IF b5 is Small AND IF b6 is Average AND IF b7 is *Large* AND IF b8 is *Large* AND IF b9 is Average AND IF b10 is Average AND IF b11 is Small AND IF b12 is Small Then the position is breech down
R13	If b1 is Average AND IF the b2 is *Large* and b3 is *Large* AND IF b4 is Small AND IF b5 is Small AND IF b6 is Average AND IF b7 is *Large* AND IF b8 is *Large* AND IF b9 is Average AND IF b10 is Average AND IF b11 is Small AND IF b12 is Small Then the position is breech down
R14	If b1 is Average AND IF the b2 is Average and b3 is Average AND IF b4 is Small AND IF b5 is Small AND IF b6 is *Large* AND IF b7 is *Large* AND IF b8 is *Large* AND IF b9 is *Large* AND IF b10 is *Large* AND IF b11 is Small AND IF b12 is Small Then the position is breech down. Etc

Table 6. The particularly rules description

A simulation of a structure of the fuzzy Sugeno system for body position identification is presented in Fig. 9.

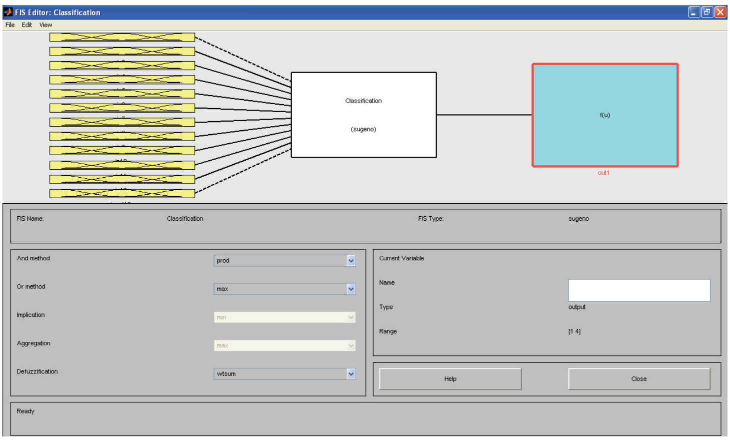
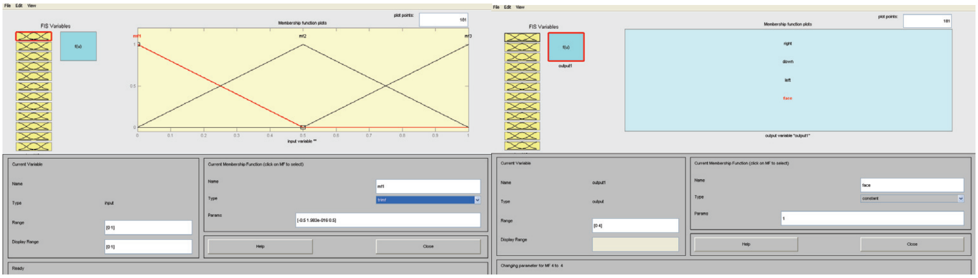


Fig. 9. Fuzzy system structure for body position classification. The system is Sugeno type with 12 inputs and 4 outputs

The input membership function, consisting of three for each input has a triangle form (Fig. 10a) and an output singleton (Fig. 10b).



a) b)

Fig. 10. The inputs and output membership function simulated in MATLAB

The simulation of eight rules, which represent the identification of breech down position (class - 1) has the output value of 0.7 (Fig. 11). The class is identified with 70%precision.

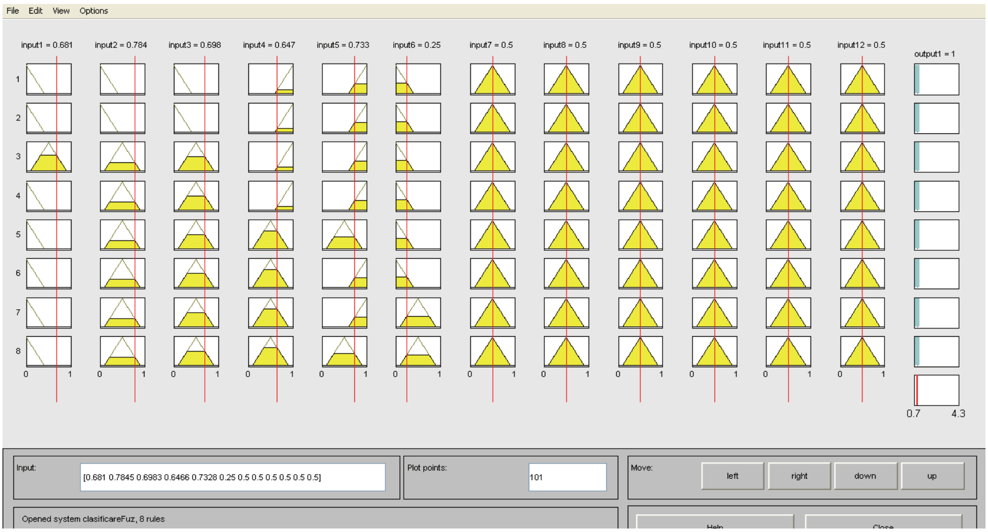


Fig. 11. The rule viewer for the breech down position

The represented fuzzy system is one solution possible for body position identification using sampling values of the sensors.

The fuzzy system with 12 inputs is expensive. Another solution is to use the sensors placed to the shoulders and abdomen area. The number of sensor used is five. A simple simulation using ANFIS Editor of MATLAB is represented in the next figure. The neuro-fuzzy is generated using Grid Partition, and for the learning stage is used the back-propagation algorithm. The system has five inputs and four outputs. The error resulted after the testing stage is 1,2069%. This proved that the body position on the bed can be identified using the sensors placed only in the important area.

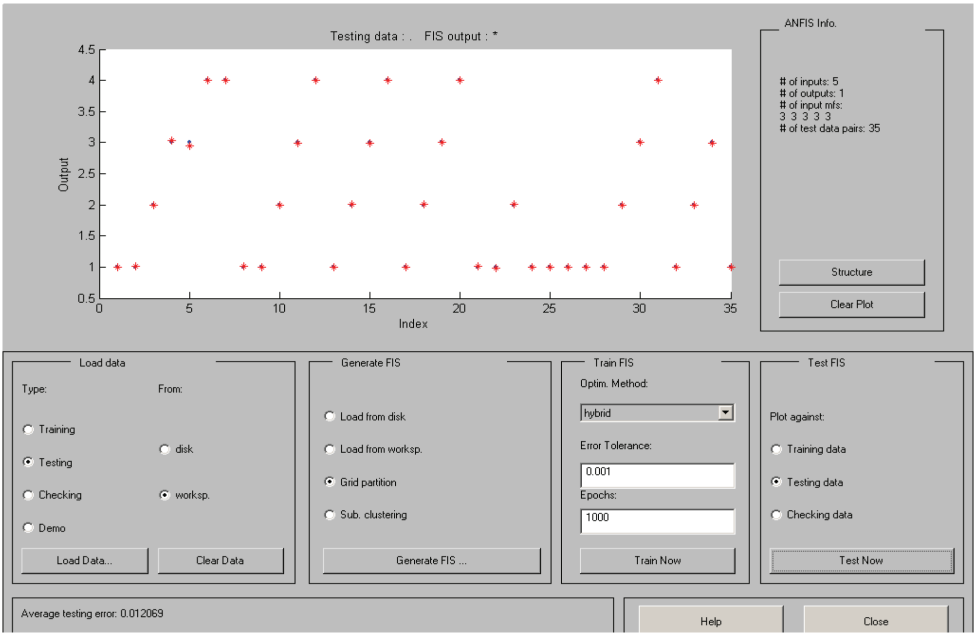


Fig. 12. The results after the testing stage of the fuzzy Sugeno system with five inputs and four outputs

5.Conclusion

Monitoring systems are very important in medical electronics. Their design and analysis facilitate the medical staff work and offer the user independence.

The original system presented in the paper can be used to detect body position on the bed as well as the type of body movement. Using: the body movement, the time period between two successive movement and the sensor amplitude, one can identify the sleep type (normal, agitate, abnormal, convulsive, etc). The system can be adapted to the person and does not depend on their weight, size or position. The classes of the body positions must be established after a large data base calibration for several patients.

The fuzzy system for classification can be selected with respect to the human's bodies and habitudes which are not standard; each of them has their particularity which can be linguistic defined. Each person has particular sleep habitudes and their preference body position during the rest period.

The sensors system offers the **subject "signature"**. It can replace the continuum monitoring of the patient and alarms the medical personal in case of problems.

6. Acknowledgment

Most of the concepts presented in the paper are the result of the research work done in the framework of the ID1223 project, under the supervision of the National University Research Council (CNCSIS), Romania(Oanta&Panait, 2007).

7. References

Agarwal R.&Gotman J. (2004). Computer-Assisted Sleep Staging, IEEE Transaction in Biomedical Engineering. Vol. 48, no. 12, December, 2004, pp. 1412-1423

Alametsa J., Varri A., Koivuluoma M., Barna L., (2004). The potential of EMFi Sensors in heart activity Monitoring, 2nd OpenECG Workshop - Integration of the ECG into the HER&Interoperability of ECG Device Systems. 1-3, April, 2004.

Fukuda, Kubota (1996). Learning, adaptation, and evolution for fuzzy controller. Fuzzy control methodology. 1996, pp.17-31

Hans-Hermann B., (2008). Origins and extensions of the k-means algorithm in cluster Analysis. Journal Electronique d'Histoire des Probabilités et de la Statistique Electronic Journal for History of Probability and Statistics. Vol 4, no2; écembre/December 2008

Hirota A., Tamaki A, Nittono H., Hayashi M., Hori T., (2001). Volitional Lifestyle and Nocturnal Sleep in the Healthy Elderly, Sleep research online, 2001, pp.91-96

Hnatiuc M.&Fontaine J. G. (2006). Human behavior analysis and classification using an intelligent mattress. *The Scientific Memories Sections of the Romanian Academy*. Vol. XXVIII, no. IV, Iasi, 2006, ISSN 1210-3084, pp.48-56, Iasi

Hnatiuc M., Caranica A., (2009). Communication between the Sensor Levels for Monitoring Subjects with Disabilities, Proceedings AT-EQUAL, 22-26 july 2009, pp 87-91 ISBN 978-0-7695-3753-5, Iasi, Romania

Oanta E., Odagescu I., TamasI., (2006). An Original Software for the Decision *Making Process. ICBE2006–International Conference on Business Excellence*. October, 2006, ISBN (10) 973-594-847-8, ISBN (13) 978-973-594-847-4, pag. 296-301, Brasov, Romania

Oanta, E.; Panait, C.; Nicolescu, Hnatiuc M.; B.; Dinu, S.; Pescaru, A.; Nita, A.; Gavrila, G., (2007). Computer Aided Advanced Studies in Applied Elasticity from an Interdisciplinary Perspective, Progress reports of the ID1223 Scientific Research Project, National University Research Council (CNCSIS), Romania, 2007-2010

Ross P.E., (2004). Managing care through the air. *IEEE Spectrum*. December. 2004, pp. 14-19

Shochart T; Oksenberg A., Hadas N., Molotsky A.&Lavie P. (2003). The KickStrip: A Novel Testing Device for Periodic Limb Movement Disorder. *Sleep review*. Vol. 26, no. 4, 2003, pp. 480-484.

Virone G., (2003). Architecture et Simulation Locales du Systeme d'Information Domotique-Sante Intergre a Domicile (SID2) pour detection de situation a Risque et l'aide a la Decicion. Thèse Doctorat. 2003, http://www. Grenoble-univ-rech.org

Watanabe T.&Watanabe K. (2004). Noncontact Method for Sleep Stage Estimation. *IEEE Transaction on Biomedical Engineering*. Vol. 51, no. 10, October, 2004, pp. 1735-1748.

Zadeh, L.A. (1965). Fuzzy sets. Information and Control 8 (3): 338-353. doi:10.1016/S0019-9958(65)90241-X. ISSN 0019-9958.

Zadeh, L.A. (1968). Fuzzy algorithms. Information and Control 12 (2): 94–102. doi:10.1016/S0019-9958(68)90211-8. ISSN 0019-9958.

