

# Picture inference system: a new fuzzy inference system on picture fuzzy set

Le Hoang Son<sup>1,2</sup> · Pham Van Viet<sup>3,4</sup> · Pham Van Hai<sup>4</sup>

Published online: 29 October 2016  
© Springer Science+Business Media New York 2016

**Abstract** In this paper, we propose a novel fuzzy inference system on picture fuzzy set called picture inference system (PIS) to enhance inference performance of the traditional fuzzy inference system. In PIS, the positive, neutral and negative degrees of the picture fuzzy set are computed using the membership graph that is the combination of three Gaussian functions with a common center and different widths expressing a visual view of degrees. Then, the positive and negative defuzzification values, synthesized from three degrees of the picture fuzzy set, are used to generate crisp outputs. Learning in PIS including training centers, widths, scales and defuzzification parameters is also discussed. The system is adapted for all architectures such as the Mamdani, the Sugeno and the Tsukamoto

fuzzy inferences. Experimental results on benchmark UCI Machine Learning Repository datasets and an example in control theory - the Lorenz system are examined to verify the advantages of PIS.

**Keywords** Fuzzy inference system · Inference performance · Lorenz system · Picture fuzzy set · Picture inference system

## 1 Introduction

Inference is the process of formulating a nonlinear mapping from a given input space to output space, which provides a basis from which decisions can be made. The process of fuzzy inference involves all the membership functions, operators and if–then rules [30]. *Fuzzy inference system* (FIS) is also called as fuzzy rule based system, fuzzy expert system, fuzzy associative memory, fuzzy controller, fuzzy model or simply fuzzy system on the basis of the target for which the system is designed [7]. For instance, if the target of a system consists of temperature controlling tasks then the fuzzy system will be called as fuzzy controller and if the target is an expertise in medicine then the designed system is called fuzzy expert system [7].

Fuzzy inference systems are widely applicable in economic, scientific and engineering application areas due to the intuitive nature of the system and ability to analyze human judgments. For example, Khan, Daachi & Djouani [14] presented a fault detection for wireless sensor networks based on modeling a sensor node by Takagi–Sugeno–Kang (TSK) FIS, where a sensor measurement of a node is approximated by a function of the sensor measurements of the neighboring nodes. Nayak et al. [25] developed a FIS for predicting customer buying behavior where three

---

This work is dedicated to Prof. Bui Cong Cuong (Institute of Mathematics, VAST) for 3-year presence of the picture fuzzy set.

---

✉ Le Hoang Son  
lehoangson@tdt.edu.vn  
  
Pham Van Viet  
phamvanviet93@gmail.com  
  
Pham Van Hai  
haipv@soict.hust.edu.vn

- <sup>1</sup> Division of Data Science, Ton Duc Thang University, Ho Chi Minh City, Vietnam
- <sup>2</sup> Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
- <sup>3</sup> VNU University of Science, Vietnam National University, Hanoi, Vietnam
- <sup>4</sup> School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam

different methods namely grid partitioning, fuzzy *c*-means and subtractive clustering were used to get the membership values during the fuzzification of inputs. Öztaysi et al. [26] described potential applications of FIS in disaster response, which is one of the critical stages of disaster management, which necessitates spontaneous decision making when a disaster occurs. There are numbers of FIS applications in various fields as shown in [31] ranging from computer networks design, diagnosis of prostate cancer, umbilical cord acid–base analysis, implementing onboard planning and scheduling for autonomous small satellite, to the applications in power systems.

A number of researchers in FIS have suggested how such systems can be tuned to **enhance inference performance** [2]. The first attempt involves the deterministic nature of the system. Cai, Hao & Yang [5] proposed an architecture for high-order fuzzy inference systems combining the kernel-based fuzzy *c*-means clustering and support vector machine. Chaudhari & Patil [7] proposed a multilayer system with defuzzification and weighted average, which can reduce the cost of defuzzification and lack of output expressivity that causes risk when used as a controller. Luo, Wang & Sun [20] introduced a novel adaptive stability scheme for a class of chaos system with uncertainties. Mieszkowicz-Rolka & Rolka [24] used a flow graph representation of a decision table with linguistic values. Pancho et al. [27] proposed visual representations of fuzzy rule-based inference for expert analysis of comprehensibility. Ton-That, Cao & Choi [24, 45] extended fuzzy associative memory, which can be viewed as the combination of associative memory and fuzzy logic. A series of works by Liu et al. in [8, 12, 16–19, 44] have presented some adaptive fuzzy controller designs for nonlinear systems. In [18], the systems are of the discrete-time form in a triangular structure and include the backlash and the external disturbance. A MIMO controller, composed of *n* subsystems nested lower triangular form and dead-zone nonlinearly inputs in non-symmetric nonlinear form was shown in [19]. Fuzzy logic systems were employed to approximate the unknown functions and the differential mean value theorem was used to separate dead-zone inputs herein [19].

It is obvious that learning capability is an influential factor to characterize efficient fuzzy rule based systems. Inference parameters include the central tendency and dispersion of the input and output fuzzy membership functions, the rule base, the cardinality of the fuzzy membership function sets, the shapes of the membership functions and the parameters of the fuzzy AND and OR operations [2]. Liu et al. [17, 44] designed controlled systems which are in a strict-feedback frame and contain unknown functions and non-symmetric dead-zone. A reinforcement learning algorithm based on the utility functions, the critic designs, and the back-stepping technique was used to develop an optimal

control signal [17]. Likewise, evolutionary algorithms such as genetic algorithm, particle swarm optimization, differential evolution and bees algorithm have been used in [13, 15, 28]. Iman, Reza & Yashar [13] combined the Sugeno fuzzy inference system and glow worms algorithm for the diagnosis of diabetes. Khoobipour & Khaleghi [15] compared the capability of four evolutionary algorithms including genetic algorithm, particle swarm optimization, differential evolution and bees algorithms to improve the capability of a new strength fuzzy inference system for nonlinear function approximation. Rong, Huang & Liang [28] evaluated the learning ability of the batch version of Online Sequential Fuzzy Extreme Learning algorithm to train a class of fuzzy inference systems which cannot be represented by the Radial Basis Function networks. Fuzzy logic systems for the optimal tracking control problem were used to approximate the long-term utility function with the support of direct heuristic dynamic programming (DHDP) setting [12] and the fuzzy-neural networks and the back-stepping design technique [8, 16].

Despite having reasonable results in comparison with FIS, those relevant researches should be intensified on advanced fuzzy sets to achieve better performance [23]. Castillo, Martínez-Marroquín, Melin, Valdez & Soria [6] compared several bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Maldonado, Castillo & Melin [21] used particle swarm optimization for average approximation of interval type-2 FIS. Melin [22, 23] implemented a new type-2 FIS method, which is a FIS on type-2 fuzzy set, for the detection of edges. It is indeed that extending FIS on advanced fuzzy sets has grasped a great attention [6, 22, 23]. Recently, a generalized fuzzy set namely *picture fuzzy set* (PFS) has been proposed in [9]. It is a generalization of fuzzy set (FS) of Zadeh [47] and intuitionistic fuzzy set (IFS) of Atanassov [3] with the debut of the positive, the negative, the neutral and the refusal degrees showing various possibilities of an element to a given set. PFS has a variety of applications in real contexts such as the confidence voting and personnel selection. Deploying fuzzy rule-based systems and soft computing methods on PFS would result in better accuracy [33]. Some preliminary researches on the soft computing methods on PFS have clearly demonstrated the usefulness of PFS in the modeling and performance improvement over traditional fuzzy tools [32–42, 46]. Thus, our objective in this research is to extend FIS on PFS in order to achieve better accuracy.

In this paper, a novel fuzzy inference system on PFS called *picture inference system* (PIS). In PIS, the positive, the neutral and the negative degrees of the picture fuzzy set are computed using the membership graph that is the combination of three Gaussian functions with a common center and different widths to express a visual view of degrees.

Then, the positive and negative defuzzification values, synthesized from three degrees of the picture fuzzy set, are used to generate crisp outputs. Learning for PIS including training centers, widths, scales and defuzzification parameters is also discussed to build up a well-approximated model. The proposed method is empirically validated on the benchmark UCI Machine Learning Repository datasets [4].

The rests of the paper are organized as follows. Section 2 introduces the preliminary knowledge including the descriptions about the types of FIS and basic notions of PFS. Section 3 presents the new picture inference system including the design and the learning method. Section 4 shows the validation on both the Lorenz system and the Housing dataset. Section 5 gives the conclusions and delineates further works.

## 2 Preliminary

In this section, some types of FIS including the Mamdani, the Sugeno and the Tsukamoto fuzzy inferences are described in Section 2.1. Section 2.3 introduces the basic notions of PFS.

### 2.1 Types of fuzzy inference systems

There are three types of fuzzy inference systems [30] such as:

- Mamdani fuzzy inference,
- Sugeno (or Takagi-Sugeno) fuzzy inference,
- Tsukamoto fuzzy inference.

A Mamdani fuzzy inference consists of two inputs  $x$  and  $y$  and a single output  $z$ . Each input  $x, y$  and output  $z$  has  $N, M$

and  $L$  membership functions, respectively. The system has  $R$  rules in the form:

$$k: \text{If } x \text{ is } A_i^{(k)} \text{ and } y \text{ is } B_j^{(k)} \text{ then } z \text{ is } C_l^{(k)} \tag{1}$$

where  $k = 1..R, i = 1..N, j = 1..M$  and  $l = 1..L$ .  $N, M$  and  $L$  are the numbers of membership functions for inputs and output, respectively.  $N, M, L$  can take any value depending on the model we construct, in this example:  $N=M=L=2$ . In this system, max/min is the most common rule of composition and the centre of method is used for defuzzification.

A Sugeno fuzzy inference has  $R$  rules in the form:

$$k: \text{If } x \text{ is } A_i^{(k)} \text{ and } y \text{ is } B_j^{(k)} \text{ then } z^{(k)} = f(x,y) \tag{2}$$

where  $k = 1..R, i = 1..N$  and  $j = 1..M$ .  $N$  and  $M$  are the numbers of membership functions for inputs. This system uses the weighted average operator or the weighted sum operator for defuzzification.

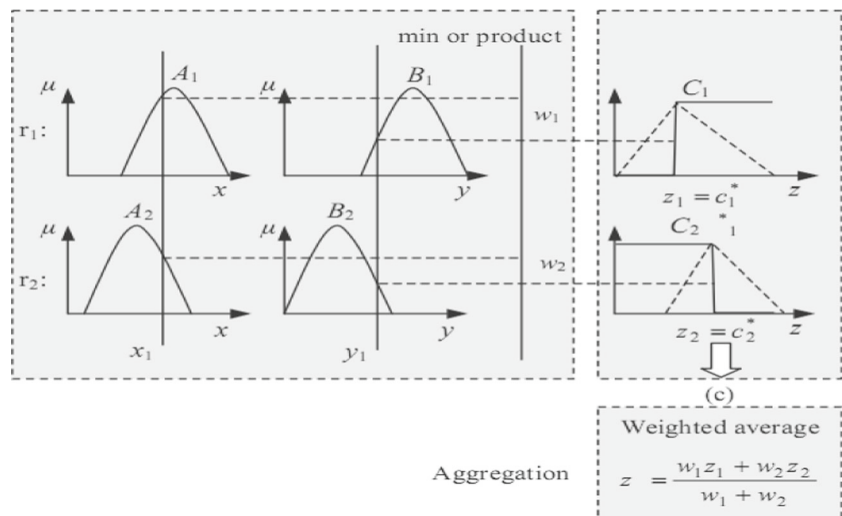
In the Tsukamoto fuzzy inference, the consequent of each fuzzy if-then rule is represented by a monotonic membership function.

$$k: \text{If } x \text{ is } A_i^{(k)} \text{ and } y \text{ is } B_j^{(k)} \text{ then } z \text{ is } C_l^{(k)}, \tag{3}$$

The Tsukamoto fuzzy model aggregates each rule's output by the method of weighted averages. Figure 1 illustrates a zero-order Sugeno inference.

