INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 6.078
(Volume 6, Issue 2)
Available online at: www.ijariit.com

Analytical overview of defuzzification methods

K. S. Gilda
kalpana.manudhane@gmail.com
College of Engineering and Technology, Akola, Maharashtra

Dr. S. L. Satarkar
shri1satarkar@rediffmail.com
College of Engineering and Technology, Akola, Maharashtra

ABSTRACT

Fuzzy systems are of basically two types- fuzzy controllers and fuzzy reasoning systems. Defuzzification is a last step of fuzzy system, which converts fuzzy output set to a crisp value. This paper discusses essential set of parameters that a defuzzification method should exhibits. Classification of different defuzzification methods is presented and each method is evaluated briefly based on the parameters discussed. The analysis of these methods emphasizes need of a defuzzification method suitable for fuzzy reasoning systems as well as fuzzy controllers.

Keywords— *Defuzzification, Parameters, Fuzzy, Membership function*

1. INTRODUCTION

The concept of fuzzy sets was introduced by [1]. Fuzzy sets are characterized by membership functions. Fuzzy logic enables us to do approximate reasoning rather than being exact. Fuzzy logic is widely used in many different fields such as control systems, robotics, medical science and expert systems. Most of these applications can be regarded as system with numerical input and numerical output; but internally these systems work with fuzzy values [2]. Every fuzzy system consists of following stages (This is explained in [3], [4]):

1.1 Fuzzification of the input variables

The first step is to take the inputs (crisp) and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions (*fuzzification*). After the inputs are fuzzified, role of fuzzy rules comes in picture. Rules are in the form *antecedents imply to the consequent*. Fuzzy rules form the triggers of fuzzy engine. For example:

If BP is *LOW operator* SUGAR is *HIGH* then HEALTH-RISK is *CRITICAL*.

Where, BP (with set of fuzzy terms- *LOW*, *NORMAL* & *HIGH*) and SUGAR (with set of fuzzy terms- *LOW*, *NORMAL* & *HIGH*) are input parameters (or fuzzy variable), HEALTH-RISK is output parameter with set of fuzzy terms- *CRITICAL*, *NORMAL*. Fuzzy *operator* joining two variables may be and / or. The input to the fuzzy operator is membership values from fuzzified input variables.

1.2 Implication

Before applying the implication method, you must determine the rule weight. Every rule has a *weight* (a number from 0 through 1). Generally, this weight is 1 and thus has no effect. However, you can decrease the effect of one rule relative to the others by changing its weight value. After proper weighs has been assigned to each rule, the implication method is implemented. The input for the implication process is a single number given by the antecedents, and the output is a reshaped fuzzy set. Implication is implemented for each rule.

1.3 Aggregation

Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set.

1.4 Defuzzification

The last is defuzzification in which aggregated fuzzy output set is converted into a single crisp value. This step has more significance in case of fuzzy controllers.

The paper mainly focuses on defuzzification method. Section 2 elaborates basic notations and definitions. Section 3 describes various parameters that a defuzzification method should satisfy. Section 4 introduces different categorized defuzzification methods. Section 5 presents discussion and conclusion.

2. PRELIMINARY NOTATIONS AND DEFINITIONS

Notations are being used as described in [5] , [2]. Fuzzy sets are characterized by membership function μ. A membership function maps each element x of the universe of discourse X to a degree of membership which is a number between 0 (indicating element not included) and 1(indicating fully included).

$$\mu: X \rightarrow [0,1]$$

Here we consider X bounded by lower bound x_{inf} and upper bound x_{sup}, belongs to the set of real numbers R, so that:

$$X=[\, x_{inf} , x_{sup}]\subset R$$

The set of all elements that have nonzero degree of membership is called the support of a membership function.

$$\text{supp}(\mu(x)) = \{x \mid x \in X \text{ and } \mu(x) > 0\}$$

The set of elements whose are fully included (degree of membership is 1) is called kernel of a membership function.

$$\text{ker}(\mu(x)) = \{x \mid x \in X \text{ and } \mu(x) = 1\}$$

The set of elements having largest degree of membership is called core of a membership function.