Zemankova-Leech, M. (1983). Fuzzy Relational Data Bases. Ph. D. Dissertation. Florida State University.

Zimmermann, H. (2001). Fuzzy set theory and its applications. Boston: Kluwer Academic Publishers. ISBN 0-7923-7435-5, 2001, Boston,

Academic Publishers. ISBN 0-7923-7435-5, 2001, Boston.

10

Students' Evaluation based on Fuzzy Sets Theory

Eduardo André Mossin, Rodrigo Palucci Pantoni and Dennis Brandão
University of São Paulo –USP
Brazil

1. Introduction

The traditional students' evaluation method is a process designed to evaluate the qualitative aspects but, in fact, its final result is a grade that values the quantitative aspect. The challenge of this work is to consider a modifying, dynamic and coherent method of evaluation in the industrial automation area.

Especially in the academic area, online laboratories are a commonly used distance learning model where the student practices the theoretical knowledge using a fieldbus simulator. These practical sessions can be performed in any location, as long as an authenticated, secure access is provided.

At the Sao Carlos Engineering School, University of Sao Paulo (EESC/USP), a FOUNDATION Fieldbus™ simulator named FBSIMU (Function Block Simulator) is being used as on online laboratory (Mossin, 2007). Engineering students can get involved with automation system technologies, as well as control theory, by implementing an application, therefore being able to explore and practice their knowledge.

Considering the training and learning environment, it is important to implement a robust evaluation system to help teachers in the industrial automation area identifying students' strong and weak points. The traditional evaluation method is a process designed to evaluate the qualitative aspects but, in fact, its final result is a grade that values the quantitative aspect (Rosa, 1999). Our challenge is to consider a modifying, dynamic and coherent method of evaluation.

A graduated student in Automation must know the technologies related to programmable controllers, computer networks and manufacture automation. Besides these technologies, the student must also know how to configure and specify industrial communication systems, supervise instrumentation and control projects, identify control strategies in industrial processes, tune and optimize control loops in industrial processes, etc.

Furthermore, there is a large amount of industrial protocols (FOUNDATION Fieldbus™ HSE, FOUNDATION Fieldbus™ H1, HART, PROFIBUS, DeviceNet, among others) available in the market and each one of those requires different particularities from the student.

If we look at the traditional teaching method, even an excellent student in a certain area could fail the course and miss great opportunities for not having the formal education the teacher was ideally expecting.

Regarding the inadequacy of the traditional method, this paper proposes the use of the fuzzy set technique that will be applied in the evaluation process of the industrial automation systems learning area, aiming to lessen the evaluation complexity and ambiguity in this area. It is also important to emphasize that this fuzzy learning evaluation methodology may be methodology may be used when training industrial plant operators and engineers who have already been working in the area but must be trained in new, emerging technologies.

Fuzzy sets are sets whose elements have degrees of membership. Fuzzy sets have been introduced as an extension of the classical notion of set. In classical set theory, the membership of elements in a set is assessed in binary terms according to a bivalent condition - an element either belongs or does not belong to the set. By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set; this is described with the aid of a membership function valued in the real unit interval [0, 1]. Fuzzy sets generalize classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1.

The next two sections will present in details the description and the solution for the problem, respectively. Section 4 details the evaluation methodology proposed in this research and section 5 outlines its results. Finally, section 6 delineates the conclusions related to the applications of this study.

2. Problem description

Industrial automation is a broad area and it would not be fair to grade a student simply as approved or not approved. This type of classification provides no information whether the approved student has capabilities in every area. The same misjudgement can occur with a student who failed the course. The student may be extremely good in one particular area but he may fail in other aspects.

Note that the traditional evaluation, besides being not fair in some aspects, makes it difficult to define a student’s profile.

Considering the traditional teaching method, an excellent student in a certain area could fail the course and miss great opportunities for not having an ideal formal education. Using fuzzy sets as an evaluation method, the student would have a report indicating the excellence in a specific area (for example, identifying control strategies). The same report could also inform that the student did not succeed in another area, for example, in programmable controllers. If an organization is looking for a specialist in control strategies, this student could be fairly indicated and chosen for this position. Using the traditional evaluation method, this student could have missed the position.

It is possible then to determine, in a clearer way, the capabilities and the deficiencies of a student in different areas of knowledge in industrial automation using an evaluation system based on fuzzy sets.

Some works have been developed related to evaluation systems based on Fuzzy sets. Some authors presented a method based on fuzzy rules to assign grades to students (Chen & Lee, 1999), (Cheng & Yang, 1998) and (Bai & Chen, 2006). Chang & Sun (1993) presented a method to measure the performance of the students in Junior High School. Another author (Chiang & Lin, 1994) presented a method to evaluate teachers. Another work (Law, 1996) presented a method using fuzzy numbers to assign grades to the students. To finish, another author (Weon & Kim, 2001) created a system that considers the difficulty, the importance and the complexity of each question before determining a final grade.

Some of the ideas implemented by those authors were combined to define the solution for the problem, as described in the next section.

3. Problem solution

As described in section 2, an evaluation resulting only in a final grade can be unfair if not considering the many specializations of a specific sector. Moreover, such method intends to evaluate qualitative aspects but, in fact, the final result values the quantitative aspect.

In this context, this article proposes an evaluation method based on fuzzy sets, available in the FBSIMU simulator, used at the EESC\USP for teaching the FOUNDATION Fieldbus™ protocol.

It is proposed divides the students' evaluation in three distinct industrial automation areas: FOUNDATION Fieldbus™ industrial protocol concepts (FFP); control strategy configuration (CSC); and industrial plant simulation (IPS).

The proposed fuzzy system consists of three inputs and one output as showed in the Figure 1. Inputs are the automation areas and the output is related to the final student evaluation.

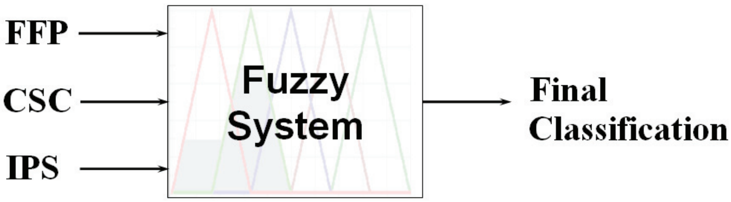


Fig. 1. Proposed Fuzzy System

The general view of the proposed methodology follows these steps: (1) apply a list of questions with items to evaluate the students; (2) calculate the grade from the evaluation for each of the three areas based on Table 1; (3) classify the student as Insufficient, Average or Excellent in the three areas, based on the grades given in step 2; (4) assign the final result to the student based on the result from step 3 and the fuzzy rules from Table 2 ("Fail", "Pass with Many Restrictions", "Pass with Few Restrictions" and "Pass"). The next section will detail the steps described above.

4. Methodology applied for the evaluation

The evaluation methodology will be exposed in three steps. The first step (sub-section 4.1) is calculating the student grade in each area considered in this research. After that (sub-section 4.2), the student classification in each area, using fuzzy theory, will be explained. To finish the process, the sub-section 4.3 explains the method to obtain final classification of a given student.

4.1 Calculating the grade

Each question in the evaluation may required from the student specific knowledge in more than one of the three areas considered in this research (FFP, CSC, and IPS) and, therefore, each area will have a specific weight for each question. For example, one question shows some FOUNDATION Fieldbus™ control strategies for a specific process. To answer this question, the student must know the FOUNDATION Fieldbus™ protocol theory and, moreover, the details on how to create a control strategy. The Table 1 shows the weight of each question in each area. Considered the information related to question one. For this question, the area FFP has weight of 0.90. The area CSC has weight 0.95. The question weight for the area IPS is zero.

Question number	Area	Weight
Q1	FFP	0.50
	CSC	0.50
	IPS	0.00
Q2	FFP	0.20
	CSC	0.20
	IPS	0.60
Q3	FFP	0.30
	CSC	0.10
	IPS	0.60
Q4	FFP	0.20
	CSC	0.45
	IPS	0.35
Q5	FFP	0.75
	CSC	0.15
	IPS	0.10
Q6	FFP	0.25
	CSC	0.55
	IPS	0.20
Q7	FFP	0.10
	CSC	0.10
	IPS	0.80
Q8	FFP	0.10
	CSC	0.80
	IPS	0.10
Q9	FFP	0.35
	CSC	0.15
	IPS	0.50
Q10	FFP	0.00
	CSC	0.10
	IPS	0.90

Table 1. Weight for each question.

Having the weight value for each area and for each question, we then calculate the grade of a specific student for a particular area using Formula (1).

$$\text{Grade}\ (\text{Area}) = \frac{V1 * PQ1(\text{Area}) + V2 * PQ2(\text{Area}) + \ldots + Vn * PQn\ (\text{Area})}{PQ1(\text{Area}) + PQ2(\text{Area}) + \ldots + PQn(\text{Area})} \quad (1)$$

In Formula (1), the field "Area" represents FFP, CSC or IPS. The value of the variable Vn is "1" for the right answers and "0" for the wrong answers. Finally, PQ1, PQ2 … PQn, represent the weight that a particular area has over a specific question (Table 1). To better understand Formula (1), suppose that a student got the questions 1, 2 and 3 right, but the others, in a total of ten questions, were wrong. So, the value Vn for these correct answers is "1" and for the wrong answers is "0". The weigh values in the area FFP for the three "correct" questions would be 0.5, 0.3 and 0.2. Therefore, the calculation would be:

$$\text{Grade}(\text{FFT}) = = \frac{1 * 0{,}5 + 1 * 0{,}2 + 1 * 0{,}3}{0{,}5 + 0{,}2 + 0{,}3 + 0.2 + 0.75 + 0.25 + 0.10 + 0.10 + 0.35 + 0.00} = 0.363$$

4.2 Classifying the student

The next step in this work is classifying the student according to the three proposed control and automation areas. At this point, a fuzzy system is used to classify the student based on the grade obtained in the step described in sub-section 4.1.

Figure 2, 3 and 4 shows the fuzzy pertinence function of each research area (FFP, CSC and IPS). The universe of discourse is the numeric value of the grade (from 0.0 to 1.0) in one of the three areas discussed in this article.

It is important to notice that each area has different pertinence functions. This difference is related to the contribution level of each area for the students'. For example, a student who has a good knowledge in FOUNDATION industrial Fieldbus™ concepts (FFP) and an average knowledge in the industrial area plant simulation (IPS) than other student with a little knowledge in FFP area and an excellent knowledge in IPS area. Comparing figure 2 with figure 4, it is evident that the pertinence function IN (Insufficient) of the FFP area has the universe of discourse from 0.0 to 0.6 whereas in the IPS area the same function has the universe of discourse from 0.0 to 0.3. This means that the minimum grade required to the student to be considered an Average student in the FFP area is bigger than the grade required to the same student to be considered an Average student in the IPS area. This occurs because the IPS area to the students' formation.