Several researchers have compared the performances of those FIS systems in [10, 11, 29]. The results demonstrated the performance comparison of the three systems and the advantages of using Sugeno-type over Mamdani-type. Moreover, in fuzzy controllers, the root sum square inference engine is one of the most promising strategies and has better performance over the max-product and the max-min.

**Fig. 1** Zero-order Sugeno fuzzy inference [30]



### 2.2 Picture fuzzy sets

**Definition 1** A picture fuzzy set (PFS) [9] in a non-empty set  $X$  is,

$$A = \{ \langle x, \mu_A(x), \eta_A(x), \gamma_A(x) \mid x \in X \rangle, \tag{4}$$

where  $\mu_A(x)$  is the positive degree of each element  $x \in X$ ,  $\eta_A(x)$  is the neutral degree and  $\gamma_A(x)$  is the negative degree satisfying the constraints,

$$\mu_A(x), \eta_A(x), \gamma_A(x) \in [0, 1], \quad \forall x \in X, \tag{5}$$

$$0 \leq \mu_A(x) + \eta_A(x) + \gamma_A(x) \leq 1, \quad \forall x \in X. \tag{6}$$

The refusal degree of an element is calculated as  $\xi_A(x) = 1 - (\mu_A(x) + \eta_A(x) + \gamma_A(x))$ ,  $\forall x \in X$ . In cases  $\xi_A(x) = 0$  PFS returns to the traditional IFS set. It is obvious that PFS is an extension of IFS where the refusal degree is appended to the definition.

*Example 1* [36, 38, 39]. In a democratic election station, the council issues 500 voting papers for a candidate. The voting results are divided into four groups accompanied with the number of papers that are “vote for” (300), “abstain” (64), “vote against” (115) and “refusal of voting” (21). Group “abstain” means that the voting paper is a white paper rejecting both “agree” and “disagree” for the candidate but still takes the vote. Group “refusal of voting” is either invalid

voting papers or did not take the vote. This example happened in reality and IFS could not handle it since the refusal degree (group “refusal of voting”) does not exist.

Now, some basic picture fuzzy operations, picture distance metrics and picture fuzzy relations are briefly presented [9]. Let  $PFS(X)$  denote the set of all PFS sets on universe  $X$ .

**Definition 2** For  $A, B \in PFS(X)$ , the union, intersection, complement and inclusion operations are defined as follows.

$$A \cup B = \{ \langle x, \max \{ \mu_A(x), \mu_B(x) \}, \min \{ \eta_A(x), \eta_B(x) \}, \min \{ \gamma_A(x), \gamma_B(x) \} \mid x \in X \rangle, \tag{7}$$

$$A \cap B = \{ \langle x, \min \{ \mu_A(x), \mu_B(x) \}, \min \{ \eta_A(x), \eta_B(x) \}, \max \{ \gamma_A(x), \gamma_B(x) \} \mid x \in X \rangle, \tag{8}$$

$$\bar{A} = \{ \langle x, \gamma_A(x), \eta_A(x), \mu_A(x) \mid x \in X \rangle, \tag{9}$$

$$A \subseteq B \text{ iff } \forall x \in X : \mu_A(x) \leq \mu_B(x) \text{ and } \eta_A(x) \leq \eta_B(x) \text{ and } \gamma_A(x) \geq \gamma_B(x), \tag{10}$$

$$A = B \text{ iff } A \subseteq B \text{ and } B \subseteq A. \tag{11}$$

**Definition 3** For  $A, B \in PFS(X)$ , some operators on PFS are:

$$A + B = \{ \langle x, \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x), \eta_A(x) \cdot \eta_B(x), v_A(x) \cdot v_B(x) \mid x \in X \rangle \tag{12}$$

$$A \cdot B = \{ \langle x, \mu_A(x) \cdot \mu_B(x), \eta_A(x) \cdot \eta_B(x), v_A(x) + v_B(x) - v_A(x) \cdot v_B(x) \mid x \in X \rangle \tag{13}$$

$$A @ B = \left\{ \left\langle x, \frac{1}{2} (\mu_A(x) + \mu_B(x)), \frac{1}{2} (\eta_A(x) + \eta_B(x)), \frac{1}{2} (v_A(x) + v_B(x)) \right\rangle \mid x \in X \right\} \tag{14}$$

$$A \$ B = \left\{ \left\langle x, \sqrt{\mu_A(x) \cdot \mu_B(x)}, \sqrt{\eta_A(x) \cdot \eta_B(x)}, \sqrt{v_A(x) \cdot v_B(x)} \right\rangle \mid x \in X \right\} \tag{15}$$

$$A \# B = \left\{ \left\langle x, \frac{2\mu_A(x) \cdot \mu_B(x)}{\mu_A(x) + \mu_B(x)}, \frac{2\eta_A(x) \cdot \eta_B(x)}{\eta_A(x) + \eta_B(x)}, \frac{2v_A(x) \cdot v_B(x)}{v_A(x) + v_B(x)} \right\rangle \mid x \in X \right\} \tag{16}$$

$$A * B = \left\{ \left\langle x, \frac{\mu_A(x) + \mu_B(x)}{2(\mu_A(x) \cdot \mu_B(x) + 1)}, \frac{\eta_A(x) + \eta_B(x)}{2(\eta_A(x) \cdot \eta_B(x) + 1)}, \frac{v_A(x) + v_B(x)}{2(v_A(x) \cdot v_B(x) + 1)} \right\rangle \mid x \in X \right\} \tag{17}$$

*Remark 1* For a scalar  $c$  and a PFS  $A$ , the  $c \times A$  is performed according to (12), for instance  $c = 2$ :

$$\begin{aligned} 2 \times A &= 2.A \\ &= A + A \\ &= \left\{ \left\langle x, 2\mu_A(x) - \mu_A^2(x), \eta_A^2(x), v_A^2(x) \right\rangle \mid x \in X \right\} \end{aligned} \tag{18}$$

**Definition 4** For  $A, B \in PFS(X)$ , the Cartesian product of these PFS sets is,

$$A \times_1 B = \{ \langle (x, y), \mu_A(x) \cdot \mu_B(y), \eta_A(x) \cdot \eta_B(y), \gamma_A(x) \cdot \gamma_B(y) \mid x \in A, y \in B \rangle, \tag{19}$$

$$A \times_2 B = \{ \langle (x, y), \mu_A(x) \wedge \mu_B(y), \eta_A(x) \wedge \eta_B(y), \gamma_A(x) \vee \gamma_B(y) \mid x \in A, y \in B \rangle. \tag{20}$$

**Definition 5** The distances between  $A, B \in PFS(X)$  are the normalized Hamming distance and the normalized Euclidean in formulae (21–22), respectively.

$$d_p(A, B) = \frac{1}{N} \sum_{i=1}^N (|\mu_A(x_i) - \mu_B(x_i)| + |\eta_A(x_i) - \eta_B(x_i)| + |\gamma_A(x_i) - \gamma_B(x_i)|), \tag{21}$$

$$e_p(A, B) = \sqrt{\frac{1}{N} \sum_{i=1}^N ((\mu_A(x_i) - \mu_B(x_i))^2 + (\eta_A(x_i) - \eta_B(x_i))^2 + (\gamma_A(x_i) - \gamma_B(x_i))^2)}. \tag{22}$$

**Definition 6** The picture fuzzy relation  $R$  is a picture fuzzy subset of  $A \times B$ , given by

$$R = \{ \{(x, y), \mu_R(x, y), \eta_R(x, y), \gamma_R(x, y)\} \mid x \in A, y \in B \}, \tag{23}$$

$$\mu_R, \eta_R, \gamma_R : A \times B \rightarrow [0, 1], \tag{24}$$

$$\mu_R(x, y) + \eta_R(x, y) + \gamma_R(x, y) \leq 1, \tag{25}$$

$$\forall (x, y) \in A \times B.$$

$PFR(A \times B)$  is the set of all picture fuzzy subset on  $A \times B$ .

**Definition 7** (Zadeh extension principle for PFS) For  $i = 1, 2, \dots, n, U_i$  is a universe and  $V \neq \emptyset$ . Let  $f : U_1 \times_2 \dots \times_2 U_n \rightarrow V$  be a mapping, where  $y = f(z_1, \dots, z_n)$ . Let  $z_i$  is a linguistic variable on  $U_i$  for  $i = 1, 2, \dots, n$ . Suppose, for all  $i, z_i$  is  $A_i$  where  $A_i$  is a PFS on  $U_i$ . Then, the output of the mapping  $f$  is  $B$ , which is a PFS on  $V$  defined for  $\forall y \in V$  by,

$$B(y) = \left\{ \begin{array}{l} \left( \bigvee_{D(y)} \left( \bigwedge_{i=1}^n \mu_{A_i}(u_i) \right), \bigwedge_{D(y)} \left( \bigwedge_{i=1}^n \eta_{A_i}(u_i) \right), \bigwedge_{D(y)} \left( \bigvee_{i=1}^n \nu_{A_i}(u_i) \right) \right) \text{ if } f^{-1}(y) \neq \emptyset \\ (0, 0, 0) \text{ if } f^{-1}(y) = \emptyset \end{array} \right\} \tag{26}$$

$$D(y) = f^{-1}(y) = \{u = (u_1, \dots, u_n) : f(u) = y\}. \tag{27}$$

*Remark 2* a) For some PFSs-  $A_1, \dots, A_n$  and a function-  $f$ , the positive degree, neutral degree and negative degree of the PFS-  $f(A_1, \dots, A_n)$  are

$$\mu_B(y) = \begin{cases} \bigvee_{D(y)} \left( \bigwedge_{i=1}^n \mu_{A_i}(u_i) \right) & \text{if } f^{-1}(y) \neq \emptyset \\ 0 & \text{if } f^{-1}(y) = \emptyset \end{cases},$$

$$\eta_B(y) = \begin{cases} \bigwedge_{D(y)} \left( \bigwedge_{i=1}^n \eta_{A_i}(u_i) \right) & \text{if } f^{-1}(y) \neq \emptyset \\ 0 & \text{if } f^{-1}(y) = \emptyset \end{cases},$$

$$\nu_B(y) = \begin{cases} \bigwedge_{D(y)} \left( \bigvee_{i=1}^n \nu_{A_i}(u_i) \right) & \text{if } f^{-1}(y) \neq \emptyset \\ 1 & \text{if } f^{-1}(y) = \emptyset \end{cases},$$

where  $D(y) = f^{-1}(y) = \{u = (u_1, \dots, u_n) : f(u) = y\}$ .

b) It is obvious that the product operations of the PFSs in Definition 4 are a special case of the Zadeh extension principle for PFS in Definition 7. We will prove this remark as follows.

