Pliant control system: implementation

József Dombi and Tamás Szépe
Department of Computer Algorithms and Artificial Intelligence
University of Szeged, Hungary
dombi@inf.u-szeged.hu, szepet@inf.u-szeged.hu

***Abstract*—The plaint system is determinated upto a generator function. Choosing this function, we get the conjunctive, disjunctive (t-norm and t-conorm), aggregation (uni-norm) operators and all type of unary operators, i.e. negation, hedges, sharpness operators, even the membership (distending) function. Special case of the pliant systems is the Dombi operator. Pliant control system (PCS) is derived from the fuzzy control system by replacing their elements with pliant components. In this article we show that PCS much more efficient, compare to fuzzy control system. The number of the rules drastically decrease while the speed increases.**

***Index Terms*—pliant control system, defuzzification, fuzzy arithmetics**

I. Introduction

As most modelling and control applications require crisp outputs, when applying fuzzy inference systems, the fuzzy system output $A(y)$ usually has to be converted into a crisp output y^*. This operation is called defuzzification. The most popular defuzzification methods are the center of gravity (COG) and the mean of maxima (MOM) methods. More general frameworks have been proposed, in which the COG and MOM defuzzification methods have their place, such as the parametric BADD (basic defuzzification distribution) and SLIDE (semi-linear defuzzification) methods of Yager and Filev [8], [26]. They are essentially based on the transformation of a possibility distribution into a probability distribution based on Klir's principle of uncertainty invariance. The main emphasis is on the learning of the parameters involved, which is treated as an optimization problem [11], [22], [25]. This issue falls outside the scope of the present paper. Note that in the literature the terms for describing the different defuzzification methods vary from source to source. The terms center of gravity defuzzification, center of area defuzzification and center of sum defuzzification, for instance, refer to different methods in some sources and are used as synonyms in other sources. Therefore, one should pay attention to the formal definitions of the defuzzification methods rather than to their names. In [14], [23] comprehensive overviews are given on defuzzification methods. In this paper the same terminology is used as in [14].

In the article [2] the authors show two computational methods, the slope-based method and the modified transformation function method, are introduced for the center of gravity defuzzification method for trapezoidal membership functions forming a fuzzy partition.

This article deals with the computational aspects of the pliant defuzzification method. When applying the pliant defuzzification method, the crisp output y^* of the system will change continuously when the input values change continuously, a desirable property in modelling and control applications. However, the pliant defuzzification method has a high computational burden [4], [21], which is a considerable disadvantage in control and model identification, and in tuning applications. This high cost is often circumvented by introducing new defuzzification methods that intend to approximate the center of gravity [21], [24].

Fuzzy arithmetic based α-cuts, where the result of the α-cuts represent an interval. The arithmetic can be understand as an interval arithmetic of the α-cuts. Instead of dealing with intervals we are dealing with left and right hand sided soft inequalities which define the interval. We offer a new calculation procedure of arithmetics, when these soft inequalities meet certain properties (i.e. strict monotonously increasing function represent the inequality). We show that the result of linear combinations of linear is also linear and the linear combination of sigmoid is also sigmoid function (i.e. they are closed under linear combination). We give the result of other operation, too. The soft inequalities define an interval by using proper conjunctive and disjunctive operator. We give such operations, too.

The idea the fuzzy quantities could be arithmetically combined according to the laws of fuzzy set theory is due to Zadeh [28]. Soon after, several researchers worked independently along these lines, such as Jain [10], Mizumoto and Tanaka [17], [18], Nahmias [19], Nguyen [20], Dubois and Prade [5]. It was only further on recognized that the mathematics of fuzzy quantities are an application of possibility theory, an extension of interval analysis as well as of the algebra of many-values quantities (Young [27]).

Fuzzy interval extends and updates the overview of Dubois and Prade [6]. Several theoretical details and applications can be found e.g., in monographs of Kaufmann and Gupta [12], [13], and Mares [16]. In 1987, teher was a special issue of Fuzzy Sets and Systems (Dubois and Prade [7]) devoted to the fuzzy intervals domain, and more recently another one has appeared (Fullér and Mesiar [9]).

In real world applications we often need to deal with imprecise quantities. They can be results of measurements or vague statements, e.g. I have about 40 dollars in my pocket, she is approximately 170cm tall. In arithmetics we can use $a < x$ and $x < b$ inequalites to characterize such quantities, e.g. if I have about 40 dollars then my money is probably more than 35 dollars and less than 45 dollars.