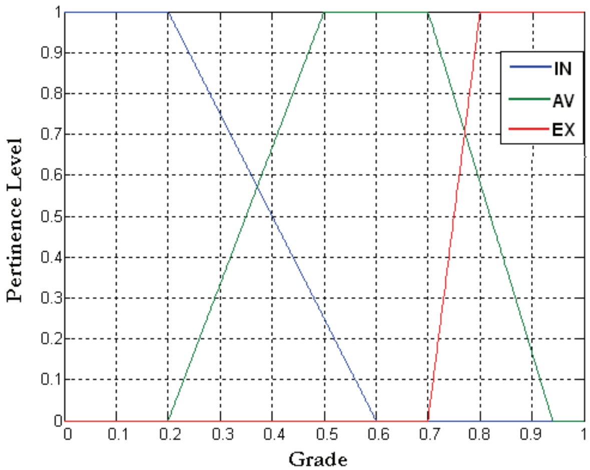


Fig. 2. Pertinence function for the student's final grades in a FFP area

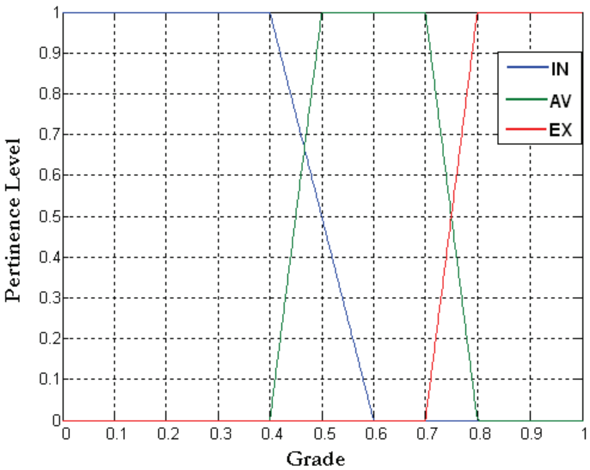


Fig. 3. Pertinence function for the student's final grades in a CSC area

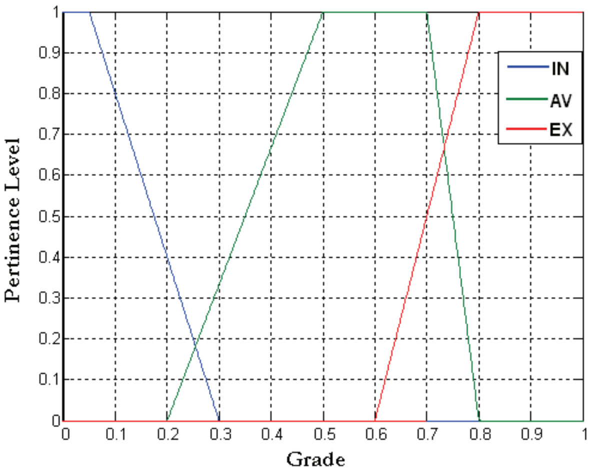


Fig. 4. Pertinence function for the student's final grades in an IPS area

Summarizing, it is very significant to analyse the importance of each area in the proposed methodology and depending on this importance level, the pertinence functions should be adjusted.

These figures also show that the output of this set will be the student's classification: Insufficient (IN), Average (AV) and Excellent (EX). Comparing the previous example where the student's grade in the area FFP is 0.36 to the pertinence function in Figure 2, the student's classification is AV with pertinence level 0.54 and IN with pertinence level 0.58.

This classification is calculated for the three evaluation areas. After obtaining the student's classification in the three areas, it is necessary to finalize the evaluation procedure and assign the general classification for the student. This step is described in the next sub-section.

4.3 Assigning the final classification to the student

The fuzzy system is used for the student's final classification. The inputs for this fuzzy system are the student's classification in each of the three areas (sub-section 4.2) and the output will be: Fail (FA), Pass with Many Restrictions (MR), Pass with Few Restrictions (FR) or Pass (PA). Figure 5 shows the pertinence function that represents this last set.
To determine whether the student will Fail (FA), Pass with Many Restrictions (MR), Pass with Few Restrictions (FR) or Pass (PA), it is necessary to find the rules active (Table 2) .

		IPS Classification		
		IN	**AV**	**EX**
Classification FFP/CSC	**IN/IN**	FA	MR	MR
	IN/AV	MR	FR	FR
	IN/EX	MR	FR	FR
	AV/IN	MR	FR	FR
	AV/AV	FR	FR	PA
	AV/EX	FR	PA	PA
	EX/IN	MR	FR	FR
	EX/AV	FR	PA	PA
	EX/EX	FR	PA	PA

Table 2. Summary of the fuzzy rules.

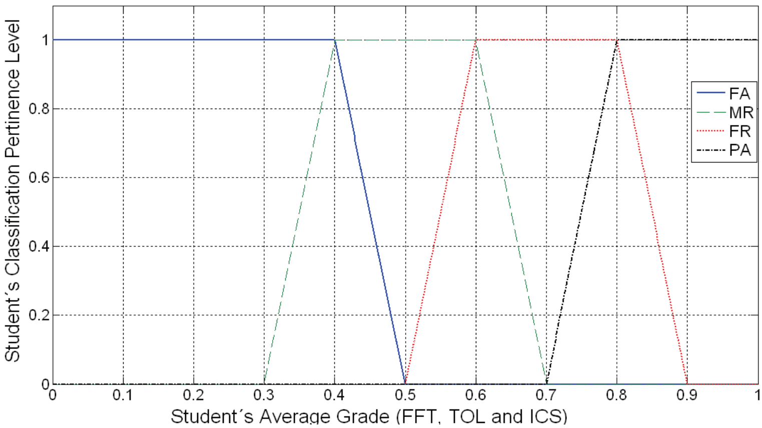


Fig. 5. Pertinence function for the student's general classification: Fail (FA), Pass with Many Restrictions (MR), Pass with Few Restrictions (FR) and Pass (PA).

The generalized *Modus-Ponens* inference process was used in the Fuzzy calculations, in addition to the *Max-Min* composition operator, the *Mandani* implication operator, and the *Maximum* operator for aggregation of the three fuzzy system inputs. The defuzzification of the output "final classification" used the *Center of Area* technique in this work because the resulting values were more appropriate when compared to the other evaluated techniques (*Mean of Maximum* and *First of Maximum*).
The goal of this system is providing a different way to classify the student. Instead of assigning a final grade and simply consider that the student was approved or not in the

evaluation, it is possible to create a report with specific information, whether the student is very good in FFP, insufficient in CSC and insufficient in IPS, for example.
The rules indicated in Table 2 are used for classifying the student. The fuzzy rules are similar to:

If (FFP is Insufficient) and (CSC is Insufficient) and (IPS is Excellent) then the student will Pass with Many Restrictions.

Consider the following example of an output from the evaluation system: "*The classification of the student Luis Carlos Silva is Pass with Few Restrictions, because he had an Average performance in FOUNDATION Fieldbus™ industrial protocol theory, an Insufficient performance in tuning and optimization of industrial process control loops, and an Excellent performance in identifying control strategies in industrial processes.*"

5. Results

The results were obtained from real questions used for evaluating the students at the EESC/USP. Three situations were considered:

- Student gets questions 1, 3, 5, 6 and 8 correct;
- Student gets questions 1, 4, 7, 8 and 9 correct;
- Student gets questions 2, 4 and 9 correct;

It is necessary to obtain the values PQn to apply Formula (1), where n is the number of the question (from 1 to 10).
Each of the following sub-items reports the results from the cases considered above.

5.1 First case (questions 1, 3, 5, 6 and 8)

Considering that the questions 1, 3, 5, 6 and 8 are correct and applying Formula (1), the value for V1, V3, V5, V6 and V8 is 1, and the value for the other questions is 0.
Therefore, using the values shown in Table 1 as a reference, we have:

- Grade in area FFP: 0.69;
- Grade in area CSC: 0.67;
- Grade in area IPS: 0.43.

Having the values from FFP, CSC and IPS, we define the first part of the student's classification: the student is Average in FFP, CSC and IPS areas.
Using the COA method, the CRISP value is obtained. Then, the final classification is applied and the student's classification is "Pass with Few Restrictions."
Bringing together the classification by area and the general classification, the current student's classification will be: "*Pass with Few Restrictions" because the performance was Average in FOUNDATION Fieldbus™ industrial protocol theory, Average in tuning and optimization of industrial process control loops, and Average in identifying control strategies in industrial processes*.
If the traditional analysis methodology were used, the student's grade would be 5 and this student would be approved in the course, with no information regarding in which area this student has a higher or lower potential. As mentioned previously in this article, a student whose grade is 5, being evaluated in the traditional method and with no knowledge in FFP, might be hired by a company instead of another student whose grade is also 5 but with a better performance in the area FFP. Using fuzzy, it is possible to identify the student's potentialities and apply a fair evaluation process.

5.2 Second case (questions 2, 3, 4, 7, and 8)

Considering that the questions 2, 3, 4, 7, and 8 are correct and applying Formula (1), the value for V2, V3, V4, V7 and V8 is 1, and the value for the other questions 0.
Therefore, using the values shown in Table 1 as a reference, we have:

- Grade in area FFP: 0.32;
- Grade in area CSC: 0.53;
- Grade in area IPS: 0.59.

Having the values from FFP, CSC and IPS, we define the first part of the student's final classification: the student is Insufficient in area FFP and Average in area CSC and IPS areas.
The fuzzy output area is obtained after executing the algorithms described previously.
Using the COA method, the CRISP value is obtained. Then, the final classification is applied and the student's classification is "Pass".
Linking the classification by area and the general classification, the current student's classification will be: *"Pass" because the performance was Insufficient in FOUNDATION Fieldbus™ industrial protocol theory, Average in tuning and optimization of industrial process control loops, and Average in identifying control strategies in industrial processes*.
Analyzing this result, we conclude that the student, who would be approved with the traditional evaluation method, is approved with one restriction because the performance in the area FFP was Insufficient.

5.3 Third case (questions 3, 7, 9 and 10)

Considering that the questions 3, 7, 9 and 10 are correct and applying Formula (1), the value for V3, V7, V9 and V10 is 1, and the value for the other questions is 0.
Therefore, using the values shown in Table 1 as a reference, we have:

- Grade in area FFP: 0.27;
- Grade in area CSC: 0.14;
- Grade in area IPS: 0.67.

Having the values from FFP, CSC and IPS, we define the first part of the student's final classification: the student is Insufficient in area FFP, Insufficient in area CSC, and Average in area IPS.
The resulting fuzzy area is obtained after executing the algorithms described previously.
Using the COA method, the CRISP value is obtained. Then, the final classification is applied and the student's classification is "Pass with Many Restrictions".
Bringing together the classification by area and the general classification, this student's classification will be: *"Pass with Many Restrictions" because the performance was Insufficient in FOUNDATION Fieldbus™ industrial protocol theory, Insufficient in tuning and optimization of industrial process control loops, and Average in identifying control strategies in industrial processes.*
In this case, the student would not be approved in the traditional method. Using the fuzzy system described in this article, the student's classification was "Pass with Many Restrictions." The student would receive a report from the course and it would be recommended for this student to take a new learning stage based on the indicated deficiencies.

6. Conclusion

The evaluation system implemented in this work does not aim to disapprove the students but to orientate them during the complete learning process, indicating potentialities and deficiencies.

This method essentially analyzes the student in all aspects. Another important fact in this method is that the qualitative aspects predominate over the quantitative aspects.

A student who was not approved in the traditional method may be approved in the fuzzy method but oriented to take extra classes in specific areas of deficiency, as well as be stimulated in areas where the student has greater knowledge and capabilities. Conversely, if the student was not approved in the diffuse method (fuzzy) because of a great deficiency on the three areas of knowledge, he or she will not be approved in the traditional method either.

It is important to remember that this work can be explored to expand the rules and incorporate other inputs and outputs to the evaluation. Besides, the same methodology can be applied in different areas, others than the control area.

Many times in the traditional evaluation process, teachers use the grade as a repressing mechanism, in an authoritative manner, causing the student to be concerned about calculating the grade instead of learning. With a more detailed analysis, the student’s focus may change, increasing the learning curve in the desired areas.

7. References

Mossin, E. A., Passarini, L .C., Brandão, D. (2007). Networked control systems distance learning: state of art, tendencies and a new fieldbus remote laboratory proposal. 2007 IEEE International Symposium on Industrial Electronics. Vigo, Spain.

Rosa, A., Gariba, M. J., Casagrande, J. B., Santos, C. S., Schneider, M. C. K., (1999). Superando a avaliação da aprendizagem tradicional, através da lógica de conjuntos fuzzy. Revista Scripta, Florianópolis, n. 2, p. 37-47, 1999.

Chen, M. S., Lee, C. H., (1999). New methods for students' evaluation using fuzzy. Fuzzy Sets and Systems, Ed. Elsevier North-Holland, Inc., Amsterdam. V104, issue. 2, p. 209-218, 1999.

Cheng, C. H., & Yang, K. L. (1998). Using fuzzy sets in education grading system. Journal of Chinese Fuzzy Systems Association, 4(2), 81–89.

Bai, S. M., & Chen, S. M. (2006). Automatically constructing grade membership functions for students’ evaluation for fuzzy grading systems. In Proceedings of the 2006 world automation congress, Budapest, Hungary.

Chang, D. F., & Sun, C. M. (1993). Fuzzy assessment of learning performance of junior high school students. In Proceedings of the 1993 first national symposium on fuzzy theory and applications, Hsinchu, Taiwan, Republic of China (pp. 1-10).