*Proof* Consider the Cartesian product in (20). For  $A, B \in PFS(X)$ , assume that  $U_1 = U_2 = X, z_1 = A, z_2 = B$  and  $f$  is a bijective then,

$$\begin{aligned} \mu_{f(A,B)}(y) &= \mu_{A \times_2 B}(y) \\ &= \begin{cases} \bigvee_{D(y)} (\mu_A(u_1) \wedge \mu_B(u_2)) & \text{if } f^{-1}(y) \neq \emptyset \\ 0 & \text{if } f^{-1}(y) = \emptyset \end{cases} \\ &= \mu_A(u_1) \wedge \mu_B(u_2), \end{aligned}$$

$$\nu_{f(A,B)}(y) = \nu_{A \times_2 B}(y) = \gamma_A(u_1) \vee \gamma_B(u_2),$$

$$\eta_{f(A,B)}(y) = \eta_{A \times_2 B}(y) = \eta_A(u_1) \bigwedge \eta_B(u_2),$$

where  $D(y) = f^{-1}(y) = \{u = (u_1, u_2) : f(u) = y\}$ . Thus,  $A \times_2 B$  is a special case of the Zadeh extension principle for PFS in Definition 7. Similarly, for  $A, B \in PFS(X)$ ,  $A \times_1 B$  is a special case of the Zadeh extension principle for PFS in Definition 7 with the product t-norm instead of the minimum t-norm.  $\square$

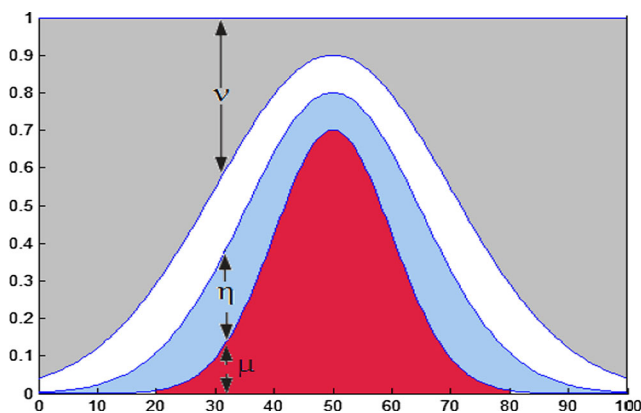
### 3 Picture inference system

Picture inference system including design and learning phase is discussed in this section. The design of PIS is given in Section 3.1. Section 3.2 shows the learning method for PIS.

#### 3.1 PFS design

According to Definition 1, PFS constitutes of three degrees of positive, neutral and negative memberships simulating different states of human’s feeling such as agree, neutral and disagree. Using the three degrees, it is convenient to estimate and make better approximation for FIS. In this section, a novel fuzzy inference system on PFS called *picture inference system* (PIS) including the design based on the membership graph and the general picture inference scheme is proposed. The advantage of the new system is that the set of rules can be reused without any change. The rule set is exactly the same, but the inference is more complex and requires more effort to deal with other degrees. Each rule generates a firing degree with three equivalent values namely positive, neutral and negative. There is no clear method to combine them or to use them for defuzzification or aggregation step so that in this section those steps are invoked to gain the crisp value.

In a classic fuzzy system, each linguistic variable is represented by a membership function that generates a membership degree when the corresponding input variable is given a crisp value. In the new model, to be able to gain three values as mentioned above, a *membership graph* that is the combination of three lines giving a visual view of each degree is used. Figure 2 shows the membership graph where the lowest, the medium and the highest lines respectively expressed the positive, neutral and negative degree. For simplicity, it is assumed that these lines use the same form of function. In the other words, the Gaussian function with a common center and different widths is used for the lines. There are some reasons for us to construct such the graph. In



**Fig. 2** The membership graph of PFS where  $(\mu, \eta, \nu)$  are the positive, neutral and negative degrees

real life as in Example 1, the graph could be a good approximation for elections where the number of people who vote for is the largest at the center and the number of people who abstain and vote against are quite low. In weather now-casting, the membership graph is useful to demonstrate the main direction and the right and left expansion. Herein, the orientations in next hours are dependent on both the main direction and expansions of the current state with equivalent probabilities. In two sides of the center, the number of people who vote for decreases significantly, the number of people who abstain increases slightly and then decreases, and the number of people who vote against only increases as the opposite idea is getting sharp. Visually, the red area in Fig. 2 represents a normal membership degree.

The formulae for the positive, neutral and negative degrees are expressed in (28–30) respectively.

$$\mu_A(x) = \exp\left(-\frac{(x - c)^2}{2\sigma_\mu^2}\right), \tag{28}$$

$$\eta_A(x) = \exp\left(-\frac{(x - c)^2}{2\sigma_\eta^2}\right) - \mu_A(x), \tag{29}$$

$$\nu_A(x) = 1 - \exp\left(-\frac{(x - c)^2}{2\sigma_\nu^2}\right), \tag{30}$$

where  $c$  is the center of all three functions,  $\sigma_\mu$  (resp.  $\sigma_\eta, \sigma_\nu$ ) is the width value (i.e., the standard deviation) of positive (resp. neutral, negative) function. From now on, for shorter denotation, the membership graph is denoted as set three functions -  $(f(x), g(x), h(x))$  that describe three lines in the graph. It is clear that

$$\mu_A(x) = f_A(x), \tag{31}$$

$$\eta_A(x) = g_A(x) - f_A(x), \tag{32}$$

$$\nu_A(x) = 1 - g_A(x). \tag{33}$$

The *general picture inference scheme* is described in Fig. 3

Some steps are drawn as follows.

*Step1:* Compute the positive, the neutral and the negative degrees using the membership graph.

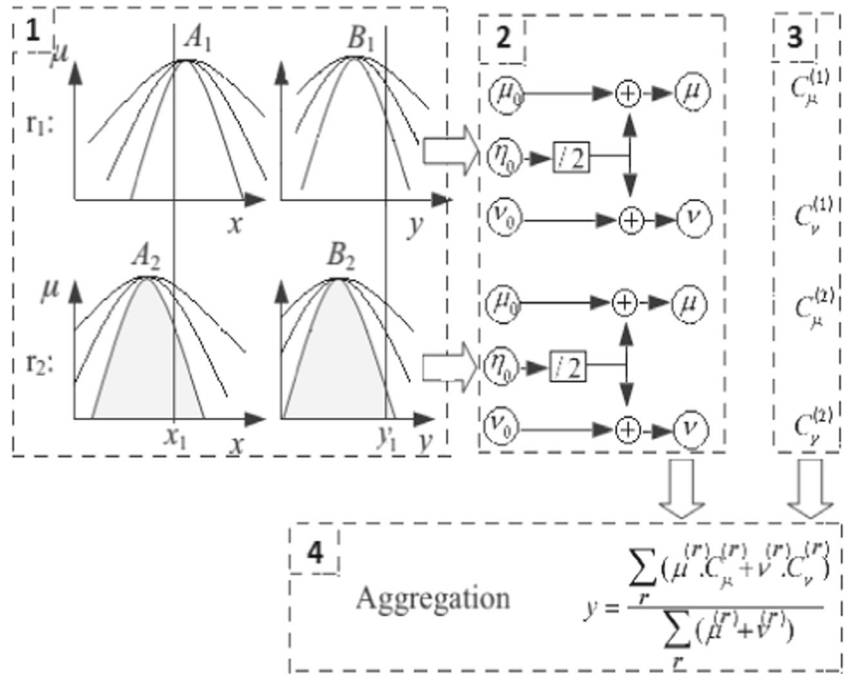
Obviously, with  $n$  crisp values of one vector input, each rule generates  $n$  sets of three degrees. To extract the firing strength of each rule, the following formulae (34–36) are used.

$$\mu_0^{(r)} = \min_j(\mu_j^{(r)}(x)), \tag{34}$$

$$\eta_0^{(r)} = \min_j(\eta_j^{(r)}(x)), \tag{35}$$

$$\nu_0^{(r)} = \max_j(\nu_j^{(r)}(x)). \tag{36}$$

**Fig. 3** General picture inference scheme where (1.4) refer to steps



In these formulae,  $r$  is the index of rule and  $j$  is the index of variable. Remember that the operations of three formulae are presented in [9], which guarantee the property (6).

*Step 2: Distribute the neutral degree to the positive and negative degrees.*

After extracting the firing strength, it is required to defuzzify them into a single crisp value. People who abstain do not either vote for or vote against; hence they are divided into half: one for the group of people who vote for, one for the group of people who vote again. As apply the idea in our model, we gain:

$$v^{(r)} = v_0^{(r)} + \frac{\eta_0^{(r)}}{2}, \tag{37}$$

$$\mu^{(r)} = \mu_0^{(r)} + \frac{\eta_0^{(r)}}{2}. \tag{38}$$

In formulae (37–38),  $\mu^{(r)}, v^{(r)}$  are the final positive and negative degrees. Next, those values are mixed and defuzzified by formula (39), which in essence is the weighted average method.

$$y^{(r)} = \frac{\mu^{(r)} \cdot C_\mu^{(r)} + \nu^{(r)} \cdot C_\nu^{(r)}}{\mu^{(r)} + \nu^{(r)}}. \tag{39}$$

*Step 3: Find the positive and negative defuzzification values.*

In Fig. 3,  $C_\mu^{(r)}, C_\nu^{(r)}$  are the defuzzification values associated with rule  $r$ -th. In the Mamdani model,  $C_\mu^{(r)}, C_\nu^{(r)}$  are constants; in the Sugeno model, they can be constants or the values computed from defuzzification. Denote  $C_\mu^r (C_\nu^r)$  is the positive (negative) defuzzification value. It is clear

that the power of fuzzy systems is from the parameters that make up the systems, but fuzzy systems are affected to the initial values of parameters. Therefore, picking up suitable defuzzification values should be taken into consideration. The process is presented for all architectures as follows.

- In the Mamdani model:*

$$C_\nu^{(r)} = \frac{\sum_{p, Label(p) \neq Label(r)} C_\mu^{(p)}}{L - 1}, \tag{40}$$

where  $L$  is the number of labels of the output variable. Formula (40) means that  $C_\nu^{(r)}$  is set to the average of other  $C_\mu^{(p)}$  having different labels.

- In the Sugeno model:*

The recipe can be reused if many rules share the same label of defuzzification. If each rule has its own function, the following strategy is applied.

$$f^{(r)}(x) = \frac{\sum_{p \neq r} f^{(p)}(x)}{R - 1}. \tag{41}$$

Again, the upper indices  $r$  and  $p$  are indexed of a rule. The positive and negative defuzzification values are calculated using defuzzification function. Picking up initial defuzzification parameters is not simple; fortunately, they can be adjusted easily in the learning phase.

- In the Tsukamoto model:*

The negative defuzzification functions are not needed to find because the classic model often consists of two opposite monotonic functions with respect to two terms of the

linguistic output variable. For each rule, we can find the negative defuzzification value using the opposite function to the function of the current rule.

*Step 4: Aggregate to find the crisp value of output.*

What we have now is the crisp value for one rule only. To mix all these crisp values, the following formulae are used.

$$y = \frac{\sum_r y^{(r)}(\mu^{(r)} + \nu^{(r)})}{\sum_r (\mu^{(r)} + \nu^{(r)})}. \tag{42}$$

Rewrite (42) as,

$$y = \frac{\sum_r (\mu^{(r)} \cdot C_\mu^{(r)} + \nu^{(r)} \cdot C_\nu^{(r)})}{\sum_r (\mu^{(r)} + \nu^{(r)})}. \tag{43}$$

### 3.2 Learning phase

In this section, the learning phase in PIS is discussed. Learning is an important process to adjust parameters to build up a well-approximated model. There are existing methods for learning in FIS such as the back-propagation, which is a powerful tool to train. However in our case, it is possible that constrain (6) is violated throughout the learning process. In Section 3.1, we use the membership graph with the following conditions being held to guarantee constraint (6).

$$\sigma_\mu \leq \sigma_\eta \leq \sigma_\nu, \tag{44}$$

where  $\sigma_\mu, \sigma_\eta, \sigma_\nu$  are the width values of positive, neutral, negative functions in formulae (28–30), respectively. These values follow the rule in formulae (45–46) where  $|\alpha|, |\beta|$  are absolute values of numbers.

$$\sigma_\eta = \sigma_\mu(1 + |\alpha|), \tag{45}$$

$$\sigma_\nu = \sigma_\eta(1 + |\beta|). \tag{46}$$

It seems that the requirement is met but the system may not run smoothly since the derivatives in back-propagation must be computed. Thus, smooth functions like square values in formulae (47–48) should be used instead of absolute values.

$$\sigma_\eta = \sigma_\mu(1 + \alpha^2), \tag{47}$$

$$\sigma_\nu = \sigma_\eta(1 + \beta^2). \tag{48}$$

Finally, in learning process, some parameters such as the centers  $c$ , the width  $\mu$  of positive degree and the scales  $\alpha$  and  $\beta$  for each linguistic term have to be optimized. We now present here two main steps and formulae of learning. If denoting the index of input vector as  $i$ , the index of rule as  $r$ , the index of input variable as  $j$  and the input of membership function as  $k$  then  $\mu_{jk}^{(ir)}$  is the positive degree of variable  $j$

in input vector  $i$  with the rule-  $r$  and the index of membership function  $-k$ . Note that  $k$  is determined through  $r$  and  $j$  as in the formula (49).

$$k = R(r, j). \tag{49}$$

$R$  is a matrix that represents the rule set of the dataset implying relationship between three indices  $r, j, k$ .