$$\text{core}(\mu(x)) = \{x \mid x \in X \text{ and } \max(\mu(x))\}$$

Defuzzification is expressed by a defuzzification operator D or f^{-1}. This operator maps membership function (representing fuzzy set) to crisp number (i.e. to element of the universe of discourse).

$$f^{-1}: \mu(x) \rightarrow X$$

3. ESSENTIAL PARAMETERS FOR DEFUZZIFICATION

To evaluate defuzzification methods we need set of parameters. Runkler and Glesner [2] proposed a set of constraints by grouping different parameters. These constraints were also elaborated in [5], [6], [7]. The set of constraints include basic constraints, graphically motivated constraints, constraints motivated by fuzzy operations and application specific constraints. Let's discuss in detail.

3.1 Basic constraints

Some fundamental mathematical concepts are collected in this group. It includes zero element, one element and monotony.

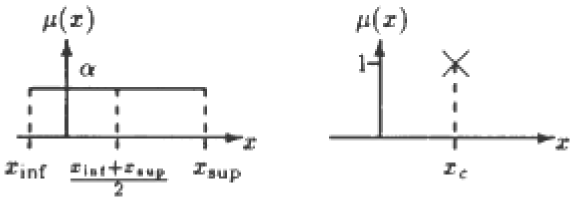

Fig. 1: (a) Zero element (b) One element

(a) Zero element: The membership function, which assigns a constant degree of membership, to all elements of X are defuzzified to the mean of the universe (see Fig. 1(a)).

(b) One element: A membership function with only one nonzero value. The preference of one and only element enforces that the one element is defuzzified to exactly this element (see Fig. 1(b)).

(c) Monotony: Monotony is a property defined by the following four sub properties (see Fig. 2 a–d):

- Removing a part of the membership function right from the defuzzified value does not move the defuzzified value to the right.
- Adding something to the membership function right from the defuzzified value does not move the defuzzified value to the left.
- Removing a part of the membership function left from the defuzzified value does not move the defuzzified value to the left.
- Adding something to the membership function left from the defuzzified value does not move the defuzzified value to the right.

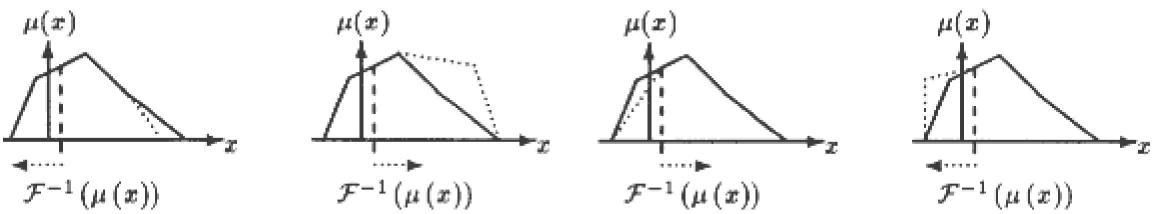

Fig. 2: Monotony (a) Remove right (b) Add right (c) Remove left (d) Add left

It means that defuzzication value can only move to that side where the elements are getting better". This property is also mentioned in [2].

(d) Continuity: The criterion of continuity is motivated by the fact that a discontinuous defuzzification seems unnatural. A small variation in any of the degrees of membership should not result in a big change in the defuzzification value.

3.2 Graphically motivated constraints

These are extracted from considerations about the graphical representation of membership functions. It is natural to demand for defuzzification constraints relating to the graphic representation of membership functions, i.e. dealing with symmetry, translation and scaling.

(a) Symmetry: Symmetry means that the relative position of the defuzzified value does not vary if the orientation of the support interval changes as shown in Fig. 3.