Fuzzy numbers can also be used to represent imprecise quantities. Pliant numbers are created by *softening* the $a < x$ and $x < b$ inequalities, i.e. replacing the crisp characteristic function with two fuzzy membership functions and applying a fuzzy conjunction operator to combine the two functions. We refer to the softened inequalities as *fuzzy inequalities*.

We call the distending function corresponding to the $x < a$ interval the left side of the fuzzy number and denote it as δ_l. Similarly we refer to the distending function corresponding to the $x < b$ interval as the right side of the fuzzy number and denote it as δ_r.

We will use the following terminology: function representing the soft inequality called distending function. The word pliant means flexible instead of using distending we use soft inequalities and additive pliant is when $f_c(x) + f_d(x) = 1$ (at nilpotent operator case) and multiplicative pliant if $f_c(x)f_d(x) = 1$ (at strict monotone operator case).

Naturally one would like to execute arithmetic operations over fuzzy numbers. Fuzzy arithmetic operations are generally carried out using the α-cut method. In Section II we propose a new and efficient method for arithmetic calculations. The next two sections discuss the arithmetic operations and their properties for two classes of fuzzy distending functions. Section II-A investigates additive pliant functions, i.e. distending functions represented as lines. Section II-B presents multiplicative pliant functions, i.e. distending functions based on pliant inequalities.

In our article the Pliant defuzzification method is based on fuzzy arithmetics.

II. Fuzzy arithmetics

Fuzzy arithmetic operations are based on the extension principle of arithmetics. In arithmetics we can find the result of an arithmetic operation by measuring the distance of the operands from the zero point than applying the operation on these distances. Fig. 1 presents this idea in case of addition.

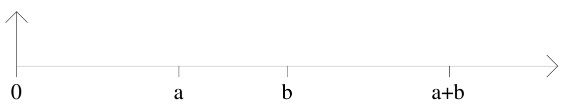

Fig. 1. Arithmetic addition

In fuzzy arithmetics we deal with fuzzy numbers. Fuzzy numbers are mappings from real numbers to the $[0, 1]$ real interval. Operations are executed by creating an α-cut for all $\alpha \in [0, 1]$ and using the arithmetic principle to get the resulting value for each α value. Fig. 2 demonstrates fuzzy addition with fuzzy numbers represented as lines. The dotted triangle number is the sum of the two other triangle numbers.

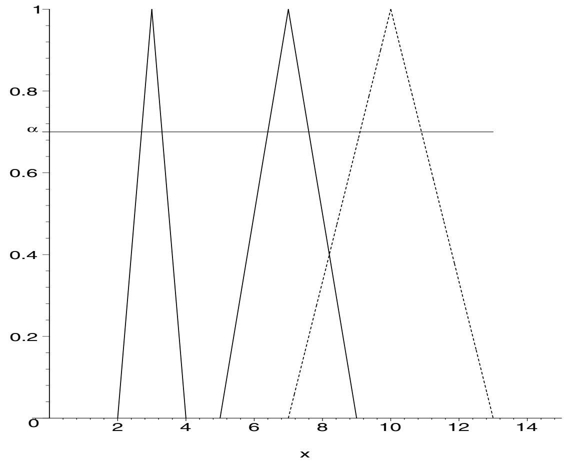

Fig. 2. Fuzzy addition with α-cut.

This way we can have all the well-known unary ($-x$, x^2) and binary operations ($x + y$, xy, $x \, mod \, y$) available as fuzzy operations. However the calculation of fuzzy operations with α-cut is tedious and often impractical. In this paper a new efficient method is proposed which is equivalent with the α-cut.

Fuzzy numbers are often composed of two strictly monotone functions, i.e. the left side denoted as δ_l, and the right side denoted as δ_r of the fuzzy number. Fuzzy operations can be carried out by first applying them to the left sides than to the right sides of the operands.

This separation allows us to treat fuzzy numbers as strictly monotone functions when dealing with fuzzy arithmetic operations. In the following we omit the subscript from δ_l and δ_r and simply write δ with the inherent assumption that we shall only do arithmetic operations with functions representing the same side of fuzzy numbers.