#### 3.2.1 Phase 1: learning centers, widths and scales

We start the learning phase with the definition of error of each input vector.

$$e_i = y_d^{(i)} - y^{(i)}, \tag{50}$$

where  $y_d^{(i)}$  is the desired output and  $y^{(i)}$  is the output calculated from system. The objective function is,

$$\bar{E} = \frac{1}{2N} \sum_i e_i^2. \tag{51}$$

$N$  is the number of input vectors. Taking the derivative of (51), we have

$$\frac{\partial \bar{E}}{\partial y^{(i)}} = \frac{-e_i}{N}. \tag{52}$$

The partial derivatives of  $y^{(i)}$  with respect to  $\mu^{(ir)}$  and  $\nu^{(ir)}$  are,

$$\begin{aligned} \frac{\partial y^{(i)}}{\partial \mu^{(ir)}} &= \frac{\frac{\sum_r (\mu^{(ir)} \cdot C_\mu^{(ir)} + \nu^{(ir)} \cdot C_\nu^{(ir)})}{\sum_r (\mu^{(ir)} + \nu^{(ir)})}}{\partial \mu^{(ir)}} \\ &= \frac{C_\mu^{(ir)} \sum_r (\mu^{(ir)} + \nu^{(ir)}) - \sum_r (\mu^{(ir)} \cdot C_\mu^{(ir)} + \nu^{(ir)} \cdot C_\nu^{(ir)})}{\left(\sum_r (\mu^{(ir)} + \nu^{(ir)})\right)^2} \\ &= \frac{\frac{C_\mu^{(ir)} \sum_r (\mu^{(ir)} + \nu^{(ir)})}{\sum_r (\mu^{(ir)} + \nu^{(ir)})} - \frac{\sum_r (\mu^{(ir)} \cdot C_\mu^{(ir)} + \nu^{(ir)} \cdot C_\nu^{(ir)})}{\sum_r (\mu^{(ir)} + \nu^{(ir)})}}{\sum_r (\mu^{(ir)} + \nu^{(ir)})} \\ &= \frac{C_\mu^{(ir)} - y^{(r)}}{\sum_r (\mu^{(ir)} + \nu^{(ir)})}, \end{aligned} \tag{53}$$

Analogously:

$$\frac{\partial y^{(i)}}{\partial \nu^{(ir)}} = \frac{y^{(r)} - C_\nu^{(ir)}}{\sum_r (\mu^{(ir)} + \nu^{(ir)})}. \tag{54}$$

From formulae (37–38), it follows that

$$\frac{\partial \mu^{(ir)}}{\partial \mu_0^{(ir)}} = 1, \tag{55}$$

$$\frac{\partial \nu^{(ir)}}{\partial \nu_0^{(ir)}} = 1. \tag{56}$$



From (53–56), we have

$$\begin{aligned} \frac{\partial \bar{E}}{\partial \mu_0^{(ir)}} &= \frac{\partial \bar{E}}{\partial y^{(i)}} \frac{\partial y^{(i)}}{\partial \mu^{(ir)}} \frac{\partial \mu^{(ir)}}{\partial \mu_0^{(ir)}} \\ &= \frac{-e_i}{N} \frac{y^{(r)} - C_\mu^{(ir)}}{\sum_r \mu^{(ir)} + v^{(ir)}}, \end{aligned} \tag{57}$$

Analogously

$$\frac{\partial \bar{E}}{\partial v_0^{(ir)}} = \frac{-e_i}{N} \frac{C_v^{(ir)} - y^{(r)}}{\sum_r \mu^{(ir)} + v^{(ir)}}. \tag{58}$$

From formulae (37–38), we obtain

$$\frac{\partial \bar{E}}{\partial \eta_0^{(ir)}} = \frac{1}{2} \left( \frac{\partial \bar{E}}{\partial \mu_0^{(ir)}} + \frac{\partial \bar{E}}{\partial v_0^{(ir)}} \right). \tag{59}$$

Note that

$$\frac{\partial \mu_0^{(ir)}}{\partial \mu_{jk}^{(ir)}} = \begin{cases} 1, & \text{if } \mu_0^{(ir)} = \min_j(\mu_{jk}^{(ir)}) \\ 0, & \text{otherwise} \end{cases}, \tag{60}$$

$$\frac{\partial \eta_0^{(ir)}}{\partial \eta_{jk}^{(ir)}} = \begin{cases} 1, & \text{if } \eta_0^{(ir)} = \min_j(\eta_{jk}^{(ir)}) \\ 0, & \text{otherwise} \end{cases} \tag{61}$$

$$\frac{\partial v_0^{(ir)}}{\partial v_{jk}^{(ir)}} = \begin{cases} 1, & \text{if } v_0^{(ir)} = \max_j(v_{jk}^{(ir)}) \\ 0, & \text{otherwise} \end{cases}. \tag{62}$$

The partial derivatives with respect to center  $c_{jk}$  are expressed in formulae (63–65). Note that the Gaussian functions are used where  $\sigma_{jk}$  represents for the width of positive function.

$$\begin{aligned} \frac{\partial \mu_{jk}^{(ir)}}{\partial c_{jk}} &= \exp\left(-\frac{(x_j^{(i)} - c_{jk})^2}{2\sigma_{jk}^2}\right) \frac{(x_j^{(i)} - c_{jk})}{\sigma_{jk}^2} \\ &= \mu_{jk}^{(ir)} \frac{(x_j^{(i)} - c_{jk})}{\sigma_{jk}^2}, \end{aligned} \tag{63}$$

$$\begin{aligned} \frac{\partial \eta_{jk}^{(ir)}}{\partial c_{jk}} &= \frac{\partial \left( \exp\left(\frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \alpha_{jk}^2)\sigma_{jk})^2}\right) - \exp\left(-\frac{(x_j^{(i)} - c_{jk})^2}{2\sigma_{jk}^2}\right) \right)}{\partial c_{jk}} \\ &= \exp\left(\frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \alpha_{jk}^2)\sigma_{jk})^2}\right) \frac{(x_j^{(i)} - c_{jk})}{((1 + \alpha_{jk}^2)\sigma_{jk})^2} \\ &\quad - \frac{\partial \mu_{jk}^{(ir)}}{\partial c_{jk}}, \end{aligned} \tag{64}$$

$$\begin{aligned} \frac{\partial v_{jk}^{(ir)}}{\partial c_{jk}} &= \frac{\partial \left( 1 - \exp\left(\frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2}\right) \right)}{\partial c_{jk}} \\ &= -\exp\left(\frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2}\right) \\ &\quad \frac{(x_j^{(i)} - c_{jk})}{((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2}. \end{aligned} \tag{65}$$

The upgrade scheme for center  $c_{jk}$  is,

$$\begin{aligned} \Delta c_{jk} &= -\eta \frac{\partial \bar{E}}{\partial c_{jk}} \\ &= -\eta \left( \sum_{i,r} \frac{\partial \bar{E}}{\partial \mu_0^{(ir)}} \frac{\partial \mu_0^{(ir)}}{\partial c_{jk}} + \sum_{i,r} \frac{\partial \bar{E}}{\partial \eta_0^{(ir)}} \frac{\partial \eta_0^{(ir)}}{\partial c_{jk}} \right. \\ &\quad \left. + \sum_{i,r} \frac{\partial \bar{E}}{\partial v_0^{(ir)}} \frac{\partial v_0^{(ir)}}{\partial c_{jk}} \right) \\ &= -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial \mu_0^{(ir)}} \frac{\partial \mu_0^{(ir)}}{\partial \mu_{jk}^{(ir)}} \frac{\partial \mu_{jk}^{(ir)}}{\partial c_{jk}} \\ &\quad + -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial \eta_0^{(ir)}} \frac{\partial \eta_0^{(ir)}}{\partial \eta_{jk}^{(ir)}} \frac{\partial \eta_{jk}^{(ir)}}{\partial c_{jk}} \\ &\quad + -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial v_0^{(ir)}} \frac{\partial v_0^{(ir)}}{\partial v_{jk}^{(ir)}} \frac{\partial v_{jk}^{(ir)}}{\partial c_{jk}} \end{aligned} \tag{66}$$

where  $\eta$  is learning rate. Similarly, the partial derivatives of three types of degrees with respect to width  $\sigma_{jk}$  are,

$$\begin{aligned} \frac{\partial \mu_{jk}^{(ir)}}{\partial \sigma_{jk}} &= \frac{\partial \exp\left(-\frac{(x_j^{(i)} - c_{jk})^2}{2\sigma_{jk}^2}\right)}{\partial \sigma_{jk}} \\ &= \exp\left(-\frac{(x_j^{(i)} - c_{jk})^2}{2\sigma_{jk}^2}\right) \cdot \frac{(x_j^{(i)} - c_{jk})^2}{\sigma_{jk}^3} \\ &= \mu_{jk}^{(ir)} \frac{(x_j^{(i)} - c_{jk})^2}{\sigma_{jk}^3}, \end{aligned} \tag{67}$$

$$\begin{aligned} \frac{\partial \eta_{jk}^{(ir)}}{\partial \sigma_{jk}} &= \frac{\partial \left( \exp\left(\frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \alpha_{jk}^2)\sigma_{jk})^2}\right) - \exp\left(-\frac{(x_j^{(i)} - c_{jk})^2}{2\sigma_{jk}^2}\right) \right)}{\partial \sigma_{jk}} \\ &= \exp\left(\frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \alpha_{jk}^2)\sigma_{jk})^2}\right) \frac{(x_j^{(i)} - c_{jk})^2}{((1 + \alpha_{jk}^2)\sigma_{jk})^3} \\ &\quad (1 + \alpha_{jk}^2) - \frac{\partial \mu_{jk}^{(ir)}}{\partial \sigma_{jk}}, \end{aligned} \tag{68}$$

$$\begin{aligned} \frac{\partial v_{jk}^{(ir)}}{\partial \sigma_{jk}} &= \frac{\partial \left( 1 - \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) \right)}{\partial \sigma_{jk}} \\ &= - \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) \\ &\quad \frac{(x_j^{(i)} - c_{jk})^2}{((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^3} (1 + \alpha_{jk}^2)(1 + \beta_{jk}^2). \end{aligned} \tag{69}$$