Chiang, T. T., & Lin, C. M. (1994). Application of fuzzy theory to teaching assessment. In Proceedings of the 1994 second national conference on fuzzy theory and applications, Taipei, Taiwan, Republic of China (pp. 92–97).

Law, C. K. (1996). Using fuzzy numbers in education grading system. Fuzzy Sets and Systems, 83(3), 311–323.

Weon, S., & Kim, J. (2001). Learning achievement evaluation strategy using fuzzy membership function. In Proceedings of the 31st ASEE/IEEE frontiers in education conference, Reno, NV (Vol. 1, pp. 19–24).

11

Combination of Particle Swarm and Ant Colony Optimization Algorithms for Fuzzy Systems Design

Chia-Feng Juang
Department of Electrical Engineering, National Chung-Hsing University
Taiwan, R.O.C.

1. Introduction

Fuzzy systems (FSs) have been extensively applied to automatic control, pattern recognition, and decision analysis. However, a common bottleneck is encountered in the derivation of fuzzy rules, which is often difficult, time consuming, and relies on expert knowledge. To automate the design of FSs, many metaheuristic learning algorithms have been proposed. One major optimization category uses Swarm Intelligence (SI) model (Kennedy et al., 2001). The SI technique studies collective behavior in decentralized systems. Its development was based on mimicking the social behavior of animals or insects in an effort to find the optima in the problem space. SI models are initialized with a population of random solutions. One well-known SI model is particle swarm optimization (PSO) (Kennedy & Eberhart, 1995). Many modified PSO models have been proposed and successfully applied to different optimization problems (Clerc & Kennedy, 2002; Bergh & Engelbrecht, 2004; Ratnaweera et al., 2004; Juang, 2004; Kennedy & Mendes, 2006; Parrott & Li, 2006; Chen & Li, 2007). FS design using PSO has also been proposed in several studies (Juang, 2004; Chatterjee et al., 2005; Juang et al., 2007; Araujo & Coelho, 2008; Sharma et al., 2009).
Another well-known SI is ant colony optimization (ACO) (Dorigo & Stutzle, 2004). The ACO technique is inspired by real ant colony observations. It is a multi-agent approach that was originally proposed to solve difficult discrete combinatorial optimization problems, such as the traveling salesman problem (TSP) (Dorigo et al., 1996; Dorigo & Gambardella, 1997). In the original ACO meta-heuristic, artificial ant colonies cooperate to find good solutions for difficult discrete optimization problems. Different ACO models have been applied to FS design problems (Cassillas et al., 2000; Cassillas et al., 2005; Mucientes & Casillas; 2007; Juang & Lo, 2007; Juang et al., 2008; Juang & Lu; 2009). In (Cassillas et al.,2000; Mucientes & Casillas; 2007; Juang et al., 2008; Juang & Lu; 2009), the FS input space was partitioned in grid type with antecedent part parameters of an FS manually assigned in advance. In (Juang & Lo, 2007), the FS input space was flexibly partitioned using a fuzzy clustering-like algorithm in order to reduce the total number of rules. For all of these studies, the consequent part parameters were optimized in discrete space using ACO. Since only the consequent part parameters are optimized, and the optimization space is restricted to be discrete, the designed FSs are unsuitable for problems where high accuracy is a major concern.

Several studies on the combination of PSO or ACO with other optimization algorithms have been proposed in order to improve the performance of the original optimization model. In (Juang, 2004), a hybrid of GA and PSO, called HGAPSO, was proposed. In HGAPSO, new individuals are created not only by PSO but also by the crossover and mutation operations of a GA. In (Fan et al., 2004; Juang & Hsu, 2005), the simplex method was introduced into PSO. In (Ling et al., 2008), a hybrid of PSO with wavelet mutation was proposed. To apply the ACO technique to solve continuous optimization problems, several studies on the combination of ACO with other continuous optimization methods have been performed (Feng & Feng, 2004; Ge et al., 2004). An ACO followed by immune operation for optimization in a continuous space was proposed in (Feng & Feng, 2004). The incorporation of a deterministic searching algorithm (the Powell method) into ACO for continuous optimization was proposed in (Ge et al., 2004).

This chapter studies the combination of PSO and ACO for FSs design. One problem of PSO in FS design is that its performance is affected by initial particle positions, which are usually randomly generated in a continuous search space. A poor initialization may result in poor performance. Searching in the discrete-space domain by ACO helps to find good solutions. However, the search constraint in a discrete-space domain restricts learning accuracy. The motivation on the combination of ACO and PSO is to compensate the aforementioned weakness of each method in FS design problems. Two combination approaches, sequential and parallel, for PSO and ACO proposed in (Juang & Lo, 2008; Juang & Wang, 2009) are described and discussed in this Chapter.

This chapter is organized as follows. Section 2 describes the FS to be designed. The rule generation algorithm and rule initialization are also described in this section. Section 3 describes PSO and how to apply it to FS design. Section 4 describes ACO and how to apply it to FS design. Section 5 describes the sequential combination of PSO and ACO for FS design. Section 6 describes the parallel combination of PSO and ACO for FS design. Finally, Section 7 draws conclusions.

2. Fuzzy systems

2.1 Fuzzy system functions

This subsection describes the FS to be designed. The FS is of zero-order Takagi-Sugeno-Kang (TSK) type. That is, the ith rule, denoted as R_i, in the FS is represented in the following form:

$$R_i : \text{If } x_1(k) \text{ is } A_{i1} \text{ And } \ldots \text{ And } x_n(k) \text{ is } A_{in}, \text{ Then } u(k) \text{ is } a_i \tag{1}$$

where k is the time step, $x_1(k),...,x_n(k)$ are input variables, $u(k)$ is the system output variable, A_{ij} is a fuzzy set, and a_i is a crisp value. Fuzzy set A_{ij} uses a Gaussian membership function

$$M_{ij}(x_j) = \exp\{-(\frac{x_j - m_{ij}}{b_{ij}})^2\} \tag{2}$$

where m_{ij} and b_{ij} represent the center and width of the fuzzy set A_{ij}, respectively. In the inference engine, the fuzzy AND operation is implemented by the algebraic product in fuzzy theory. Thus, given an input data set $\vec{x} = (x_1, \ldots, x_n)$, the firing strength $\phi_i(\vec{x})$ of rule i is calculated by

$$\phi_i(\vec{x}) = \prod_{j=1}^{n} M_{ij}(x_j) = \exp\left\{-\sum_{j=1}^{n}\left(\frac{x_j - m_{ij}}{b_{ij}}\right)^2\right\} \tag{3}$$

If there are r rules in an FS, the output of the system calculated by the weighted average defuzzification method is

$$u = \frac{\sum_{i=1}^{r}\phi_i(\vec{x})a_i}{\sum_{i=1}^{r}\phi_i(\vec{x})}, \tag{4}$$

where a_i is the rule consequent value in (1). There are a total of $D = r(2n+1)$ free parameters in an FS, all of which are optimized using the combination of PSO and ACO algorithms.

2.2 Rule generation and initialization

Most studies on SI-based FS design algorithms determine the number of rules by trial and errors and assign the initial FS parameters randomly and uniformly in the domain of each free parameter. The subsection describes one promising rule generation and initialization algorithm based on the fuzzy clustering-like approach that has been used in an SI algorithm (Juang et al., 2007). It is assumed that there are initially no rules in the designed FS. The rule generation method generates fuzzy rules online upon receiving training data. Rules are generated in order to ensure that at least one rule is activated with a firing strength larger than a pre-defined threshold $\phi_{th} \in (0,1)$ for each input $\vec{x}$. Geometrically, as Fig. 1 shows, this

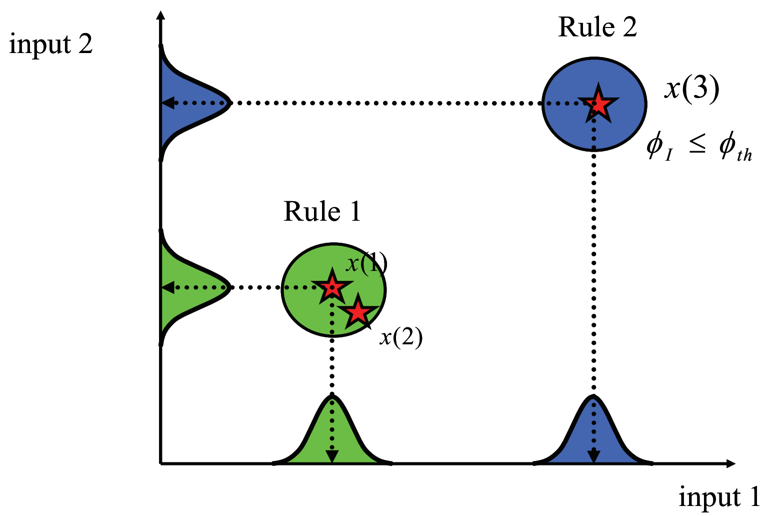


Fig. 1. Distributions of input data, generated fuzzy rules that properly cover the data, and initial shapes of the corresponding fuzzy sets in each input dimension.

threshold ensures that each input data is properly covered by a rule in the input space. According to this concept, the firing strength $\phi_i(\vec{x})$ in (3) is used as the criterion to decide if a new fuzzy rule should be generated. For each incoming piece of data $\vec{x}(k)$, find

$$I = \arg \max_{1 \le i \le r} \phi_i(\vec{x}(k)) \tag{5}$$

where r is the number of existing rules at time t. If $\phi_I \le \phi_{th}$ or r =0, then a new fuzzy rule is generated to cover $\vec{x}(t)$ and $r \leftarrow r+1$. A smaller ϕ_{th} value generates a smaller number of rules. The generation of the r th rule also generates the r th new fuzzy set in each input variable. That is, the number of fuzzy sets in each input dimension is equal to the number of fuzzy rules in the designed FS. To reduce the number of fuzzy sets in each input dimension, the fuzzy set generation criterion proposed in (Juang et al., 2007; Juang & Wang, 2009) can be further employed though it adds computation cost. For each newly generated fuzzy rule, the corresponding center and width of Gaussian fuzzy set A_{rj} in each input variable are assigned as follows:

$$m_{rj} = x_j(k),\ b_{rj} = b_{fix},\ j = 1,\ldots,n \tag{6}$$

where b_{fix} is a pre-specified constant value. Since the centers and widths of all fuzzy sets can be further tuned by PSO, all of the initial widths are simply set to the same value of b_{fix}.

3. Particle swarm optimization (PSO) for FS design

This section first describes the basic concept of PSO. The application of PSO to optimize the generated FS in Section 2 is then is then described. The swarm in PSO is initialized with a population of random solutions (Kennedy & Eberhart, 1995). Each potential solution is called a particle. Each particle has a position, which is represented by a position vector $\vec{s}_i$. A swarm of particles moves through the problem space, with the velocity of each particle represented by a velocity vector $\vec{v}_i$. At each time step, a function f is evaluated, using $\vec{s}_i$ as an input. Each particle keeps track of its own best position, which is associated with the best fitness it has achieved so far, in a vector $\vec{p}_i$. Furthermore, each particle is defined within the context of a topological neighborhood that is made up of itself and other particles in the swarm. The best position found by any member of the neighborhood is tracked in $\vec{p}_i^g$. For a global version of PSO, $\vec{p}_i^g$ is defined as the best position in the whole population. At each iteration t, a new velocity for particle i is obtained by using the individual best position, $\vec{p}_i(t)$, and the neighborhood best position, $\vec{p}_i^g(t)$:

$$\vec{v}_i(t+1) = w\vec{v}_i(t) + c_1\vec{\phi}_1 \cdot (\vec{p}_i(t) - \vec{x}_i(t)) + c_2\vec{\phi}_2 \cdot (\vec{p}_i^g(t) - \vec{x}_i(t)), \tag{7}$$

where w is the inertia weight, c_1 and c_2 are positive acceleration coefficients, and $\vec{\phi}_1$ and $\vec{\phi}_2$ are uniformly distributed random vectors in [0,1], where a random value is sampled for each dimension. The limit of $\vec{v}_i$ in the range $[-\vec{v}_{\max}, \vec{v}_{\max}]$ is problem dependent. For some problems, if the velocity violates this limit, it is reset within its proper limits. Depending on their velocities, each particle changes its position according to the following equation:

$$\vec{s}_i(t+1) = \vec{s}_i(t) + \vec{v}_i(t+1). \tag{8}$$

Based on (7) and (8), the particle population tends to cluster around the best.
The use of PSO for FS design, i.e., optimization of all free parameters in an FS, is described as follows. For the FS in (1) that consists of n input variables and r rules, all of its free parameters can be described by the following position vector

$$\vec{s} = [m_{11}, b_{11}, \cdots, m_{1n}, b_{1n}, a_1, \cdots \cdots, m_{r1}, b_{r1}, \cdots, m_{rn}, b_{rn}, a_r] \in \Re^D \tag{9}$$

After the rule generation and initialization process described in Section 2, the initial antecedent part parameters are determined. Based on the solution vector representation in (9) and the antecedent part parameter initialization in (6), the ith solution vector $\vec{s}_i$ is generated:

$$\begin{aligned} \vec{s}_i &= [s_{i1}\ s_{i2}\ \dots\ \dots\ s_{iD}] \\ &= [m_{11} + \Delta m^i_{11}, b_{fix} + \Delta b^i_{11}, \cdots, m_{1n} + \Delta m^i_{1n}, b_{fix} + \Delta b^i_{1n}, a_1, \cdots\cdots, \\ &\quad m_{r1} + \Delta m^i_{r1}, b_{fix} + \Delta b^i_{r1}, \cdots, m_{rn} + \Delta m^i_{rn}, b_{fix} + \Delta b^i_{rn}, a_r] \end{aligned} \tag{10}$$

where Δm_{ij} and Δb_{ij} are small random numbers. The parameter a_i is a random number randomly and uniformly distributed in the FS output range. The evaluation function f for a particle $\vec{s}_i$ is computed according to the performance of the FS constituted of the parameters in (10).