Lemma 2.1: Let $\delta_1, \delta_2, \ldots, \delta_n$ ($n \geq 1$) be strictly monotone functions representing soft inequalities and let F be an n-ary fuzzy operation over them. If

$$\delta = F(\delta_1, \delta_2, \ldots, \delta_n),$$

then

$$\delta(z) = \left(F\left({\delta_1}^{-1}, {\delta_2}^{-1}, \ldots, {\delta_n}^{-1}\right)\right)^{-1}(z) \iff \quad (1)$$
$$\delta(z) = \sup_{F(x_1, x_2, \ldots, x_n) = z} \min\left\{\delta_1(x_1), \delta_2(x_2), \ldots, \delta(x_n)\right\}.$$

Proof: [3] ■

Theorem 2.2: Let $\delta_1, \delta_2, \ldots, \delta_n$ ($n \geq 1$) be strictly monotone functions representing fuzzy inequalities and let F be an n-ary fuzzy operation over them. If

$$F\left({\delta_1}^{-1}, {\delta_2}^{-1}, \ldots, {\delta_n}^{-1}\right)$$

is strictly monotone then F has all the properties as its non-fuzzy interpretation.

Proof: [3] ■

A. Additive Pliant

Triangle fuzzy numbers are commonly used to represent approximate values. A triangle fuzzy number has one line on each side. We can add triangle fuzzy numbers by first adding their left lines and than adding their right lines together.

Lemma 2.1 let us derive a general formula for adding lines.

Definition 2.3: We say that a line $l(x)$ is given by its mean value if

$$l(x) = m(x-a) + \frac{1}{2}$$

as shown in Fig. 3.

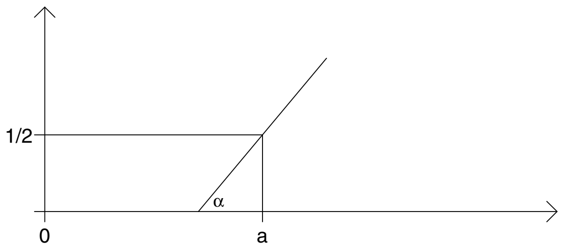

Fig. 3. Line given by its mean value a and tangent $m = \tan \alpha$.

The inverse of $l(x)$ denoted as $l^{-1}(y)$ can be calculated easily

$$l^{-1}(y) = \frac{y - \frac{1}{2}}{m} + a.$$

1) Addition:

Theorem 2.4: Let $l_i(x) = m_i(x - a_i) + \frac{1}{2}$ ($i \in \{1, \ldots, n\}$) lines given by their mean values. The fuzzy sum of l_i lines denoted as l is also a line and can be given as

$$l(x) = l_1(x) \oplus \ldots \oplus l_n(x) = m(x-a) + \frac{1}{2}$$

where

$$\frac{1}{m} = \sum_{i=1}^{n} \frac{1}{m_i} \quad \text{and} \quad a = \sum_{i=1}^{n} a_i.$$

Proof: [3] ■

2) Multiplication by scalar:

Theorem 2.5: Let

$$l(x) = m(x-a) + \frac{1}{2}$$

line given by their mean values.

The scalar multiplication of the lines is:

$$c \odot l(x) = m'(x - a') + \frac{1}{2}$$

where

$$a' = ca \qquad m' = \frac{m}{c}$$

Proof: [3] ■

3) Properties of Operations:

Theorem 2.6: Addition is commutative and associative over lines.

Proof: The properties can be easily seen from the construction of $\frac{1}{m}$ and a in Theorem 2.4. ■

Theorem 2.7: Multiplication over lines is commutative, associative and distributive over addition.

Proof: Theorem 2.2 guarantees that these properties holds. ■

B. Multiplicative Pliant

Let us start by introducing a special fuzzy inequality, the *pliant inequality* and examine its most important properties.

1) Pliant Inequality Model:

Definition 2.8: A pliant inequality is given as a sigmoid function of

$$\{a <_\lambda x\} = \frac{1}{1 + e^{-\lambda(x-a)}} = \sigma_a^{(\lambda)}(x)$$

where a is the mean value, i.e. $\sigma_a^{(\lambda)}(a) = \frac{1}{2}$.

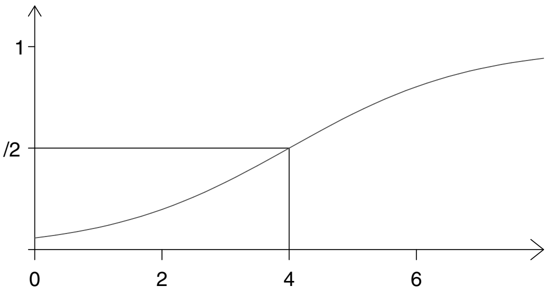

Fig. 4. Pliant inequality with $\lambda = 0.7$ and $a = 4$ parameters.