Similar to formula (62), the upgrade scheme for width  $\sigma_{jk}$  is,

$$\begin{aligned} \Delta \sigma_{jk} &= -\eta \frac{\partial \bar{E}}{\partial \sigma_{jk}} = -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial \mu_0^{(ir)}} \frac{\partial \mu_0^{(ir)}}{\partial \mu_{jk}^{(ir)}} \frac{\partial \mu_{jk}^{(ir)}}{\partial \sigma_{jk}} \\ &\quad + -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial \eta_0^{(ir)}} \frac{\partial \eta_0^{(ir)}}{\partial \eta_{jk}^{(ir)}} \frac{\partial \eta_{jk}^{(ir)}}{\partial \sigma_{jk}} \\ &\quad + -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial v_0^{(ir)}} \frac{\partial v_0^{(ir)}}{\partial v_{jk}^{(ir)}} \frac{\partial v_{jk}^{(ir)}}{\partial \sigma_{jk}} \end{aligned} \tag{70}$$

The scale  $\alpha$  appears in neutral and negative degrees only so that the positive degree can be ignored in this case.

$$\begin{aligned} \frac{\partial \eta_{jk}^{(ir)}}{\alpha_{jk}} &= \frac{\partial \left( \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) - \exp \left( -\frac{(x_j^{(i)} - c_{jk})^2}{2\sigma_{jk}^2} \right) \right)}{\partial \alpha_{jk}} \\ &= \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) \frac{(x_j^{(i)} - c_{jk})^2}{((1 + \alpha_{jk}^2)\sigma_{jk})^3} \sigma_{jk} 2\alpha_{jk}, \end{aligned} \tag{71}$$

$$\begin{aligned} \frac{\partial v_{jk}^{(ir)}}{\alpha_{jk}} &= \frac{\partial \left( 1 - \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) \right)}{\partial \alpha_{jk}} \\ &= - \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) \\ &\quad \frac{(x_j^{(i)} - c_{jk})^2}{((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^3} \sigma_{jk} (1 + \beta_{jk}^2) 2\alpha_{jk}. \end{aligned} \tag{72}$$

Similarly to formula (70), the upgrade scheme of  $\alpha_{jk}$  is,

$$\begin{aligned} \Delta \alpha_{jk} &= -\eta \frac{\partial \bar{E}}{\partial \alpha_{jk}} = -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial \eta_0^{(ir)}} \frac{\partial \eta_0^{(ir)}}{\partial \eta_{jk}^{(ir)}} \frac{\partial \eta_{jk}^{(ir)}}{\partial \alpha_{jk}} \\ &\quad + -\eta \sum_{i,r} \frac{\partial \bar{E}}{\partial v_0^{(ir)}} \frac{\partial v_0^{(ir)}}{\partial v_{jk}^{(ir)}} \frac{\partial v_{jk}^{(ir)}}{\partial \alpha_{jk}} \end{aligned} \tag{73}$$

Similarly, the partial derivative with respect to  $\beta_{jk}$  is,

$$\begin{aligned} \frac{\partial v_{jk}^{(ir)}}{\beta_{jk}} &= \frac{\partial \left( 1 - \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) \right)}{\partial \beta_{jk}} \\ &= - \exp \left( \frac{-(x_j^{(i)} - c_{jk})^2}{2((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^2} \right) \\ &\quad \frac{(x_j^{(i)} - c_{jk})^2}{((1 + \beta_{jk}^2)(1 + \alpha_{jk}^2)\sigma_{jk})^3} \sigma_{jk} (1 + \alpha_{jk}^2) 2\beta_{jk}. \end{aligned} \tag{74}$$

Finally, we get the upgrade scheme for  $\beta_{jk}$  which is similar to the upgrade scheme of  $\alpha_{jk}$ .

$$\Delta \beta_{jk} = -\eta \frac{\partial \bar{E}}{\partial \beta_{jk}} = \eta \sum_{i,r} \frac{\partial \bar{E}}{\partial v_0^{(ir)}} \frac{\partial v_0^{(ir)}}{\partial v_{jk}^{(ir)}} \frac{\partial v_{jk}^{(ir)}}{\partial \beta_{jk}}. \tag{75}$$

### 3.2.2 Phase 2: Learning defuzzification parameters

It is infeasible if we ignore optimizing the positive and negative defuzzification parameters. This step is very simple, not time-consuming and quite trivial so that we do not mention here. The only thing has to be kept in mind is that optimizing can be different between the Mamdani and Sugano models in determining negative.

Being noted that initializing the parameter in FIS is much simpler than in PIS, a trick that could be used to improve the two-step learning phase above is to initialize parameters from trained FIS which means picking up a very small value of  $\alpha$  and a very large value of  $\beta$ . The following steps describe such the improvement.

*Step 1: Training positive functions.*

In this step we actually train the center and width.

*Step 2: Training defuzzification parameters corresponding to positive degrees.*

*Step 3: Training the height of each positive function.*

In FIS, each membership function has the same height value, e.g. 1. A height value is added to the membership function as follows.

$$\mu_0(x) = h_\mu \exp \left( -\frac{(x - c)^2}{2\sigma_\mu^2} \right), \tag{76}$$

where  $h_\mu$  is height of the positive function,  $c$  is center of all three functions and  $\sigma_\mu$  is width of the positive function. Each function should have its own height value, which aims to adjust the rule set and fixes the relation between rules.

*Step 4: Initializing neutral degrees.*

At first, neutral values are randomly set up and then repeated many times until a set of degrees that gives less error than that of the previous step is found. Neutral degrees do not need to be too small so that they are initialized with sizes of positive degrees. It is obvious that neutral degrees are equally distributed to positive and negative degrees so no matter how large neutral values are, positive (resp. negative) degrees still have large (resp. small) values. The neutral degree function is defined as follows,

$$\eta_0(x) = h_\eta \mu_0(x), \tag{77}$$

where  $h_\eta$  is height of neutral function.

*Step 5: train neutral degrees.*

This step trains the raw neutral degrees.

*Step 6: Initialize and train the original negative degrees.*

Since the minimum (resp. maximum) operator is used on positive (resp. negative) degrees, it is required to set up a very small value of the negative degree by some methods such as the triangular function as in formula (78).

$$v_0(x) = h_v(1 - f_\Delta(x, c, \sigma_v)), \tag{78}$$

where  $f_\Delta$  is a symmetric triangular membership function with center  $c$  and width  $\sigma_v$ .

$$f_\Delta(x, c, \sigma_v) = \begin{cases} 0 & \text{if } x < c - \sigma_v \\ \frac{x - c + \sigma_v}{\sigma_v} & \text{if } c - \sigma_v \leq x < c \\ \frac{-x + c + \sigma_v}{\sigma_v} & \text{if } c \leq x < c + \sigma_v \\ 0 & \text{if } c + \sigma_v \leq x \end{cases} \tag{79}$$

$c$  is center of the three functions and  $\sigma_v$  is width of the negative function.

*Step 7: Train the defuzzification parameter corresponding to negative degrees.*

After having all degrees, the last defuzzification parameters are trained accordingly.

By these steps, we are able to train parameters of the PIS model. Note that after the learning phase, some membership values could be negative. We have to control this by checking the learning strictly, for example setting up the absolute values for those negative memberships after each round of height learning. The problem that the sum of memberships can be greater than 1 can be fixed by normalization.

### 3.3 Remarks

Although PIS is more advantages than FIS, there are some counter-examples showing the reverse. Consider an example:

$x_1 + x_2 = y$  with  $x_1, x_2 \in [0, 10]$  and 4 fuzzy rules:

- R1: If  $x_1$  is small and  $x_2$  is small then  $y$  is small.
- R2: If  $x_1$  is small and  $x_2$  is large then  $y$  is medium.
- R3: If  $x_1$  is large and  $x_2$  is small then  $y$  is medium.
- R4: If  $x_1$  is large and  $x_2$  is large then  $y$  is large. (80)

Assume that the membership functions for those rules are:

$$\mu_{SMALL}(x_1) = \mu_{SMALL}(x_2) = \begin{cases} 1 & \text{if } x < 0 \\ 1 - 0.1x & \text{if } 0 \leq x \leq 10 \\ 0 & \text{if } 10 < x \end{cases}$$

$$\mu_{LARGE}(x_1) = \mu_{LARGE}(x_2) = \begin{cases} 0 & \text{if } x < 0 \\ 0.1x & \text{if } 0 \leq x \leq 10, \\ 1 & \text{if } 10 < x \end{cases} \tag{81}$$

$$\mu_{SMALL}(y) = \begin{cases} 1 & \text{if } y < 0 \\ 1 - 0.05y & \text{if } 0 \leq y \leq 20 \\ 0 & \text{if } 20 < y \end{cases}$$

$$\mu_{MEDIUM}(y) = \begin{cases} 0 & \text{if } y < 0 \\ 0.1y & \text{if } 0 \leq y < 10 \\ 2 - 0.1y & \text{if } 10 \leq y < 20 \\ 0 & \text{if } 20 \leq y \end{cases}$$

$$\mu_{LARGE}(y) = \begin{cases} 0 & \text{if } y < 0 \\ 0.05y & \text{if } 0 \leq y \leq 20 \\ 1 & \text{if } 20 < y \end{cases} \tag{82}$$

The defuzzification values for (43) are:

Rule	$C_\mu$	$C_\nu$
R1	0	20
R2	10	20
R3	10	0
R4	20	0

Assume that the negative degree:

$$\gamma_{\check{A}}(x) = 1 - \mu_{\check{A}}(x), \quad \forall x \in X. \tag{83}$$

For a discrete case:  $x_1 = 3$  and  $x_2 = 7$ , we calculate the outputs of each rule according to the Sugeno model as follows.

Rule	$\mu$		$\nu$		max	FIS $C_\mu * \mu$	PIS $C_\nu * \nu$
	$\mu(x_1)$	$\nu(x_1)$	$\mu(x_2)$	$\nu(x_2)$			
R1	0.7	0.3	0.3	0.7	0.7	0.3*0	0.7*20
R2	0.7	0.7	0.7	0.3	0.3	0.7*10	0.3*20
R3	0.3	0.3	0.3	0.7	0.7	0.3*10	0.7*0
R4	0.3	0.7	0.3	0.7	0.3	0.3*20	0.7*0

Then, the outputs calculated by FIS and PIS are:

$$FIS = \frac{0.3 * 0 + 0.7 * 10 + 0.3 * 10 + 0.3 * 20}{0.3 + 0.7 + 0.3 + 0.3} = 10 \tag{84}$$

$$PIS = \frac{0.7 * 20 + 0.3 * 20 + 0.7 * 0 + 0.7 * 0}{0.7 + 0.3 + 0.7 + 0.7} = 8.33 \tag{85}$$

The accurate result of this case is 10, which is identical to that of FIS. This clearly shows that FIS is better than PIS in

this example. FIS performs better than PIS because it only uses the positive degree, which is well-oriented by the data. In real life situations where data may be variant and complex as shown in Example 1 and the weather nowcasting problem (the negative and neutral memberships appear therein), PIS is more effective than FIS because it can handle these membership degrees concurrently. In the next section, we will validate the performance of PIS through experiments.

### 4 Evaluation

#### 4.1 An illustration on the Lorenz system

Control is a practical area and there are various investigations on solving the control problem using fuzzy systems. In this section, PIS is applied to solve a classic control problem - the Lorenz system defined below.

$$\frac{dx}{dt} = \sigma(y - x), \tag{86}$$

$$\frac{dy}{dt} = x(\rho - z) - y, \tag{87}$$

$$\frac{dz}{dt} = xy - \beta z, \tag{88}$$

where  $\sigma, \rho, \beta$  are positive constants and often set as:  $\sigma = 10, \rho = 28, \beta = 8/3$  [1, 43]. It is noted that the system has an equilibrium point at the origin of coordinate. The component  $x$  with signal  $u$  is controlled. The change of  $x$  affects the whole system, which is proven to be asymptotically stable. Rewrite formula (86) as follow.

$$\frac{dx}{dt} = \sigma(y - x) + u. \tag{89}$$

In formula (89), there are only variables  $x$  and  $y$  so that to construct the rule set we do not need to create the membership graph for  $z$ . In this part, to study and design the system that is stable under the control of signal  $u$ , the Lyapunov theory [43] is used. In that way, the system will be stable in Lyapunov sense. The dynamical system in the interval  $[-40, 40]^3$  is examined as well. Each variable ( $x$  or  $y$ ) has 3 linguistic terms: N, Z and P.