4. Ant colony optimization (ACO) for FS design

ACO is a meta-heuristic algorithm inspired by the behavior of real ants, and in particular how they forage for food (Dorigo & Caro, 1999; Dorigo & Stutzle, 2004). It was first applied to the traveling salesman problem (TSP). In ACO, a finite size colony of artificial ants is created. Each ant then builds a solution to the problem. While building its own solution, each ant collects information based on the problem characteristics and on its own performance. The performance measure is based on a quality function $F(\cdot)$. ACO can be applied to problems that can be described by a graph, where the solutions to the optimization problem can be expressed in terms of feasible paths on the graph. Among the feasible paths, ACO is used to find an optimal one which may be a locally or globally optimal solution. The information collected by the ants during the search process is stored in the pheromone trails, τ, associated to the connection of all edges. These pheromone trails play the role of a distributed long-term memory about the whole ant search process. The ants cooperate in finding a solution by exchanging information via the pheromone trials. Edges can also have an associated heuristic value, η, representing *a priori* information about the problem instance definition or run-time information provided by a source different from the ants. Ants can act concurrently and independently, showing a cooperative behavior. Once all ants have computed their tours (i.e. at the end of the each iteration), ACO algorithms update the pheromone trail using all the solutions produced by the ant colony. Each edge belonging to one of the computed solutions is modified by the amount of pheromone that is proportional to its solution value. The pheromone trail may be updated locally while an ant builds its trail or globally when all ants have built their trails.
Let $\tau_{ij}(t)$ be the pheromone level on edge (i, j) at iteration t, and η_{ij} be the corresponding heuristic value. The probability that an ant chooses j as the next vertex when it is at the vertex i at iteration t is given by

$$p_{ij}(t) = \begin{cases} \dfrac{\tau_{ij}(t)\eta_{ij}^{\beta}(t)}{\sum_{z\in J(i)}\tau_{iz}(t)\eta_{iz}^{\beta}(t)}, & \text{if } j \in J(i) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

where $J(i)$ is the set of vertices that remain to be visited by the ant, β is a parameter that determines the relative influence of the pheromone trail and the heuristic information. After all ants have completed their tours, the pheromone level is updated by

$$\tau_{ij}(t+1) = \rho\tau_{ij}(t) + \Delta\tau_{ij}(t) \,, \tag{12}$$

where $\rho \in (0,1)$ is a parameter such that $1-\rho$ represents the evaporation coefficient. The update value $\Delta\tau_{ij}$ is related to the quality value F. Many updating rules for $\Delta\tau_{ij}$ have been studied (Dorigo & Stutzle, 2004), like ant system (Dorigo et al., 1996), ant colony system (Dorigo & Gambardella, 1997), MAX MIN ant system (Stutzle & Hoos, 2000), and hypercube framework ACO (Blum & Dorigo, 2004). The major differences between these ACO algorithms and AS are the probability selection techniques or pheromone update.

ACO can be applied to design the consequent part parameters in an FS. The general design approach is described as follows. Consider the FS whose structure and antecedent part parameters are determined according to (5) and (6) in Section 2. Suppose the consequent part is selected from a discrete set $U = \{u_1,\ldots,u_m\}$. For each rule, there are m candidate actions to be chosen. Each rule with competing consequent may be written as

$$R_i: \text{If } x_1 \text{ is } A_{i1} \text{ And } \ldots \text{ And } x_n \text{ is } A_{in} \text{ Then } u(k) \text{ is } u_1 \text{ Or } u_2 \text{ Or } \ldots \text{ Or } u_m \,. \tag{13}$$

That is, we have to decide one from a total of m^r combinations of consequent parts. This combinatorial problem is solved by ACO. To select the consequent value of each rule by ACO, we regard a combination of selected consequent values for a whole FS as a tour of an ant. For example, in Fig. 2, there are four rules, denoted by $R_1,\ldots,R_4$, in an FS and three candidate values , $u_1,\ldots,u_3$, for each rule. Starting from the initial state, the nest, the ant moves through $R_1,\ldots,R_3$ and stops at R_4, where the tour of this ant is marked by a bold line. For each rule, the node visited by the ant is selected as the consequent part of the rule. For the whole FS constructed by the ant in Fig. 2, the consequent values in R_1, R_2, R_3, and R_4 are u_2, u_3 , u_1, and u_3, respectively. Selection of the consequent value is partially based on pheromone trails between each rule. The size of the pheromone matrix is $r \times m$ and each entry in the matrix is denoted by τ_{ij}, where $i = 1,\ldots,r$ and $j = 1,\ldots,m$. As shown in Fig. 2, when the ant arrives at rule R_q, then selection of the m candidate values (denoted by nodes) of R_{q+1} is partly based on τ_{q+1j}, $j = 1,\ldots,N$. By using only the pheromone matrix, the transition probability is defined by

$$p_{ij}(t) = \frac{\tau_{ij}(t)}{\sum_{z=1}^{m}\tau_{iz}(t)} \tag{14}$$

In an FS, different fuzzy rules may share the same consequent value. That is, when value a_j is selected as the consequent of rule i, it may also be selected as the consequent of following rules i+1,…,r. For this reason, the set $J(i)$ in (11) is released to the whole set U for all i in (14).

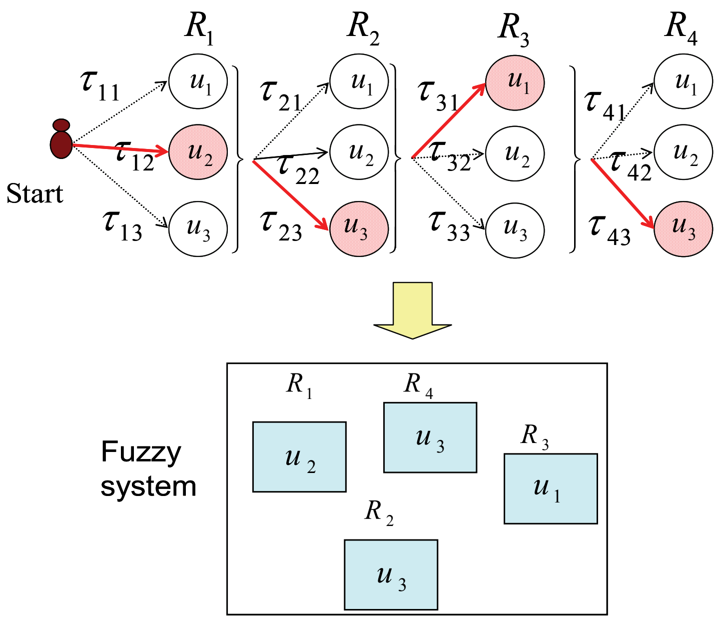


Fig. 2. The relationship between an ant path and the selected consequent values in an FS.

The ACO works without the use of heuristic values, and the consequent part can be simply selected by using (14). The use of heuristic values can be further employed for learning performance improvement. However, determination of the heuristic value η usually requires *a priori* information about the problem instance. For an unknown plant control problem using FSs, it is difficult to assign proper heuristic values in advance. For this problem, in (Juang & Lo, 2007), a new heuristic value assignment approach is proposed for controlling an unknown plant. For the control problems considered in this study, it is assumed that neither *a priori* knowledge of the plant model nor training data collected in advance are available. This study proposes an on-line heuristic value update approach according to temporal difference error between the actual output $y(k)$ and the desired output $y_d(k)$. In (Juang et al., 2008), a simple heuristic value assignment approach is proposed for controlling a plant with an unknown model except the information on the change of output direction with control input. This study assigns heuristic values to each candidate consequent value according to the corresponding fuzzy rule inputs which are control error $e(k) = y(k) - y_d(k)$ and its change with time $\Delta e(k) = e(k) - e(k-1)$. In (Juang & Lu, 2009), the q-values in a reinforcement fuzzy Q-learning algorithm are used as heuristic values for an unknown plant control. Each candidate is assigned with a q-value which is updated using success and failure reinforcement signals during the reinforcement learning process.

5. Sequential combination of ACO and PSO

This section describes the sequential combination approach of ACO and PSO proposed in (Juang & Lo, 2008). In this sequential combination approach, the rule consequent of each online generated rule described in Section 2 is first learned by ACO. The advantage of using ACO for rule consequent learning is that it can help determine a good fuzzy rule base for subsequent learning. However, the search constraint in a discrete-space domain restricts learning accuracy, and the ants do not optimize antecedent part parameters. Therefore, after ACO learning, PSO is then employed to further optimize both the antecedent and consequent parameters, where initial particles in PSO are generated according to learning results from ACO.

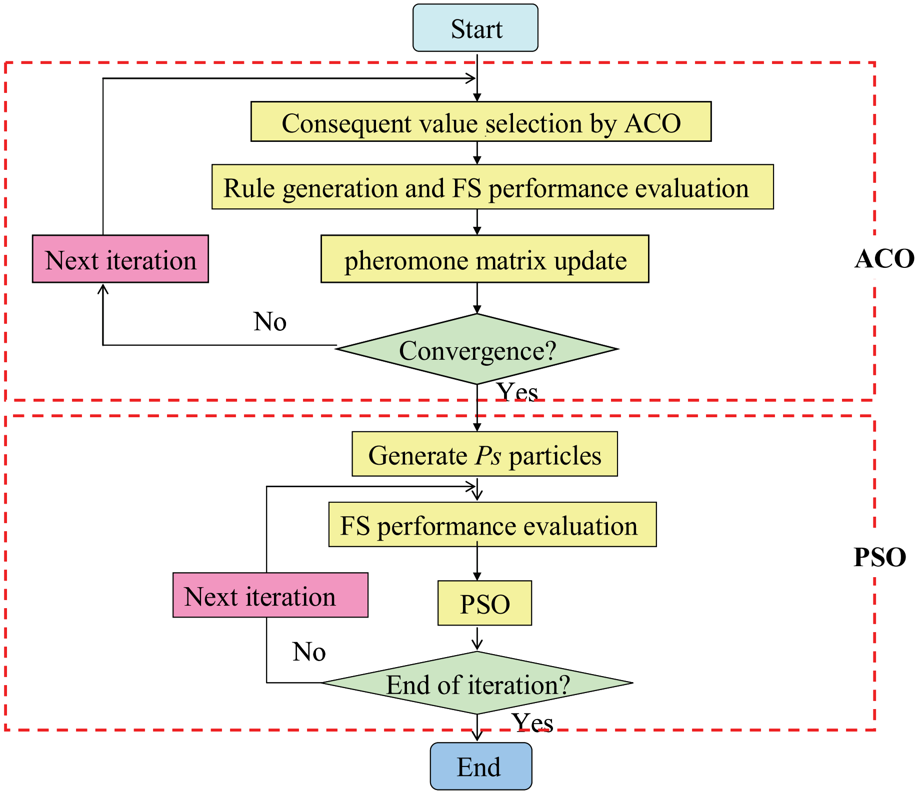


Fig. 3. Flow chart of sequential combination of ACO and PSO for FS design.