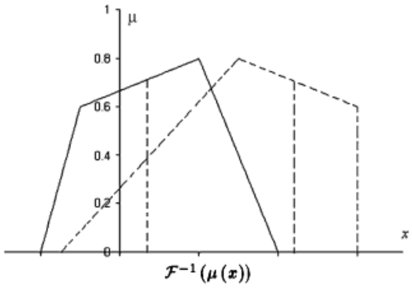

Fig. 3: Symmetry

(b) x-Translation: The relative position of the defuzzified value has to remain constant when the membership function is moved to the left or to the right (see Fig. 4(a)).

(c) x-Scaling: When scaling the support interval by a constant factor, again the relative defuzzification value position must be maintained (see Fig. 4(b)).

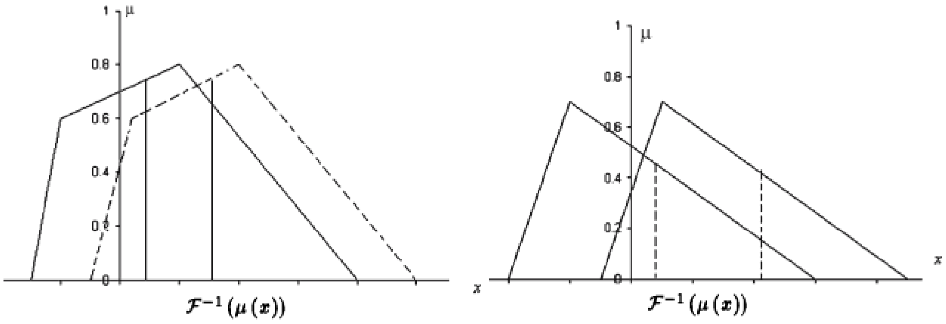

Fig. 4: (a) x-translation (b) x-scaling

(d) **μ-Translation (offset):** When adding a constant offset to the membership function i.e. increasing the degree of membership of every element of the universe, the defuzzified value remains constant (see Fig. 5(a))

(e) **μ-Scaling:** Multiplying all membership values by a constant value does not change the shape of the membership function, but only the range of the membership value interval. This scaling does not change the defuzzified value (see Fig. 5(b)).

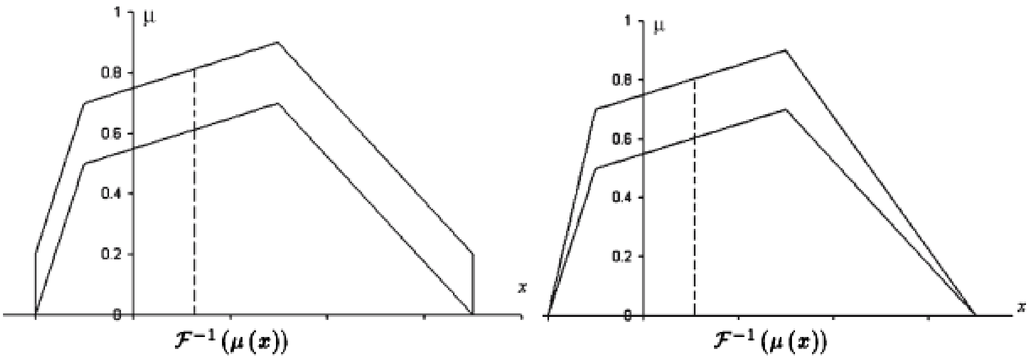

Fig. 5: (a) μ -translation (b) μ-scaling

(f) **Core selection:** An element with a higher degree of membership is considered as having "more of the property that is related to the fuzzy set". It seems reasonable to let the defuzzification operator make a selection amongst the elements with the hightest degree of membership.

3.3 Constraints motivated by fuzzy operations

This group gives a connection between the definitions of fuzzy operations and the definition of defuzzification operators. Fuzzy systems perform unary and binary operations. The unary operations are the hedges. The set of binary operations can be split into *t*-norms and *t*-conorms. Today in most applications the minimum operator or the algebraic product serves as *t*-norm while the maximum operator or the bounded sum is used as *t*-conorm.

(a) **t-Norm:** Some considerations about the properties of t-norms make it reasonable to require that the defuzzified result of a t-norm operation should be a member of the interval constrained by the defuzzified values of the two t-norm arguments.