**Table 1** The membership graph of linguistic term

Term	Parameters
N	((16, -40), (20, -40), (800, -40))
Z	((4, 0), (5, 0), (200, 0))
P	((16, 40), (20, 40), (800, 40))

**Table 2** Bad parameter functions for classic fuzzy controllers

Term	Parameters
N	(-16,40)
Z	(5,5)
P	(16,40)

Applying the algorithm in [43], the candidate function to design the controller is found.

$$V(x, y, z) = \frac{1}{2}(x^2 + y^2 + z^2). \tag{90}$$

From the candidate function, the complete set of rules is described as follows.

If  $x$  is P and  $y$  is P then  $u = -y(\rho + \sigma)$  (91)

If  $x$  is N and  $y$  is N then  $u = -y(\rho + \sigma)$  (92)

If  $x$  is P and  $y$  is N then  $u = -1$  (93)

If  $x$  is N and  $y$  is P then  $u = 1$  (94)

If  $x$  is P and  $y$  is Z then  $u = \sigma x + \frac{y^2 + \beta z^2}{x} - 10(\sigma + \rho)$  (95)

If  $x$  is N and  $y$  is Z then  $u = \sigma x + \frac{y^2 + \beta z^2}{x} + 10(\sigma + \rho)$  (96)

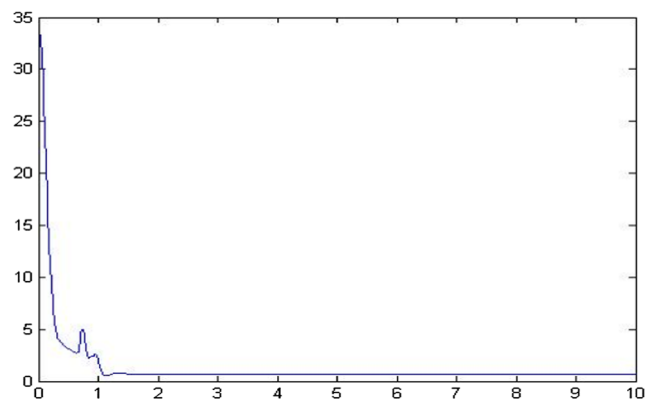
If  $x$  is Z and  $y$  is P then  $u = -y(\rho + \sigma)$  (97)

If  $x$  is Z and  $y$  is N then  $u = -y(\rho + \sigma)$  (98)

If  $x$  is Z and  $y$  is Z then  $u = -y(\rho + \sigma)$  (99)

where N, Z, P are the negative, zero and the positive degrees respectively. The Gaussian function is used to make the membership graph and it does not change the rules above.

This function is dependent on center  $c$  and width  $\sigma$  so let us denote  $(\sigma, c)$  instead of writing a full exponential form



**Fig. 4** Radius versus time of bad fuzzy controller

**Table 3** Bad membership graph for picture fuzzy controller

Term	Parameters
N	((16, -40), (20, -40), (800, -40))
Z	((4, 5), (5, 5), (200, 5))
P	((16, 40), (20, 40), (800, 40))

of the function.  $x$  and  $y$  have the same membership graph as in Table 1. In this table, the short denotations for the membership graph and Gaussian function are used. The problem needed to solve is determining the defuzzification values. This could be a problem to picture fuzzy controller because in PIS, initial parameters are adjusted by training but cannot be done as in picture fuzzy controller. In order to calculate the defuzzification values, the formulae (40–41) are used. Note that some functions are the same but they have different meanings; this is the reason we should use the formulae (40–41).

Recall that we do not have any idea about the parameters so that random values for centers and widths of membership graphs were chosen and expressed in Table 1. There is no exact way to know how the system runs but we can use the Runge - Kutta (RK) method especially RK-4 to approximate the states of Lorenz system. Denote

$$X = (x, y, z)^T. \tag{100}$$

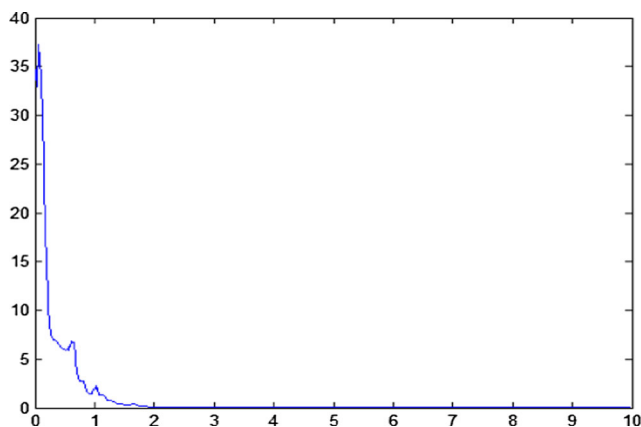
The system is rewritten as,

$$\frac{dX}{dt} = \dot{X} = f(X), \tag{101}$$

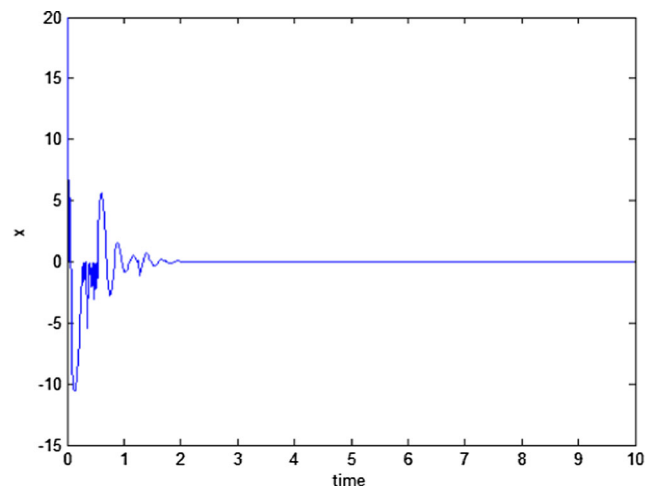
where  $\dot{X}$  is the partial derivative of width respect to time. We can estimate  $X(t)$  by

$$X_{n+1}(t) = X_n(t) + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4), \tag{102}$$

where  $X_n(t)$  is state of the system at  $n^{th}$  iteration.  $k_1, k_2, k_3, k_4$  are the functions of  $X_n(t)$  with



**Fig. 5** Radius versus time of bad picture fuzzy controller



**Fig. 6** Variable  $x$  versus time

different intervals. Specifically,  $k_1$  is the function on the slope at the beginning of interval.  $k_2$  is the function at the midpoint of interval ( $h/2$ ). Likewise,  $k_3$  is at the midpoint but starting from  $k_2$ . Lastly,  $k_4$  is the function at the end of interval.

$$k_1 = f(X_n(t)), \tag{103}$$

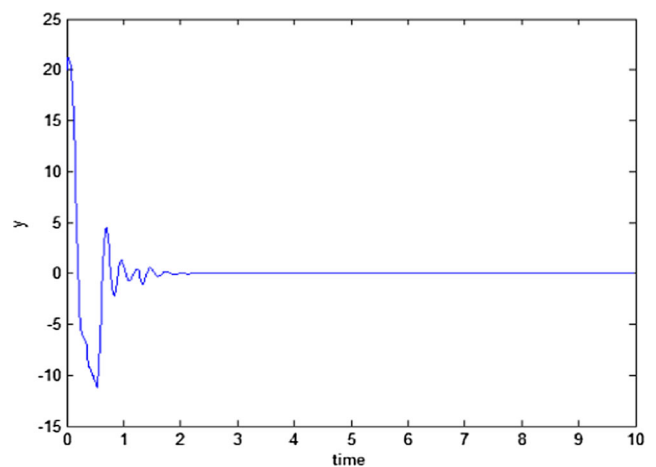
$$k_2 = f\left(X_n(t) + \frac{h}{2}k_1\right), \tag{104}$$

$$k_3 = f\left(X_n(t) + \frac{h}{2}k_2\right), \tag{105}$$

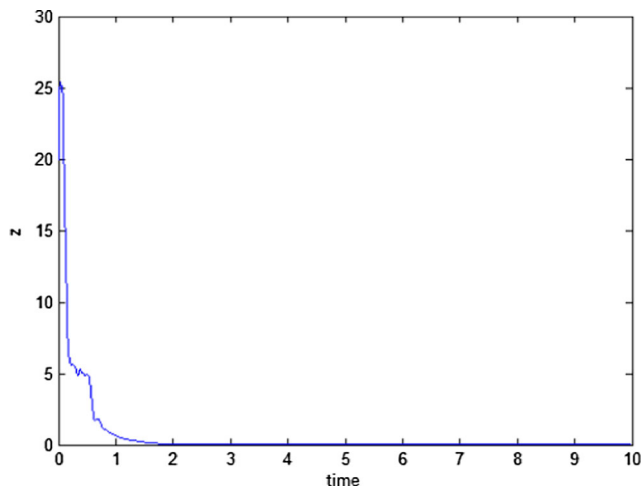
$$k_4 = f(X_n(t) + hk_3). \tag{106}$$

The step-size ( $h$ ) is a positive constant. Pick  $h = 0.01$  and examine the system in 10 seconds. The start points are  $x = 20, y = 20, z = 20$ .

Remind that design is very important and requires careful selection. For example, if the center for  $Z$  (zero) is not



**Fig. 7** Variable  $y$  versus time

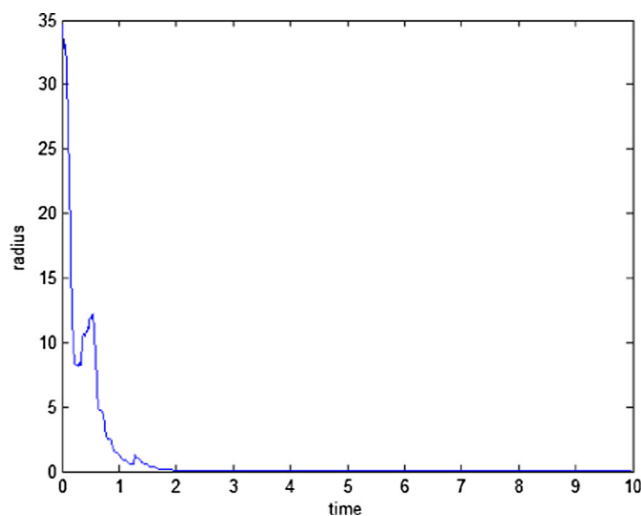


**Fig. 8** Variable z versus time

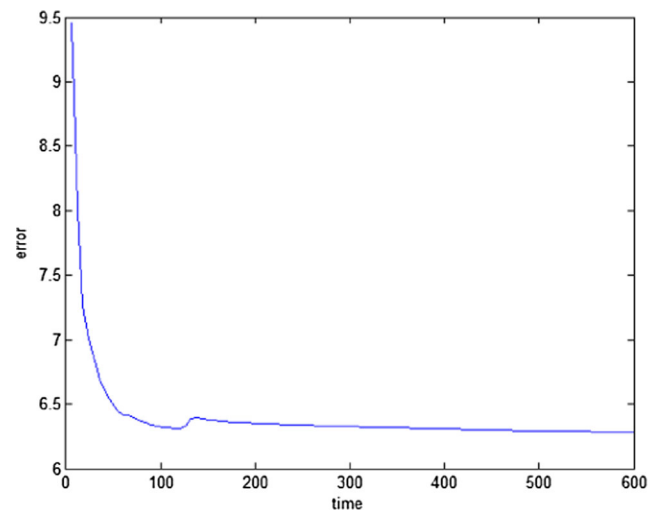
equal to zero, the controller may not be able to lead the system to the equilibrium point. Thus, the comparison of the effects of two controller systems especially on the bad selection should be investigated. The *first controller* uses fuzzy set with the parameters shown in Table 2. Note that each term has only one membership function associated with, and the same notations described in Table 1 are used. The bad design point is the center of the term being not zero.