Figure 3 shows the flow of the sequential combination of ACO and PSO. The formulation of consequent part learning by ACO is described in Section 4. Detailed function evaluation and update of pheromone levels are described as follows. The pheromone trails, τ_{ij}, on the ant tour are updated according to the performance of the constructed FS. When an ant completes a tour, the corresponding FS is evaluated by a quality function F, which is defined as the inverse of learning error. A higher F value indicates better performance. Let the population size be P_s, meaning that there are P_s ants in a colony. For each iteration, after all the ants in the colony have completed their tours, i.e., after the construction of P_s FSs, select the one with the highest F from the initial iteration until now. If a new global best ant is found in this iteration, then pheromone trails on the tour traveled by the global best ant are updated; otherwise, no pheromone update is performed in this iteration. Denote the global best ant as q^* with the corresponding quality value as F_{q^*}. The new pheromone trail $\tau_{ij}(t+1)$ is updated by

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t)\,,\ \text{if } (i,j) \in \text{global-best-tour} \tag{15}$$

where $0 < \rho < 1$ is the pheromone trail evaporation rate and

$$\Delta\tau_{ij}(t) = F_{q*} \tag{16}$$

The ACO learning iteration above repeats until the criterion for switching is met. The switching point from ACO to PSO learning is determined by the learning error convergence property of the global best ant. Let $E(t)$ denote the error index of the global best ant at iteration t. For example, $E(t)$ can be defined as root-mean-squared error (RMSE) or sum of absolute error (SAE). If

$$\frac{E(t) - E(t+50)}{E(t)} < 1\% \tag{17}$$

then ACO learning terminates and learning switches to PSO.

Using PSO releases the discrete space constraint imposed on consequent parameters when ACO is used, and searches the best consequent parameters in continuous space. In addition to the consequent parameters, PSO also searches the optimal antecedent part parameters. Like ACO, population size in the PSO is equal to P_S. The elements in position $\vec{s}$ are set as in (9). At iteration $t = 0$, the initial positions $\vec{s}_1(0), \cdots, \vec{s}_{P_s}(0)$ are generated randomly according to the best-performing FS, denoted as $\vec{s}_{ACO}$, found in ACO. Position $\vec{s}_1(0)$ is set to be the same as $\vec{s}_{ACO}$. The left $P_S - 1$ particles, $\vec{s}_2(0)$, ..., $\vec{s}_{P_s}(0)$, are generated by adding uniformly distributed random numbers to $\vec{s}_{ACO}$. That is,

$$\vec{s}_i(0) = \vec{s}_{ACO} + \vec{w}_i\,,\ i = 2, \ldots, P_s \tag{18}$$

where $\vec{w}_i$ is a random vector. The initial velocities, $\vec{v}_i(0)$, $i = 1, \ldots, P_s$, of all particles are randomly generated. The performance of each particle is evaluated according to the FS it represents. The evaluation function f is defined as the error index $E(t)$ described above. According to f, we can find individual best position $\vec{p}_i$ of each particle and the global best particle $\vec{p}_g^{\,i}$ in the whole population. Velocity and position of each particle are updated using (7) and (8), respectively. The whole learning process ends when a predefined criterion is met. In (Juang & Lo, 2008), the criterion is the total number of iterations. In (Juang & Lo, 2008), the sequential combination of ACO and PSO approach for FS design has been applied to different control problems, including chaotic system regulation control, nonlinear plant tracking control, and water bath temperature control. The performance of the sequential combination approach has been shown to be better than those of ACO, PSO and different existing FS design methods which were applied to the same problem.

6. Parallel combination of ACO and PSO

This section describes the parallel combination approach of ACO and PSO for FS design (Juang & Wang, 2009). Like the sequential combination approach, the parallel combination approach uses a constant population size and is denoted as P_s =2N. Each individual in the population represents a parameter solution of the FS as described in (9). An individual may be generated by an ant path in ACO or a particle in PSO. Individuals generated by ants and particles are called ant individuals and particle individuals, respectively. Figure 4 shows a block diagram of the algorithm. Generation of population individuals are described as follows.

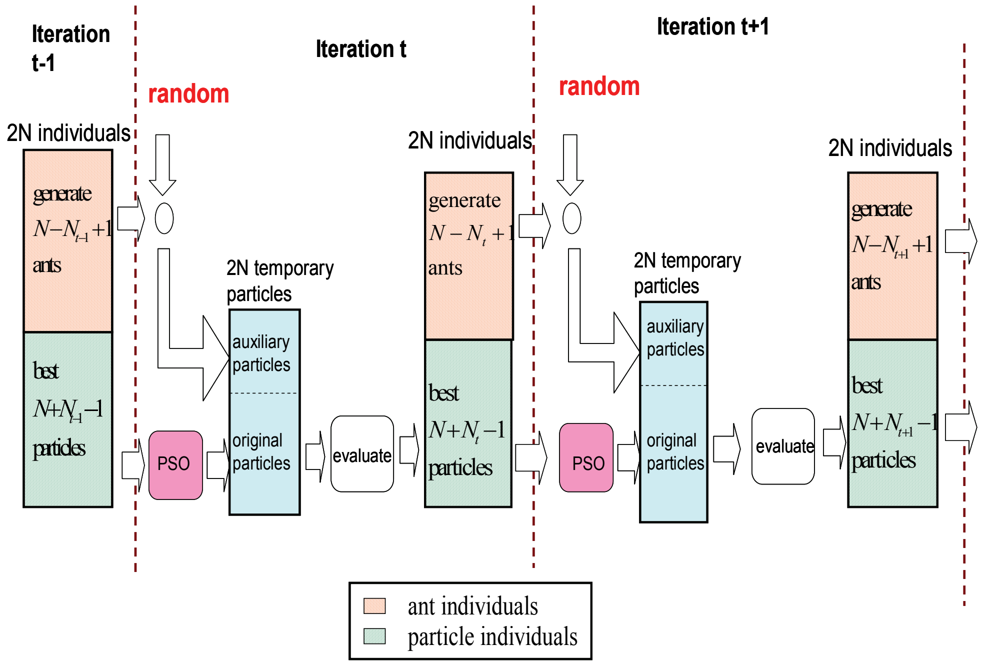


Fig. 4. Parallel combination of ACO and PSO for FS design.

In the first iteration, the rule generation algorithm in Section 2 and N different ant paths generate half of the population. The other half are generated from particle individuals. The N ant individuals contain no rules initially. New rules are generated using the criterion in (5) during the evaluation of an ant individual (FS). If the number of rules after evaluation of the N ant individuals is r_1, then the number of rules in the N particle individuals are all equal to r_1. In the parallel combination approach, the objective of using PSO is to optimize both the antecedent and consequent parameters in existing fuzzy rules; therefore, no rules are generated during the performance evaluation of a particle (FS). The initial N particle position vectors are generated using (10).

The second and subsequent iterations generate a new population with 2N new individuals. For the generation of new individuals in each iteration, the ACO-based FS design approach in Section 4 is used. In this approach, N ant paths generate N ant individuals (FSs). During the new ant individual generation process, some ants may choose the same path and generate the same individuals. This phenomenon becomes more obvious as more iterations are conducted due to pheromone matrix convergence. To consider this phenomenon, suppose that N_t ants have the same path at the tth iteration. Then only N- N_t +1 different ant individuals are reserved in the population. The performance of these N- N_t +1 individuals is evaluated and the pheromone matrix is updated according to their performance. The remaining N- N_t +1 new individuals in the population (population size is fixed at 2N) in iteration t are partly generated by particles in the last iteration. The original N + N_{t-1} - 1 particle individuals from the last iteration are optimized by PSO. The performance of these optimized particle individuals is then evaluated. In addition to these optimized particle individuals, the N - N_{t-1} + 1 ant individuals in the previous iteration (iteration t-1) also help generate auxiliary particles to improve particle search performance. Adding these ant individuals with small random values generates auxiliary particles. The purpose of adding small random values is to distinguish auxiliary particles from existing ant individuals and to improve the algorithm's exploration ability. Suppose an original individual has the form in (10), then the auxiliary particle takes the following form

$$[m_{11}+\Delta m_{11},\sigma_{11}+\Delta\sigma_{11},\cdots,m_{1n}+\Delta m_{1n},\sigma_{1n}+\Delta\sigma_{1n},a_1+\Delta a_1,\cdots,m_{rn}+\Delta m_{rn},\sigma_{rn}+\Delta\sigma_{rn},a_r+\Delta a_r] \quad (19)$$

The performance of these N - N_{t-1} + 1 auxiliary particles is then computed. These auxiliary particles together with the original N + N_{t-1} - 1 particles constitute a total of 2N temporary particles. Only the best N + N_t - 1 particles are reserved from among these 2N particles. In the next iteration, these reserved particles cooperate to find better solutions through PSO.

For ant individual update by ACO at the end of each iteration, as in Section 4, a new ant path generates a new FS (individual) according to the pheromone levels and transition probability in (14). The pheromone levels are updated using (15) and (16). For particle individual update in iteration t, PSO updates all of the N + N_{t-1} - 1 particles generated either from auxiliary particles or original particle individuals in the previous iteration. Positions and velocities are updated based on (7) and (8) using a local version of PSO. For neighborhood best particle $\vec{p}_i^g(t)$ computation, the neighbors of a particle with r_{t-1} rules for finding $\vec{p}_i^g(t)$ are defined as the particles that also have r_{t-1} rules. As in the sequential combination approach, the learning process ends when a pre-defined number of iterations is performed.

In (Juang & Wang, 2009), the parallel combination learning approach has been applied to two control examples, nonlinear plant tracking control and reversing a truck following a circular path. These examples generate training data only when fuzzy control starts. Simulation results show that the proposed method achieves a smaller control error than ACO, PSO, and other different SI learning algorithms used for comparison in each example.

7. Conclusion

This chapter describes the design of FSs using PSO, ACO, and their sequential and parallel combination approaches. The use of on-line rule generation not only helps to determine the number of fuzzy rules, but also helps to locate the initial antecedent parameters for subsequent parameter learning using PSO. For PSO, the incorporation of ACO helps to locate a good initial fuzzy rule base for further PSO learning. For ACO, the incorporation of PSO helps to find the parameters in a continuous space. The cooperative search of ACO and PSO compensates for the searching disadvantage of each optimization method. Reported results show that the two combination approaches outperform different advanced PSO and

ACO algorithms for FS design problems. Performance of these two combination approaches on other optimization problems will be studied in the future. Other different combination approaches of ACO and PSO may also be studied for further performance improvement.

8. References

Araujo, E. & Coelho, L. d. S. (2008). Particle swarm approaches using Lozi map chaotic sequences to fuzzy modeling of an experimental thermal-vacuum system, *Applied Soft Computing*, Vol. 8, No. 4, 1354-1364.

Blum, C. & Dorigo M. (2004). The hyper-cube framework for ant colony optimization, *IEEE Trans. Syst., Man, and Cyber.-Part B: Cybernetics*, Vol. 34, No. 2, 1161-1172.

Bergh, F. van den & Engelbrecht, A. P. (2004). A cooperative approach to particle swarm optimization, *IEEE Trans. Evolutionary Computation*, vol. 8, no. 3, 225-239.

Cassillas, J.; Cordon, O. & Herrera F. (2000). Learning fuzzy rules using ant colony optimization algorithms, *Proc. 2nd Workshop on Ant Algorithms – from Ant Colonies to Artificial Ants,* pp. 13-21, Brussels, Belgium.

Clerc, M. & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimentional complex space, *IEEE Trans. Evolutionary Computation*, Vol. 6, No. 1, 58-73.

Cassillas, J.; Cordon, O.; Viana, I. F. & Herrera, F. (2005). Learning cooperative linguistic rules using the best-worst ant system algorithm, *Int. Journal of Intelligent Systems*, Vol. 20, 433-452.