(b) **t-Conorm:** t-norms and t-conorms are dual operations, so it is the same property for t-conorms also. Fig. 6 shows t-norm and t-conorm.

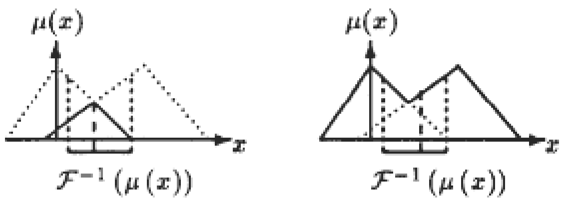

Fig. 6: (a) t-norm (b) t-conorm

(c) **Hedges:** Linguistic modifiers have two main behaviors with regard to their effect on the qualifications they modulate. They have either a behavior of reinforcement (such as very, strong) or behavior of weakening (such as “more or less”, “relatively”).

- **Restrictive modifiers/concentrators:** The reinforcing modifiers provide a characterization, which is stronger than the original one. The modifier “very” associated with the transformation tm $(\mu) = \mu^2$ was proposed by Zadeh. Other examples of these quantifiers include “extremely”, “positively”. As concentration increases (i.e. membership function get squared), defuzzification value also increases.

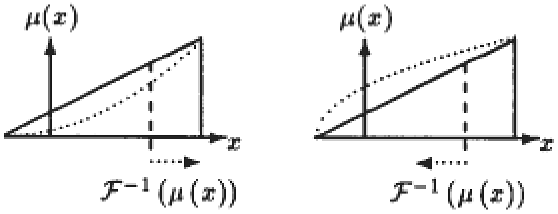

Fig. 7: (a) Concentration (b) Dilation

- **Expansive modifiers/dilators:** The weakening modifiers provide a new characterization, which is less strong than the original one. The modifier “more or less”, associated with the transformation tm$(\mu) = \mu^{1/2}$ was introduced by Zadeh. Other examples of these quantifiers include “somewhat”, “negatively”. As dilation increases (i.e. taking square-root of membership function), defuzzification value decreases. Concentration and dilation are shown in fig. 7.

3.4 Application specific constraints

In different applications specific defuzzification properties are required. Examples of two such constraints Computational efficiency and Transparency [5] are given below.

(a) **Computational efficiency:** In some applications, the computational efficiency of the defuzzification operator is extremely important. Especially in fuzzy controllers the choice of the defuzzification operator is influenced by the number of operations needed to calculate the defuzzification value.

(b) **Transparency for system design:** A transparent, easy to under-stand defuzzification operator is likely to be preferred over a complicated mathematical formula with no clear foundations.

4. DEFUZZIFICATION METHODS

Based on the essential parameters for defuzzification explained in previous section, we can now discuss and evaluate existing defuzzication methods. Here, we assume that rules have been applied to fuzzy inputs; implication and aggregation methods have been applied. So, input to defuzzification methods is aggregated fuzzy set. Defuzzification methods are broadly classified as maxima methods, Distribution methods, Area methods, parameter-based (extended method), specific methods and miscellaneous methods. This section gives brief introduction of methods in all categories.

4.1 Maxima methods

The maxima methods have the common property that they select an element from the core (i.e. elements with maximum membership value) as defuzzification value. By definition, these methods fulfill the core selection parameter. These methods satisfy parameters such as monotony and x- translation. Also, these methods are efficient. These methods are explained in [5].

4.1.1 Random Choice of Maxima (RCOM): In this basic general method all elements of the core of μ(x) are considered as being equal candidates for the defuzzification value. Under the assumption that the core only contains a finite number of elements, the defuzzification value can be calculated as the outcome of a random experiment with probabilities:

$$Prob\big(D\big(\mu(x)\big) = x_0\big) = \begin{cases} \dfrac{1}{\left|core\big(\mu(x)\big)\right|} & if\ x_0 \in core\big(\mu(x)\big) \\ 0 & otherwise \end{cases}$$

Because this method is nondeterministic, the only criterion it fulfills is the core selection criterion. This method can be used for arbitrary universes.