As shown in Fig. 4, because the controller has the rule set designed by Lyapunov theory, the system converges to  $(x, y, z) = (0.0221, 0.6178, 0.0051)$  that is very near to the equilibrium point.

The *second controller* uses picture fuzzy set with the bad design shown in Table 3. The results are illustrated in Fig. 5. It is clear that the system finally converges to the origin even the bad parameters are selected. Since the neutral and negative degrees are utilized, they pull the convergence



**Fig. 9** Radius versus time

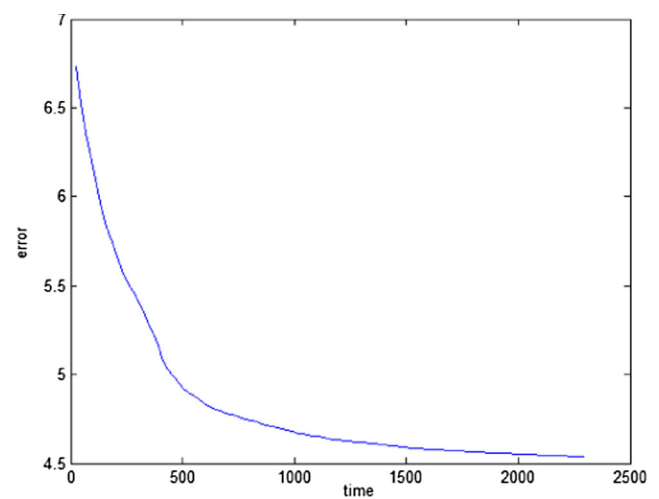


**Fig. 10** The first learning iteration of PIS on Housing dataset

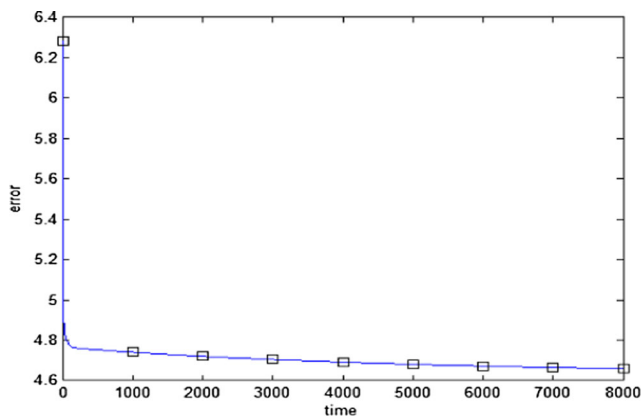
point back to the actual equilibrium point. This shows the advantages of using PFS to FIS.

We need to talk about the reason why the parameters in Table 1 are chosen. Intuitively, origin is the point at which the system is stable. Moreover, it is a crisp value to make the graph for Z (zero) thin. N and P do not show any difference in Lorenz system so that the symmetric across the zero is made. It is not necessary to make the graphs of x and y identical but in the example, we would like to simplify the computation. We now have the so-called “good design” parameters.

We have shown in this example the good parameters (in Table 1) and the bad ones (in Tables 2–3) to illustrate that using PIS would result in more accurate solutions than FIS. Starting from random initial parameters, by employing the



**Fig. 11** The first learning iteration of FIS on Housing dataset



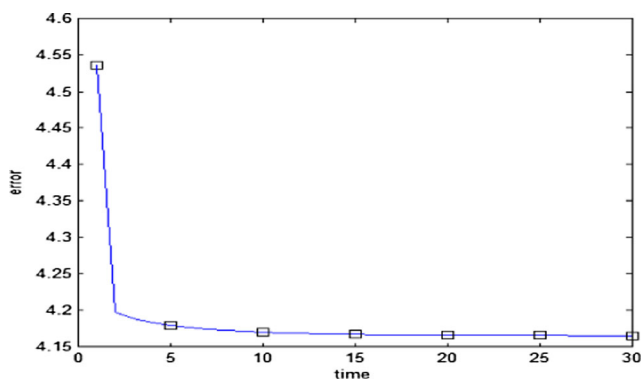
**Fig. 12** The second learning iteration of PIS on Housing dataset

membership graph with the learning strategy of PIS, good parameters can be achieved within small amount of time. The comparisons from Figs. 6, 7, 8 and 9 show that the convergence rate under control signal of PIS is fast with  $x$  coordinate being unstable in the few first seconds but quickly stable afterward (Fig. 6). Similar facts are found for  $y$  and  $z$  coordinates and time (Figs. 7–9). This depicts the advantage of PIS.

#### 4.2 The comparative results

In Section 2.1, we have discussed the Sugeno model, which is more general and has more advantages than Mamdani's [10, 11, 29]. Sugeno model is not only general but also very flexible with the performance of defuzzification functions. Each rule may have its own function increasing the power of the model. Therefore, this section compares PIS and FIS on the Sugeno model. The experiments are taken on the benchmark UCI Machine Learning Repository namely Housing, Iris, Glass and Breast Cancer [4]. The rule set consists of 140 rules. Constant function is not linear or higher-order. Additionally, the same function is shared for each rule to avoid over-fitting. The cross-validation method is Hold-out.

The results of the first learning iteration of PIS and FIS on Housing dataset are shown in Figs. 10 & 11, respectively.



**Fig. 13** The second learning iteration of FIS on Housing dataset

**Table 4** RMSE values of PIS to FIS in various datasets

Datasets	PIS	FIS
Housing	3.430	4.060
Iris	0.1779	0.1988
Glass	0.8594	0.8695
Breast Cancer	0.2670	0.2759

It is clear that PIS has the same power as FIS does. Note that both models are strongly affected by the initial values of parameters. If the initial values are slightly changed, the result of PIS is better than that of FIS. It is clear from the figures that the error of PIS at the first time is higher than that of FIS (resp. 9.4 vs. 6.7). However, PIS quickly reduces the error and reaches the stable state after 600 seconds (sec) while FIS needs more than 2000 sec even though it starts from better parameters. The experimental results clearly show that PIS performance is better than that of FIS.

The results of the second learning iteration of PIS and FIS are shown in Figs. 12 & 13, respectively. It is indicated that PIS performance is better than that of FIS. It takes PIS few rounds of training to get the error of 4.8, then 8000 rounds to reduce the error down to 4.66. The error difference is small but valuable because the smaller the error is, the harder the optimizing step runs. Back to FIS, it takes FIS 30 rounds to get to the optimal value and the model cannot be optimized more. In this learning step, computational time is not the problem because each row consumes a little time but preciseness matters. This gives us a hint of when to stop learning for a given error value.

By similar processes with those learning iterations, we gain the final error values of PIS and FIS on testing datasets as in Table 4.

The *findings* of experiments are as follows: PIS is better than FIS in term of RMSE. Table 4 shows the compared results of PIS and FIS. It is obvious that the values of PIS are smaller than those of FIS with the reduced percentages in comparison with FIS being 15.5 %, 10.5 %, 1.2 % and 3.2 %, respectively. This clearly shows the advantages of PIS over FIS.

## 5 Conclusions

In this paper, we proposed picture inference system (PIS) that integrates fuzzy inference system (FIS) with picture fuzzy sets. PIS was designed based on the membership graph and the general picture inference scheme. The pro-

posed system is adapted for all architectures such as the Mamdani fuzzy inference, the Sugeno fuzzy inference and the Tsukamoto fuzzy inference. Learning for PIS including training centers, widths, scales and defuzzification parameters is also discussed to build up a well-approximated model. The novel contribution of this paper – PIS model including the design and the learning method on picture fuzzy set was attributed to have better performance than those of FIS model. In the evaluation, the new system was validated on a classic example in control theory - the Lorenz system and on the benchmark UCI Machine Learning datasets. The findings from experimental results are as follows: i) PIS is capable to effectively perform a classic controller like the Lorenz system; ii) PIS is better than FIS in term of RMSE as shown in the experiments.

Further works of this research can be investigated in the following ways:

*Firstly*, other strategies of learning in PIS should be investigated. As being observed from the paper, PIS used the gradient method to train the parameters. This method has the advantage of fast processing but produces local optimal solutions. In order to achieve (global) better solutions, evolutionary algorithms such as genetic algorithm, particle swarm optimization, differential evolution and bees algorithm can be applied to improve the inference performance. For instance, the learning of parameters for the membership graph in phase 1 namely the centers, width and scales can be done through particle swarm optimization. Therein, a particle is encoded as the combination of three parameters. The swarm of particles is gradually optimized until stopping conditions hold. Analogously, the defuzzification parameters in phase 2 can be trained using harmony search. However, this must cooperate with the previous training from phase 1 in order to utilize the optimal results. This strategy initiates the idea of a hybrid training for PIS in further works.

*Secondly*, we should take into account the high order PIS with picture fuzzy rules. From step 3 in the description of PIS in Section 3.1, the fuzzy rule is represented and used in first order which is quite simple. In order to handle with complex control processes, high order fuzzy rules should be utilized. But more than that, as we are working on the picture fuzzy set, picture fuzzy rules should be designed and used instead of the traditional fuzzy rules. Picture fuzzy rules are characterized by three membership degrees namely the positive, neutral and negative. It is indeed necessary to take into account those degrees simultaneously in the computation. In the current PIS method, the positive, negative and neutral degrees are combined to adapt with a single output of traditional fuzzy inferences. This should be ameliorated if we would like to not only enhance the inference performance but also provide more information to the con-

trol process. Even though the empirical results of PIS are better than FIS, the combination of high order and picture fuzzy rules is a good choice to help us significantly improve the performance.

*Finally*, since real-life applications are often complicated; therefore how to apply and adapt the proposed PIS system to those is in high demand. In this paper, we just applied PIS to a popular control problem - the Lorenz system. When applying PIS to other ones, we should make trial-and-test of appropriate parameters as well as find appropriate functions for the membership graph. For instance, we can use Bell function for three lines in the membership graph instead of Gaussian. Similarly, the trapezoid and triangular functions can be used also. The mix between those functions is possible for certain kinds of applications. It is very useful if we can compare the performance of various membership graphs using different functions above.

**Acknowledgments** The authors wish to thank the Editor-in-chief and anonymous reviewers for their valuable comments and suggestions. We acknowledge the Center for High Performance Computing, VNU University of Science for executing the program on the IBM 1350 cluster server.