Chatterjee, A.; Pulasinghe, K.; Watanabe, K. & Izumi, K. (2005). A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems, *IEEE Trans. Industrial Electronics*, Vol. 52, No. 6, 1478-1489.

Chen, X. & Li, Y. (2007). A modified PSO structure resulting in high exploration ability with convergence guaranteed, *IEEE Trans. Syst., Man, and Cyber., Part B: Cybernetics*, Vol. 37, No. 5, 1271-1289.

Dorigo, M.; Maniezzo, V. & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents, *IEEE Trans. on Syst., Man, and Cybe., Part B: Cybernetics*, Vol. 26, No. 1, 29-41.

Dorigo, M. & Gambardella, L.M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Trans. on Evolutionary Computation*, Vol. 1, No. 1, 53-66.

Dorigo, M. & Caro, G. D. (1999). The ant colony optimization meta-heuristic, *New Ideas In Optimization*, D. Corne, M. Dorigo, & F. Glover Eds., pp. 11-32, London, McGraw-Hill.

Dorigo, M. & Stutzle, T. (2004). Ant Colony Optimization, MIT.

Feng, Y. J. & Feng, Z. R. (2004). An immunity-based ant system for continuous space multi-modal function optimization, *Proc. Int. Conf. Machine Learning and Cybernetics*, Vol. 2, pp. 1050-1054.

Fan, S. K. S. ; Liang, Y. C. & Zahara, E. (2004). Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions, *Engineering Optimization*, Vol. 36, No. 4, 401-418.

Ge, Y.; Meng, Q. C.; Yan, C. J. & Xu, J. (2004). A hybrid ant colony algorithm for global optimization of continuous multi-extreme functions, *Proc. Int. Conf. Machine Learning and Cybernetics*, Vol. 4, pp. 2427-2432.

Juang, C. F. (2004). A hybrid of genetic algorithm and particle swarm optimization for recurrent network design, *IEEE Trans. Syst., Man, and Cyber., Part B: Cybernetics*, Vol. 34, No. 2, 997-1006.

Juang, C. F. & Hsu, C. H. (2005). Temperature control by chip-implemented adaptive recurrent fuzzy controller designed by evolutionary algorithm, *IEEE Trans. Circuits and Systems- I: Regular Papers*, Vol. 52, No. 11, 2376-2384.

Juang, C. F.; Chung, I. F. & Hsu, C. H. (2007). Automatic construction of feedforward /recurrent fuzzy systems by clustering-aided simplex particle swarm optimization, *Fuzzy Sets and Systems*, Vol. 158, No. 18, 1979-1996.

Juang, C. F. & Lo, C. (2007). Fuzzy systems design by clustering-aided ant colony optimization for plant control, *Int. Journal of General Systems*, Vol. 36, No. 6, 623-641.

Juang, C. F.; Lu, C. M.; Lo, C. & Wang, C. Y. (2008). Ant colony optimization algorithm for fuzzy controller design and its FPGA implementation, *IEEE Trans. Industrial Electronics*, Vol. 55, No. 3, 1453-1462.

Juang, C. F. & Lo, C. (2008). Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence, *Fuzzy Sets and Systems*, Vol. 159, No. 21, 2910-2926.

Juang, C. F. & Lu, C. M. (2009). Ant colony optimization incorporated with fuzzy Q-learning for reinforcement fuzzy control, *IEEE Trans. Syst., Man, and Cyber., Part A: Systems and Humans*, Vol. 39, No. 3, 597-608.

Juang, C. F. & Wang, C. Y. (2009). A self-generating fuzzy system with ant and particle swarm cooperative optimization, *Expert Systems with Applications*, Vol. 36, No. 3P1, 5362-5370.

Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization, *Proc. of IEEE Int. Conf. on Neural Networks*, Perth, Australia, pp. 1942-1948.

Kennedy, J.; Eberhart, R. & Shi, Y. (2001). Swarm Intelligence, Morgan Kaufmann Publisher.

Kennedy, J & Mendes, R. (2006). Neighborhood topologies in fully informed and best-of-neighborhood particle swarms, *IEEE Trans. Syst., Man, and Cyber., Part C: Applications and Reviews*, Vol. 36, No. 4, 515 – 519.

Ling, S. H. ; Iu, H. H. C. ; Chan, K. Y. ; Lam, H. K. ; Yeung, B. C. W. & Leung, F. H. (2008). Hybrid particle swarm optimization with wavelet mutation and its industrial applications, *IEEE Trans. Syst., Man, and Cyber., - Part B: Cybernetics*, Vol. 38, No. 3, 743-763.

Mucientes, M. & Casillas, J. (2007). Quick design of fuzzy controllers with good interpretability in mobile robotics, *IEEE Trans. Fuzzy Systems*, Vol. 15, No. 4, 636-651.

Parrott, D. & Li, X. (2006). Locating and tracking multiple dynamic optima by a particle swarm model using speciation, *IEEE Trans. Evolutionary Computation*, Vol.10, No.4, 440-458.

Ratnaweera, A.; Halgamuge, S. K. & Watson, H. C. (2004), Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Trans. Evolutionary Computation*, Vol. 8, No. 3, 240-255.

Stutzle, T. & Hoos, H. H. (2000). MAX MIN ant system, *Journal of Future Generation Computer Systems*, Vol. 8, No. 16, 889-914.

Sharma, K. D.; Chatterjee, A. & Rakshit, A. (2009). A hybrid approach for design of stable adaptive fuzzy controllers employing Lyapunov theory and particle swarm optimization, *IEEE Trans. Fuzzy Systems*, Vol. 17, No. 2, 329-342.

12

Triangle Formation of Multi-Agent Systems with Leader-Following on Fuzzy Control

Hongyong Yang[1] and Jianzhong Gu[2]
[1]*School of Computer Science and Technology,*
[2]*School of Information and Mathematics*
Ludong University
Yantai, 264025
P. R. China

1. Introduction

In recent years, there have been an increasing number of achievements dealing with the control of agent formations (Olfati-Saber, 2006; Tanner & Jadbabaie, 2007; Fax & Murray, 2004; Jadbabaie et al., 2003; Lin et al., 2004; Olfati-Saber & Murray, 2004). The formation control of multiple moving agents has emerged as a topic of widespread interest due to its broad range of applications in military missions, environmental surveying, and space missions.

Multiple mobile autonomous agents coupled with each other through interactions can generate certain ordered behaviors, such as aggregation, cohesion, alignment, rotation and synchronization (Fax & Murray, 2004; Jadbabaie et al., 2003; Lin et al., 2004; Olfati-Saber & Murray, 2004). Among the typical approaches to formation control, distributed control strategies have aroused researchers' outstanding attention because there is no centralized supervisor and only a little sense of communication information is needed. Recently, there has been a tendency to deal with the formation control as a consensus problem. In the multi-agent systems, consensus means to reach an agreement by means of an interaction rule that specifies the information exchange between an agent and its neighbors (Fax & Murray, 2004; Jadbabaie et al., 2003; Lin et al., 2004; Olfati-Saber & Murray, 2004). Fax & Murry (Fax & Murray, 2004) analyzed the stability of the formation control with first-order consensus protocols based on the graph Laplacian. Jadbabaie et al. (Jadbabaie et al., 2003) demonstrated the results of the alignment problem which is concerned with reaching an agreement without computing any objective functions. Lin et al. (Lin et al., 2004) studied three formation strategies for groups of mobile autonomous agents with local communication topology. Moreover, by using a Lyapunov-based approach, Olfati-Saber & Murry (Olfati-Saber & Murray, 2004) solved the consensus problems in networks of agents with directed interconnection graphs and time delays.

The key idea involved in nonlinear formation control is to preserve the inter-agent distances from decaying to zero during the motion for the purpose of collision avoidance. In the formation control, tracking control problems of multi-agent systems are studied increasingly, and many results have been obtained with nearest neighbor-based rule (Hong et al., 2006; Anderson et al., 2007; Chen & Tian, 2009). Hong et al. (Hong et al., 2006) considered the tracking control of mobile agents' consensus with unmeasurable velocity or acceleration information for an active leader. Anderson et al. (Anderson et al., 2007) solved the formation control problem for the system of three-coleader agents described by first-order integrators, that is, each agent should retain a distance from other agents. Promoted by this work, Chen & Tian (Chen & Tian, 2009)) tackled three-coleader formation control problem with second-order integrators of the agent dynamics by applying a backstepping method.

In this paper, like many predecessors, we study the formation control problem of mobile agent systems to maintain the desired velocity and the inter-agent distances. We consider a simple formation with just heterogeneous three-agent system, one agent moving forward as a leader and others following the leader, that is, agent 0 (leader angent) moves freely, and the following agent 1 should maintain a distance from agent 0, agent 2 should maintain a distance from agent 0, and agent 1 and agent 2 should maintain a distance from each other. By applying fuzzy logical controller (FLC), the formation of three agents is achieved. Best to our knowledge, there are not studies for the formation multi-agent system by applying FLC.

This chapter is organized as follows: Section 2 gives the system model and the problem statement. The formation control based on fuzzy logical control is presented in Section 3. In Section 4, many computer simulations are applied to verify the formation control, and the robustness of the system is discussed by adding noise and bias. Conclusions are provided in Section 5.

2. Problem description

In this paper, we study a formation control problem of the system with three heterogeneous agents, one agent moving as a leader and the others following the leader (Fig. 1). Based on the distance preserved between each other, the followers will be moving in the direction of the leader.

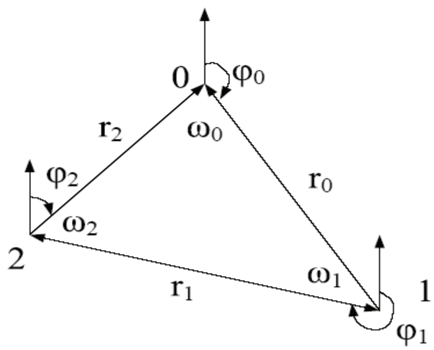


Fig. 1. Formation with three heterogeneous agents.

Notation is defined in reference to Fig. 1. Agent 0 is the leader with the dynamics:

$$\begin{aligned} \dot{z}_0 &= v_0, \\ \dot{v}_0 &= f(t, z_0, v_0). \end{aligned} \tag{1}$$

where z_0 is the position of agent 0 and v_0 is its velocity, $f(.,.,.)$ is piecewise continuous in t and locally Lipschitz in z_0 and v_0. Other agents are the followers with the variable position:

$$z_i = (x_i, y_i), i = 1,2 \tag{2}$$

and the velocity $v_i, i = 1,2$.
Fig. 1 illustrates this directed formation control system: φ_i is the angle between north and the direction of agent *i*, as seen from agent $i+1$ (agent 0 being identified with agent 3, here and subsequently); r_i is the current distance between agent *i* and agent $i+1$; d_i is the distance that ought to be maintained between agent *i* and $i+1$; ω_i is the internal angle of the triangle formed by the three agents, at agent *i*, for $i = 0, 1, 2$.
In order to control the formation of three agents, we shall make two standing assumptions. First, in the triangle, it is assumed that for $i \neq j \neq k$, the triangle inequality $d_i + d_j > d_k$ holds; thus the steady state to which the formation is supposed to tend is well-defined as a triangle. Second, it is assumed that following agent *i* just knows the positions and velocities about itself and its neighbors.
In Fig. 1, the summation of the internal angle of the triangle $\omega_0 + \omega_1 + \omega_2 = \pi$ holds, and

$$\begin{aligned} \varphi_1 - \varphi_0 &= -(\pi + \omega_1), \\ \varphi_0 - \varphi_2 &= \pi - \omega_0, \\ \varphi_2 - \varphi_1 &= \pi - \omega_2. \end{aligned} \tag{3}$$

We assume the control law of agent *i* can only use its local information. The formation control problem is to design the directed control law by using local information, such that three agents achieve the nominated formation, i.e.

$$\lim_{t\to\infty} r_i = d_i, i = 0,1,2$$

with the common desired velocity,

$$\lim_{t\to\infty} v_i = v_0, i = 1,2 .$$

Manuscript must contain clear answers to following questions: What is the problem / What has been done by other researchers and where you can contribute / What have you done / Which method or tools you used / What are your results / What is new and good, what is not good / Future research