The following properties can be seen from the figure

$$\begin{aligned} a < x & \quad then \quad \{a <_\lambda x\} > \frac{1}{2}, \\ a = x & \quad then \quad \{a <_\lambda a\} = \frac{1}{2}, \\ a > x & \quad then \quad \{a <_\lambda x\} < \frac{1}{2}. \end{aligned}$$

Definition 2.9: The first derivative of $\sigma_a^{(\lambda)}(x)$ is denoted as $\left(\sigma_a^{(\lambda)}(x)\right)'$. The following properties hold

$$\left(\sigma_a^{(\lambda)}(a)\right)' = \frac{d\sigma_a^{(\lambda)}(x)}{dx}\bigg|_{x=a} = 4\lambda;$$

depending on λ, if

$$\begin{aligned} \lambda > 0 & \quad then \quad \left(\sigma_a^{(\lambda)}(x)\right)' \text{ is strictly monotone increasing,} \\ \lambda = 0 & \quad then \quad \left(\sigma_a^{(\lambda)}(x)\right)' \equiv 0, \\ \lambda < 0 & \quad then \quad \left(\sigma_a^{(\lambda)}(x)\right)' \text{ is strictly monotone decreasing.} \end{aligned}$$

When we apply an arithmetic operation to pliant inequalities we need to make sure that the operation is meaningful, i.e.

the pliant inequalities represent the same sides of the fuzzy numbers. The following criteria formulates this requirement.

Criteria 2.10: If $\sigma_{a_1}^{(\lambda_1)}, \sigma_{a_2}^{(\lambda_2)}, \ldots, \sigma_{a_n}^{(\lambda_n)}$ are inputs to an n-ary fuzzy arithmetic operation then

$$\text{sgn}(\lambda_1) = \text{sgn}(\lambda_2) = \ldots = \text{sgn}(\lambda_n)$$

must always hold.

2) Addition:

Theorem 2.11: Addition is closed over pliant inequalities and the addition function can be given as

$$\sigma_{a_1}^{(\lambda_1)} \oplus \ldots \oplus \sigma_{a_n}^{(\lambda_n)} \quad = \quad \sigma_a^{(\lambda)} \qquad n \geq 1$$

where

$$\frac{1}{\lambda} = \sum_{i=1}^{n} \frac{1}{\lambda_i} \quad \text{and} \quad a = \sum_{i=1}^{n} a_i.$$

Proof: [3] ■

3) Multiplication by scalar:

Theorem 2.12: Let given $\sigma_a^{(\lambda)}(x)$ sigmoid function.

The scalar multiplication of the sigmoid function is:

$$c \odot \sigma_a^{(\lambda)}(x) = \sigma_{a'}^{(\lambda')}(x)$$

where

$$\lambda' = \frac{\lambda}{c} \qquad a' = ca$$

Proof: [3] ■

III. Computational methods for the pliant defuzzification

A. Pliant defuzzification and related methods

The kernel of a linguistic fuzzy model [1], [15] is the rule base consisting of r rules of the following form:

$$\begin{aligned} R_s : IF\ x_1\ IS\ B_s^1, x_2\ IS\ B_s^2, \ldots, \\ x_m\ IS\ B_s^m\ THEN\ y\ IS\ C_s, \end{aligned} \tag{2}$$

where B_s^l (resp. C_s) are linguistic values of variable X_l (resp. Y) described by membership functions B_s^l (resp. C_s) and $x = [x_1, x_2, \ldots, x_m]$ are the input values ($s = 1, \ldots, r$ and $l = 1, \ldots, m$).

B. Pliant defuzzification

The most simple rule can be described in the following way:

$$\text{if} \quad L_i(\mathbf{x}) \quad \text{then} \quad y_i,$$

where $L_i(x)$ a logical (fuzzy) expression, $y_i \in R^+$, i.e. y is not "fuzzy".

We define the validity of the rules in the following way.

$$w_i(\mathbf{x}^*) = \frac{L_i(\mathbf{x}^*)}{\sum_{i=1}^{n} L_i(\mathbf{x}^*)}, \qquad i = 1, \ldots N.$$

We note that $w_i(\mathbf{x}^*) \geq 0$ and $\sum_{i=1}^{n} w_i(\mathbf{x}^*) = 1$.

The most natural way to define y at $\mathbf{x}^*$ is:

$$y(x^*) = \sum_{i=1}^{n} w_i(\mathbf{x}_i^*) y_i.$$

In pliant system y_i are replaced by a fuzzy numbers which have a left- and righthand sided fuzzy membership function. (In pliant system terminology we call then left- and righthand sided soft inequality.)

At pliant defuzzification we calculate for all rules the weighted arithmetic sum for the threshold and slope. Similar way we do the same for the right sided inequality.