4.1.2 First of Maxima (FOM): This method is only defined on universes with an ordinal scale. This method selects smallest element from the core as defuzzification value.

$$\text{FOM}(\mu(x)) = \min \text{core}(\mu(x))$$

4.1.3 Last of Maxima (LOM): This method is only defined on universes with an ordinal scale. This method selects largest element from the core as defuzzification value.

$$\text{FOM}(\mu(x)) = \max \text{core}(\mu(x))$$

4.1.4 Middle of Maxima (MOM): Like the FOM and LOM methods, this defuzzification method requires an ordinal scale. If the core contains an odd number of elements, then the middle element of the core is selected. Otherwise, the defuzzification value depends on the implementation as follows.

For a crisp set X and $x_0 \in X$ let

$$X_{<x0} = \{\ x|x \in X \text{ and } x<x_0\} \text{ and } X_{>x0} = \{\ x|x \in X \text{ and } x>x_0\}$$

if | core(μ(x)| is odd then

$|\,\text{core}(\mu(x)_{<\text{MOM}((\mu(x))}\,| = |\,\text{core}(\mu(x)_{>\text{MOM}((\mu(x))}\,|$

else

$|\,\text{core}(\mu(x)_{<\text{MOM}((\mu(x))}\,| = |\,\text{core}(\mu(x)_{>\text{MOM}((\mu(x))}\,| \pm 1$

depending on choice of implementation.

The MOM defuzzification method is deterministic. There are just two possible definitions depending on the treatment of a core with an odd number of elements. Or we can say that there are two different MOM methods: $MOM_<$ and $MOM_>$.

4.2 Distribution methods

These methods first convert the membership function into probability distribution and then compute the expected value. These methods have a property of continuity which is essential for the fuzzy controllers.

4.2.1 Center of Gravity (COG): It is a basic general defuzzification method that computes the center of gravity of the area under membership function. This method is very popular and mostly used in fuzzy controllers.

$$\mathrm{COG}(\mu(x)) = \sum_{x_{min}}^{x_{max}} x \cdot \mu(x) \Big/ \sum_{x_{min}}^{x_{max}} \mu(x)$$

COG does not fulfill the constraint of core selection and triangular conorm. COG satisfies constraint of monotony, x-translation, x-scaling and continuity. The COG method has proven to work well, with efficient and accurate results. Its main drawback is the excessive time to calculate the center of gravity.

4.2.2 Mean of Maxima (MeOM): This method is derived from COG. It calculates the mean of all elements of core of fuzzy set.

$$MeOM\big(\mu(x)\big) = \frac{\sum_{x \in \mathrm{core}(\mu(x))} x}{\left|\mathrm{core}\big(\mu(x)\big)\right|}$$

MeOM does not satisfy the core selection constraint as it does not guarantee the selection of an element from the core for nonconvex fuzzy sets. For convex fuzzy sets, it satisfies the core selection constraint. It also satisfies the constraint of monotony, x-translation and x-scaling.

4.3 Specific methods

Specific methods combine the aggregation and defuzzification steps into one computational process [5]. The output of the system has a number of fuzzy output sets A_i. The fuzzy inference engine computes for each of these output sets a degree of applicability $\propto_i$. Instead of aggregating all output fuzzy sets, we associate with every fuzzy output set A_i a characteristic numerical value a_i. We calculate the defuzzification value immediately using a_i and $\propto_i$. This procedure eliminates the aggregation step, resulting in a gain in performance. In general, the values a_i depend on the shape of the fuzzy output set A_i and on the degree $\propto_i$.

4.3.1 Fuzzy Mean (FM): It computes weighted sum like COG. The difference is that FM uses pre-calculated numerical value a_i for each fuzzy set.

$$FM\big(\mu(x)\big) = \sum_{i=1}^{N_A} \propto_i a_i \Big/ \sum_{i=1}^{N_A} \propto_i$$

where, N_A represents the number of fuzzy output sets, $\propto_i$ represents the degree of each output set and a_i represents the numerical value of output set i. The FM method has proven to be efficient and widely used in fuzzy controllers.