3. Fuzzy logical controller for the formation of multi-agent systems

In this section, we present a Fuzzy logical control law for the formation of heterogeneous three-agent system. Following the moving disciplinarian of the system, the agent should satisfy, for $i = 1, 2$

$$\dot{z}_i = v_i .$$

Let

$$\dot{v}_i = u_i,$$

where u_i is the control input to adjust the velocity with the moving.
With the movement of the leader agent, the follower agent can calculate the distances between its neighbors. If the distance is larger than the maintained value, its velocity will be increased, otherwise, it will be decreased. Based on this moving mechanism, we construct a fuzzy logical controller to control the agent's dynamic track. This controller has two input variables and one output variable, which the distance and its difference variable are used as input parameters and the intensity of the velocity is determined by the output from the fuzzy controller.
In order to ensure the smooth of the moving track, it is asked to adjust the velocity slowly. If the regulation is too strong, the moving locus will be concussed greatly. By the upwards principle, we will build the following fuzzy logical controller.
Let the distance of two neighbors $r_i(t)$, the quantity estimate is $E_i = \frac{r_i(t) - d_i}{d_i} \times 6$, then the subjection function can be calculate by the value of E_i (Fig. 1). Let the difference variable of the distance $\Delta r_i(t) = r_i(t) - r_i(t-1)$, the quantity estimate is $\Delta E_i = \frac{\Delta r_i(t)}{2d_i} \times 6$, then the subjection function can be calculate by the value of ΔE_i (Fig. 2). By fuzzy logical rules (Table 1), FLC determines the size of the output c, where ZO PS PM PB PVB NM and NB is Zero positive-small positive-middle positive-big positive-very-big negative-middle negative-big, respectively.
The output of the FLC determines the size of the repulsion/attraction. When the output is positive, it denotes that attraction is required, and the velocity of agent will be increased. When the output is negative, repulsion is asked, and the velocity of agent will be decreased. By the quantity estimate, the fact value corresponding with the output c is $\frac{c}{6} \times M$ with sufficient large number M.

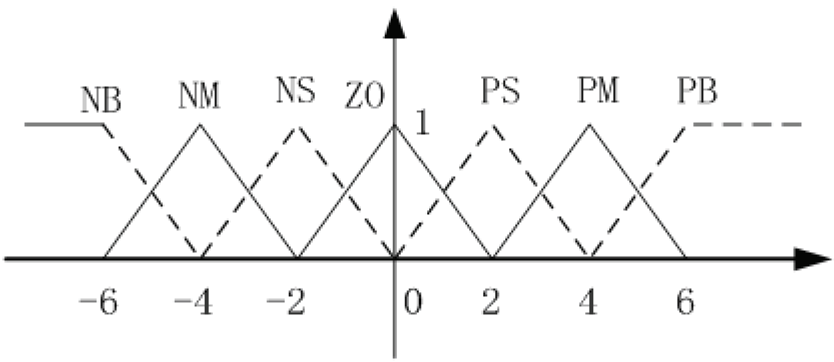


Fig. 2. The subjection function of the pair distance E_i and ΔE_i

$E_i \setminus \Delta E_i$	PB	PM	PS	ZO	NS	NM	NB
PB	PB	PB	PB	PB	PM	PS	ZO
PM	PB	PB	PB	PM	PS	ZO	NS
PS	PB	PB	PM	PS	ZO	NS	NM
ZO	PB	PM	PS	ZO	NS	NM	NB
NS	PM	PS	ZO	NS	NM	NB	NB
NM	PS	ZO	NS	NM	NB	NB	NB
NB	ZO	NS	NM	NB	NB	NB	NB

Table 1. Fuzzy rules

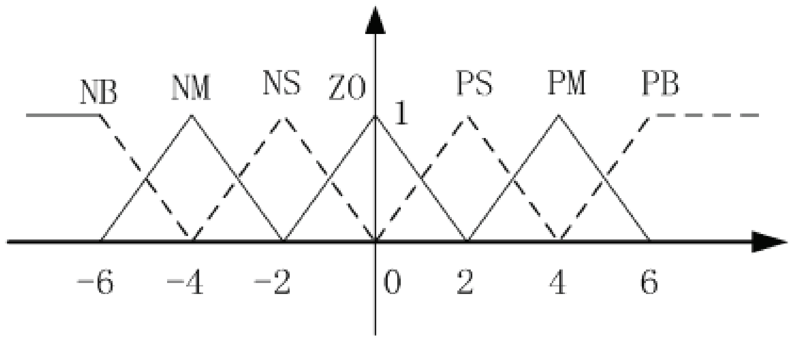


Fig. 3. The subjection function of the output *c*

Suppose there are *N* "if-then" rules,

$$L^{(i)} : \text{If } x_1 \text{ is } F_1^i \text{ , and } x_2 \text{ is } F_2^i ,\ldots\ldots, \text{and } x_n \text{ is } F_n^i \text{ then } y^i \text{ is } Y^i$$

for $i = 1,\ldots\ldots,N$. Based on the constructing process, we apply the product discursion, single value fuzziness, center average defuzziness and trigonal subjection function, to establish the FLC

$$y(x) = \frac{\sum_{i-1}^{N} \prod_{k=1}^{n} \mu_{F_k^i}(x_k) y^i}{\sum_{i-1}^{N} \prod_{k=1}^{n} \mu_{F_k^i}(x_k)} . \tag{4}$$

In order to comprehend the process of FLC easily, we illuminate the use of FLC by an example. Suppose the parameters d1=*3cm, d2*=*4cm, d3*=*5cm*, the distance between agent *i* and agent j is $r_2 = 4.5cm$, the diversification of the distance $\Delta r_i = -2cm$. Applying the fuzzy process, x_1=0.75, x_2= -1.5, and the following fuzzy rules obtain from the table 1:

If x1 is ZO with grade 0.635 and x2 is ZO with grade 0.25, then c is ZO
If x1 is ZO with grade 0.635 and x2 is NS with grade 0.75, then c is NS
If x1 is PS with grade 0.375 and x2 is ZO with grade 0.25, then c is PS
If x1 is PS with grade 0.375 and x2 is NS with grade 0.75, then c is ZO

Then the every rule output c is 0 , -2 , 2 , 0 by Fig.2, respectively. From Eq.(4), the last output of the FLC

$$c = \frac{0.635 * 0.25 * 0 + 0.635 * 0.75 * (-2) + 0.375 * 0.25 * 2 + 0.375 * 0.75 * 0}{0.635 * 0.25 + 0.635 * 0.75 + 0.375 * 0.25 + 0.375 * 0.75}$$
$$= -0.765$$

Corresponding the function value of the basis fields

$$\frac{c}{6} \times M = -0.1275M \, .$$

We can apply this output to control the moving of multi-agents by adjusting the parameter M.

4. Simulations analysis

In this section, we apply the simulations to analyze the results of the formation control of heterogeneous three-agent system. Suppose the maintained distances $d_0 = 3, d_1 = 4, d_2 = 5$, the dynamical velocity of the leader (agent 0)

$$v_0 = (0.1t^2, \sin 0.2t)^T \, .$$

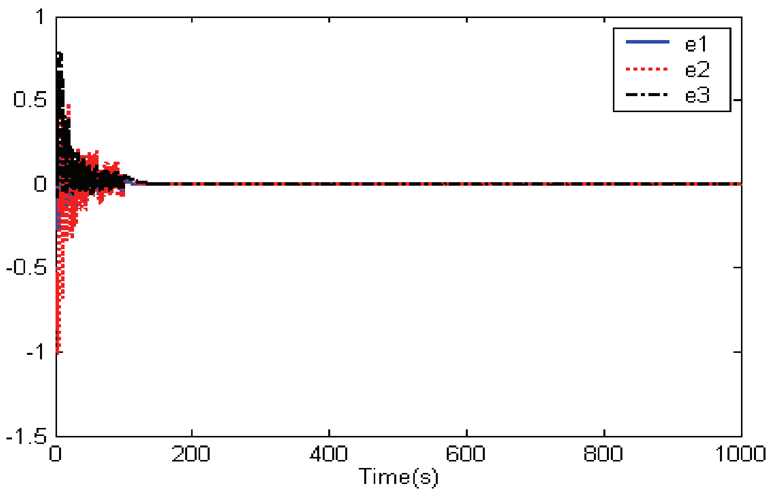


Fig. 4. Plot of errors between current distances and expected values

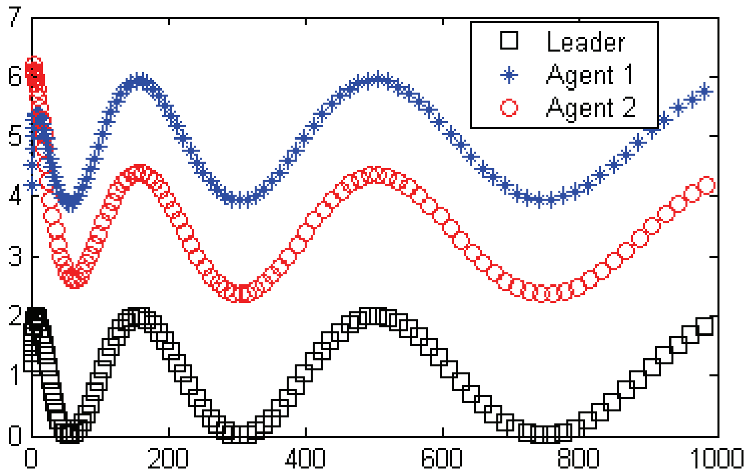


Fig. 5. Plot of agents' tracks.

Let initialized positions get randomly by computer. The errors ($e_i = r_i - d_i$) between the current distance r_i and the expected value d_i are shown in Fig. 4, and the information states of the three agents in Fig 5, respectively. These figures illuminate that the distances among the three agents reach the expected values, and the synchronizations of the moving tracks have been achieved.

5. Conclusion

This paper studies a directed formation control problem of heterogeneous multi-agent systems. The system is composed of a leader agent and two following agents with each one required maintaining a nominated distance from its neighbors. The follower is allowed to determine its movement strategy by using local knowledge of the direction of its neighbors and the current and desired distance from its neighbors. Based on the moving mechanism, a Fuzzy logical controller for multi-agent systems with leader-following is presented, which can not only accomplish the desired triangle formation but also ensure that the followers' speeds converge to the leader's velocity without collision during the motion. Simulation results are provided to illustrate the effectiveness of the control law. The proposed Fuzzy logical controller is interesting for the design of optimization algorithms that can ensure the triangle formation that multi-agent systems are required maintaining a nominated distance.

6. Acknowledgments

This work is supported by National Nature Science Foundation of China (under Grant 60774016, 60875039) and Science Foundation of Education Office of Shandong Province of China (Under grant J08LJ01)

7. References

Anderson, B.; Yu, C. & Dasgupta, S. et al. (2007). Control of a three coleaders formation in the plane. *Systems & Control Letters*, Vol. 56, 573-578, ISSN: 0167-6911

Chen, Y. & Tian, Y. (2009). A backstepping design for directed formation control of three-coleader agents in the plane. *International Journal of Robust and Nonlinear Control*, Vol.19, 729-745, ISSN: 1049-8923

Fax, A. & Murray, R. M. (2004). Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control*, Vol. 49, 1453-1464, ISSN: 0018-9286

Hong, Y.; Hu, J. & Gao, L. (2006). Tracking control for multi-agent consensus with an active leader and variable topology. *Automatica*, Vol. 42, 1177-1182, ISSN: 0005-1098

Jadbabaie, A.; Lin, J. & Morse, A. S. (2003). Coordination of groups of mobile autonomous agents using nearest neighbour rules. *IEEE Transactions on Automatic Control*, Vol. 48, 998-1001, ISSN: 0018-9286

Lin, Z.; Broucke, M. & Francis, B. (2004). Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control*, Vol. 49, 622-629, ISSN: 0018-9286

Tanner, G. H. & Jadbabaie, A. (2007). Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, Vol 52, 863-868, ISSN: 0018-9286

Olfati-Saber, R. (2006). Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, Vol, 51, 401-420, ISSN: 0018-9286

Olfati-Saber, R. & Murray, R. M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, Vol. 49, 1520-1533, ISSN: 0018-9286