## References

1. Antosiewicz HA, Gautschi W (1962) Numerical methods in ordinary differential equations. In: Survey of numerical analysis, pp 314–346
2. Arabshahi P, Marks IIRJ, Caudell TP (1993) Adaptation of fuzzy inferencing: a survey. Proceedings of the IEEE/Nagoya University WWW on Learning and Adaptive Systems:1–9
3. Atanassov KT (1986) Intuitionistic fuzzy sets. Fuzzy Sets Syst 20(1):87–96
4. Bache K, Lichman M (2013) Housing data set. UCI machine learning repository. <https://archive.ics.uci.edu/ml/datasets/housing>. Accessed 01 June 2015
5. Cai Q, Hao Z, Yang X (2012) Gaussian kernel-based fuzzy inference systems for high dimensional regression. Neurocomputing 77(1):197–204
6. Castillo O, Martínez-Marroquín R, Melin P, Valdez F, Soria J (2012) Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. Inf Sci 192:19–38
7. Chaudhari S, Patil M (2014) Study and review of fuzzy inference systems for decision making and control. American International Journal of Research in Science, Technology, Engineering & Mathematics 5(1):88–92
8. Chen CP, Liu YJ, Wen GX (2014) Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems. IEEE Transactions on Cybernetics 44(5):583–593
9. Cuong BC (2014) Picture fuzzy sets. Journal of Computer Science and Cybernetics 30(4):409–420



10. Eyoh IJ, Umoh UA (2014) A comparative analysis of fuzzy inference engines in context of profitability control. *West African Journal of Industrial and Academic Research* 9(1):13–24
11. Fahmy R, Zaher H, Elfattah Kandil A (2015) A comparison between fuzzy inference systems for prediction (with application to prices of fund in Egypt). *Int J Comput Appl* 109(13):6–11
12. Gao Y, Liu YJ (2014) Adaptive fuzzy optimal control using direct heuristic dynamic programming for chaotic discrete-time system. *J Vib Control*:1077546314534286
13. Iman B, Reza M, Yashar M (2014) Diagnosis of diabetes using a combination of the Sugeno fuzzy inference systems and glow worms algorithms. *Advances in Environmental Biology* 8(12):269–272
14. Khan SA, Daachi B, Djouani K (2012) Application of fuzzy inference systems to detection of faults in wireless sensor networks. *Neurocomputing* 94:111–120
15. Khoobipour H, Khaleghi A (2012) A novel evolutionary-fuzzy system for function approximation and comparison the robust of used evolutionary algorithms. *International Journal Computer Technology & Applications* 3:1855–1869
16. Lai G, Zhang Y, Chen CP, Xie S, Liu YJ (2016) Fuzzy adaptive inverse compensation method to tracking control of uncertain nonlinear systems with generalized actuator dead zone. doi:[10.1109/TFUZZ.2016.2554152](https://doi.org/10.1109/TFUZZ.2016.2554152)
17. Liu YJ, Gao Y, Tong S, Li Y (2016) Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone. *IEEE Trans Fuzzy Syst* 24(1):16–28
18. Liu YJ, Tong S (2014) Adaptive fuzzy control for a class of nonlinear discrete-time systems with backlash. *IEEE Trans Fuzzy Syst* 22(5):1359–1365
19. Liu YJ, Tong S (2015) Adaptive fuzzy identification and control for a class of nonlinear pure-feedback MIMO systems with unknown dead zones. *IEEE Trans Fuzzy Syst* 23(5):1387–1398
20. Luo L, Wang YH, Sun YF (2014) Adaptive stable control for chaos systems by new fuzzy inference systems without any rule base. *Applied Mechanics and Materials* 644:367–372
21. Maldonado Y, Castillo O, Melin P (2013) Particle swarm optimization for average approximation of interval type-2 fuzzy inference systems design in FPGAs for real applications. In: *Recent advances on hybrid intelligent systems*. Springer, Berlin Heidelberg, pp 33–49
22. Melin P (2012) Image processing and pattern recognition with mamdani interval type-2 fuzzy inference systems. In: *Combining experimentation and theory*. Springer, Berlin Heidelberg, pp 179–190
23. Melin P (2012) Type-1 and type-2 fuzzy inference systems for images edge detection. In: *Modular neural networks and type-2 fuzzy systems for pattern recognition*. Springer, Berlin Heidelberg, pp 7–20
24. Mieszkowicz-Rolka A, Rolka L (2014) Flow graph approach for studying fuzzy inference systems. *Procedia Computer Science* 35:681–690
25. Nayak GK, Narayanan SJ, Paramasivam I (2013) Development and comparative analysis of fuzzy inference systems for predicting customer buying behavior. *International Journal of Engineering & Technology* 5(5):4093–2108
26. Öztaysi B, Behret H, Kabak Ö, Sari IU, Kahraman C (2013) Fuzzy inference systems for disaster response. In: *Decision aid models for disaster management and emergencies*. Atlantis Press, pp 75–94
27. Pancho DP, Alonso JM, Cordon O, Quirin A, Magdalena L (2013) FINGRAMS: Visual representations of fuzzy rule-based inference for expert analysis of comprehensibility. *IEEE Trans Fuzzy Syst* 21(6):1133–1149
28. Rong HJ, Huang GB, Liang YQ (2013) Fuzzy extreme learning machine for a class of fuzzy inference systems. *Int J Uncertainty Fuzziness Knowledge Based Syst* 21:51–61
29. Shleeg AA, Ellabib IM (2013) Comparison of mamdani and sugeno fuzzy inference systems for the breast cancer risk. *International Journal of Computer, Information Science and Engineering* 7:387–391
30. Siddique N, Adeli H (2013) Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing
31. Singh G, Kumai P, Goyal D (2014) A review: Fuzzy logic and its application. *International Journal of Engineering and Technical Research*:61–66
32. Son LH (2015) A novel kernel fuzzy clustering algorithm for geodemographic analysis. *Inf Sci* 317(C):202–223
33. Son LH (2015) DPFCM: A novel distributed picture fuzzy clustering method on picture fuzzy sets. *Expert Syst Appl* 42: 51–66
34. Son LH (2016) Dealing with the new user cold-start problem in recommender systems: a comparative review. *Inf Syst* 58:87–104
35. Son LH (2016) Generalized picture distance measure and applications to picture fuzzy clustering. *Appl Soft Comput* 46(C):284–295
36. Son LH, Thong NT (2015) Intuitionistic fuzzy recommender systems: an effective tool for medical diagnosis. *Knowl-Based Syst* 74:133–150
37. Son LH, Tuan TM (2016) A cooperative semi-supervised fuzzy clustering framework for dental X-ray image segmentation. *Expert Syst Appl* 46:380–393
38. Thong NT, Son LH (2015) HIFCF: an effective hybrid model between picture fuzzy clustering and intuitionistic fuzzy recommender systems for medical diagnosis. *Expert Syst Appl* 42(7):3682–3701
39. Thong PH, Son LH (2015) A new approach to multi-variable fuzzy forecasting using picture fuzzy clustering and picture fuzzy rule interpolation method. In: *Knowledge and systems engineering*, pp 679–690
40. Thong PH, Son LH (2016) A novel automatic picture fuzzy clustering method based on particle swarm optimization and picture composite cardinality. *Knowl-Based Syst* 109:48–60
41. Thong PH, Son LH (2016) Picture fuzzy clustering: a new computational intelligence method. *Soft Comput* 20(9):3549–3562
42. Thong PH, Son LH (2016) Some novel hybrid forecast methods based on picture fuzzy clustering for weather nowcasting from satellite image sequences. *Appl Intell*. doi:[10.1007/s10489-016-0811-1](https://doi.org/10.1007/s10489-016-0811-1)
43. Tomescu ML, Preitl S, Precup RE, Tar JK (2007) Stability analysis method for fuzzy control systems dedicated controlling nonlinear processes. *Acta Polytechnica Hungarica* 4(3):127–141
44. Tong S, Li Y, Li Y, Liu Y (2011) Observer-based adaptive fuzzy backstepping control for a class of stochastic nonlinear strict-feedback systems. *IEEE Trans Syst Man Cybern Part B Cybern* 41(6):1693–1704
45. Ton-That AH, Cao NT, Choi HI (2013) Fuzzy inference systems based on fuzzy associative memory with optimized membership functions. *International Journal of Engineering Associates* 2(3):25–31
46. Viet PV, Chau HTM, Son LH, Hai PV (2015) Some extensions of membership graphs for picture inference systems. In: *Seventh international conference on knowledge and systems engineering (KSE), 2015 IEEE*, pp 192–197
47. Zadeh LA (1965) Fuzzy sets. *Information and control* 8(3):338–353



**Le Hoang Son** obtained the PhD degree on Mathematics – Informatics at VNU University of Science, Vietnam National University (VNU). He has been working as a researcher and now Vice Director of the Center for High Performance Computing, VNU University of Science, Vietnam National University since 2007. His major field includes Soft Computing, Fuzzy Clustering, Recommender Systems, Geographic Information Sys-

tems (GIS) and Particle Swarm Optimization. He is a member of International Association of Computer Science and Information Technology (IACSIT), a member of Center for Applied Research in e-Health (eCARE), a member of Vietnam Society for Applications of Mathematics (Vietsam), Editorial Board of Neutrosophic Sets and Systems (NSS), Editorial Board of International Journal of Ambient Computing and Intelligence (IJACI, SCOPUS) and associate editor of the International Journal of Engineering and Technology (IJET). Dr. Son served as a reviewer for various international journals and conferences such as PACIS 2010, ICMET 2011, ICCTD 2011, KSE 2013, BAFI 2014, NICS 2014 & 2015, ACIIDS 2015 & 2016, ICNSC15, GIS-2015, FAIR 2015, International Journal of Computer and Electrical Engineering, Imaging Science Journal, International Journal of Intelligent Systems Technologies and Applications, IEEE Transactions on Fuzzy Systems, Expert Systems with Applications, International Journal of Electrical Power and Energy Systems, Neural Computing and Applications, International Journal of Fuzzy System Applications, Intelligent Data Analysis, Computer Methods and Programs in Biomedicine, World Journal of Modeling and Simulation, Knowledge-Based Systems, Engineering Applications of Artificial Intelligence. He gave a number of invited talks at many conferences such as 2015 National Fundamental and Applied IT Research (FAIR 15'), 2015 National conference of Vietnam Society for Applications of Mathematics (VietSam15'), 2015 Conference on Developing Applications in Virtual Reality, GIS and Mobile technologies, and International Conference on Mathematical Education Vietnam 2015 (ICME Vietnam 2015), 2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS 16'), and 2016 HUST Conference on Applied Mathematics and Informatics (SAMI 16'). Dr. Son has got 67 publications in prestigious journals and conferences including 30 SCI / SCIE papers and undertaken more than 20 major joint international and national research projects. He has published 2 books on mobile and GIS applications. So far, he has awarded "2014 VNU Research Award for Young Scientists", "2015 VNU Annual Research Award" and "2015 Vietnamese Mathematical Award".

**Pham Van Viet** was born in Vietnam, in 1993. He holds the Bachelor of Engineering from Hanoi University of Science and Technology, Vietnam in 2016. His interests span the areas of Fuzzy sets and systems, and Data Mining. Currently, he is an IT engineer in Japan.



**Pham Van Hai** obtained a Masters Degree (M.Sc.) in computer science at Asian Institute of Technology. Following the Masters Degree he has been awarded Doctor of Engineering degree (Ph.D.) in computer intelligence and information science by Ritsumeikan University, Japan, visiting researcher in Computational Methods and Machine learning at Lamar University (USA). He has currently jointed research projects at Soft Intelligence Lab, Rit-

sumeikan University and worked at lecturer- assistant professor at School of Information Technology and Communication, Hanoi University of Science and Technology. He has been conducted as Master Supervisor and Phd advisor. He has published 70 international/national journal and conference publications and joined with international projects, the editorial board on a number of international journals. He is interested in how intelligent techniques and approaches can enable novel ways of interacting with intelligent computation, big data and in the new challenges that human awareness in dynamic system, intelligent hybrid systems, intelligent decision support systems, data mining, complex systems and context-aware system, Artificial Intelligence, Knowledge-based Systems for the domain of health care, telemedicine, automatic intelligent system dialogs between doctor and patient.