4.3.2 Quality Method (QM): QM is a specific defuzzification method with the following formula-

$$QM\big(\mu(x)\big) = \sum_{i=1}^{N_A} (\propto_i / d_i) a_i \Big/ \sum_{i=1}^{N_A} \propto_i / d_i$$

Where, d_i equals the width of the support of A_i. This means that the method contains a vertical ($\propto_i$) as well as a horizontal component (d_i). The aim of the QM method is to increase the importance of the more crisp output sets (i.e., the sets with a more narrow support). The weight of a particular term is big if the support is small and the degree $\propto_i$ big.

4.3.3 Plateau Average (PA): PA is a specific method. After applying degree of applicability $\propto_i$, output fuzzy sets are assumed to be in the form of plateau (trapezoidal shape). This method averages midpoint of the plateau of each output fuzzy set. Here, characteristic numerical value a_i is the midpoint of plateau.

$$PA\big(\mu(x)\big) = \sum_{i=1}^{N_A} m_i \Big/ N_A$$

Where, m_i represents the midpoint of each plateau (output fuzzy set). The PA method is conceptually simple, efficient and accurate. This method is explained in [8].

4.4 Parameter-based extended methods

The extended defuzzification methods can be characterized by the existence of one or more parameters that have to be chosen by the designer to fully determine the defuzzification process. Basic general defuzzification methods are extended with one or more parameters. Extended methods can be modified for speed into particular methods by selecting specific values of parameters.

4.4.1 Basic Defuzzification Distributions (BADD): This method is proposed in [9]. It is a distribution method which converts a fuzzy set into a probability distribution. BADD is the extended version of COG method.

$$\mathrm{BADD}(\mu(x), \gamma) = \sum_{x_{min}}^{x_{max}} x \cdot \mu(x)^{\gamma} \Big/ \sum_{x_{min}}^{x_{max}} \mu(x)^{\gamma} \quad where\ \gamma \epsilon [0, +\infty[$$

A parameter γ indicates the confidence in the system. Higher the value of γ, more the confidence. The parameter γ can be adjusted so that the method becomes specific one:

$$\text{BADD}(\mu(x), 0)=\text{MeOS}$$

where MeOS = mean of support.

$$\text{BADD}(\mu(x), 1)=\text{COG}$$

$$\lim_{\gamma\to\infty} \text{BADD}(\mu(x), \gamma) = \text{MeOM}$$

4.4.2 Weighted Fuzzy Mean (WFM): WFM is an extended method as well as a specific method. This is the parameterized version of FM method.

$$FM(\mu(x)) = \sum_{i=1}^{N_A} w_i \propto_i a_i \Big/ \sum_{i=1}^{N_A} w_i \propto_i$$

Where w_i is the weight associated with output set i. The important characteristic of this method is the possibility to introduce a degree of importance for every output set. If the weights are chosen by

$$w_i = area(O_{A_i})$$

where, O_{A_i} denotes ordinate set of A_i. In this case, WFM behaves like COG method than FM. Only the drawback is that overlapping area between two adjacent output sets is counted more than once.

4.5 Area methods

This group of defuzzification methods uses the area under the membership function to determine the defuzzification value. Methods in this group are applicable primarily for fuzzy controllers.

4.5.1 Center of Area/Bisector Method (COA): This method calculates the defuzzification value as position under curve where the areas on both sides are almost equal. The COA minimizes the expression:

$$\left| \sum_{x=\inf(X)}^{x^*} \mu(x) - \sum_{x^*}^{\sup(X)} \mu(x) \right|$$

where, x^* is a defuzzification value. The COA method does not satisfy core selection constraint and t-conorm. COA satisfies constraints of monotony, x-translation, x-scaling and continuity.

4.5.2 Center of Sum (COS): This area method is also a specific method which works on every output fuzzy set instead of aggregated output set. Defuzzification value x^* is defined as:

$$x^* = \frac{\sum_{i=1}^{k} A_i \overline{x_i}}{\sum_{i=1}^{k} A_i}$$

Where, A_i represents area of fuzzy output set i, $\overline{x_i}$ represents center of area A_i and k is the total number of rules fired. Drawback of this method is that overlapping area between adjacent fuzzy output set is counted twice.

5. DISCUSSION AND CONCLUSION

We have discussed major categories of defuzzification methods such as maxima methods (which selects element from core), distribution methods (which convert membership function into probability distribution and then compute expected value), specific methods (work on individual fuzzy output set), parameter based extended methods (which include one or more parameter to fully determine defuzzification process) and area methods (which work on area under membership function).

Maxima methods satisfy the constraint of core selection. Distribution methods and area methods provide continuity. Specific methods are more efficient as they work on every output fuzzy set rather than aggregated output set. Extended methods can be tuned to specific methods by selecting appropriate values of parameters.

Now, let's focus on another aspect. Fuzzy systems can be broadly classified into two classes- fuzzy reasoning (or knowledge) systems and fuzzy control systems. Goal of fuzzy knowledge systems is to provide qualitative reasoning system for a specific domain. In cases, where we get single output set and linguistic results convey sufficient information, there is even no need of defuzzification. But, in other when we get multiple fuzzy output sets, defuzzification step is needed to obtain single crisp value. On the other hand, accuracy and precision of defuzzification value is very crucial for fuzzy controllers.

Each fuzzy system has specific requirement for defuzzification method. As maxima methods select element from core, they are good for fuzzy reasoning system. Whereas, distribution methods (COG and others) and area methods possess property of continuity, which is essential requirement for fuzzy controllers; these methods are more suitable for fuzzy controllers. A defuzzification method is needed which is suitable for both fuzzy reasoning system and fuzzy controllers. A bridge is needed to overcome the gap between maxima methods and distribution methods. In other words, maxima method is needed which possess continuity so that it will be more efficient and suitable for fuzzy controllers.

6. REFERENCES

[1] L. A. Zadeh, "Fuzzy sets," Information and control, vol. 8, p. 338–353, 1965.
[2] T. A. Runkler and M. Glesner, "A set of axioms for defuzzification strategies towards a theory of rational defuzzification operators," in [Proceedings 1993] Second IEEE International Conference on Fuzzy Systems, 1993.
[3] K. K. a. S. B. N. Elaine Rich, Artificial Intelligence, 3, Ed., McGraw Hill Education, 2009.

[4] "https://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html," [Online].
[5] W. Van Leekwijck and E. E. Kerre, "Defuzzification: criteria and classification," Fuzzy sets and systems, vol. 108, p. 159–178, 1999.
[6] A. Chandramohan, M. V. C. Rao and M. S. Arumugam, "Two new and useful defuzzification methods based on root mean square value," Soft Computing, vol. 10, p. 1047–1059, 2006.
[7] A. Amini and N. Nikraz, "Proposing two defuzzification methods based on output fuzzy set weights," International Journal of Intelligent Systems and Applications, vol. 8, p. 1–12, 2016.
[8] S. S. Lancaster and M. J. Wierman, "Empirical study of defuzzification," in 22nd International Conference of the North American Fuzzy Information Processing Society, NAFIPS 2003, 2003.
[9] D. P. Filev and R. R. Yager, "A generalized defuzzification method via BAD distributions," International Journal of Intelligent Systems, vol. 6, p. 687–697, 1991.
[10] A. Vahidi, "A New Defuzzification Method for Solving Fuzzy Mathematical Programming Problems," Hacettepe Journal of Mathematics and Statistics, vol. 48, p. 845–858, 2019.
[11] Y. Çelikbilek, "The importance of the place of defuzzification step in fuzzy systems," Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi, vol. 23, p. 139–148, 2019.

BIOGRAPHY

Kalpana S. Gilda
Assistant Professor
College of Engineering and Technology, Akola, Maharashtra

Dr. Shrikant L. Satarkar
Head and Associate Professor
College of Engineering and Technology, Akola, Maharashtra