Pliant defuzzification is the intersection of two resulted soft inequality which is a direct calculation.

IV. Experimental results

We used the MATLAB Simulink environment to compare Fuzzy Logic ToolBox capabilities and the Pliant Controller in two well-known example: the water level control in a tank, and the inverted pendulum.

A. Water level control in a tank

The reference model for this test was the fuzzy demo called "sltank" in MATLAB 2008b. The object in this example is to set the water level in a tank using a valve. The desired water level is a square wave and the valve is bidirectional. This demo contains a fuzzy controller capable of solving this problem with 5 rules defined in it. To make this fuzzy structure Pliant compatible the membership functions were changed to sigmoid ones, and the defuzzification process was changed according to previous section.

Figure 5 shows the desired water level (blue), the result of fuzzy controller (green) and the result of Pliant controller (red).

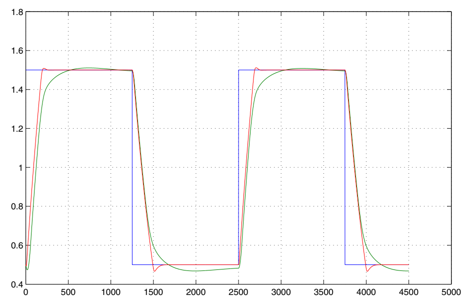

Fig. 5. Water level control performance comparison

B. Inverted pendulum

For the inverted pendulum (also known as cart and pole) problem we used the "slcp" demo application from MATLAB 2008b. The goal in this experiment is to maintain the rod in upward position while changing the cart position to a desired point. The task is difficult because the rod is unactuated. The "slcp" demo uses a fuzzy controller with 16 rules, while we used the simplest possible pliant controller with 2 rules to outperform the previous one. For fair comparison under same conditions both controller could use the same peak level force and the sum of used unsigned force had to be equal. Figure 6 shows the desired position (blue), real position using fuzzy controller (red) and using pliant controller (green). Figure 7 shows the used force for both controller (fuzzy controller is blue and pliant controller is green).

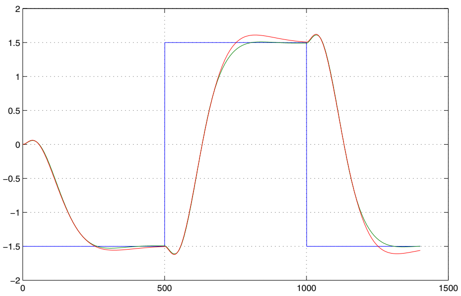

Fig. 6. Inverted pendulum performance comparison

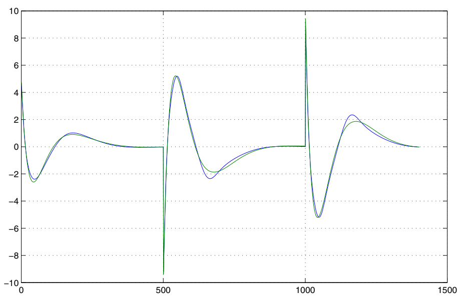

Fig. 7. Comparison of used force in inverted pendulum

C. Summary of simulation results

Table I. shows performance comparision for the Sltank (water tank demo) with 5 rules Fuzzy controller and 5 rules Pliant controller, while Table II. shows performance comparision for Slcp (inverted pendulum demo) with 16 rules Fuzzy controller and 2 rules Pliant controller.

	Fuzzy controller (5 rules)	Pliant controller (5 rules)
Sltank demo	11.1383 m	8.3091 m

TABLE I
SUM OF ABSOLUTE ERROR

	Fuzzy controller (16 rules)	Pliant controller (2 rules)
Slcp demo	10.7459 m	10.5883 m

TABLE II
SUM OF ABSOLUTE ERROR

Table III. shows runtime comparision for the 2 rules and 5 rules controllers evaluating with Fuzzy controller using COG and PCS. The values are the runtime for 1000 evaluation with random numbers on a standard desktop PC. Note that the Fuzzy controller implementation with COG defuzzification uses highly optimized architecture specific, while the PCS implementation uses suboptimal Matlab code. The real difference could be in orders of magnitude.

	Fuzzy controller	Pliant controller
5 rules (Sltank)	0.475837 s	0.355336 s
2 rules (Slcp)	0.442201 s	0.258255 s

TABLE III
RUNTIME COMPARISON

V. CONCLUSION

We can conclude that the PCS using presented defuzzification method is computationally efficient compared to the COG method. Other advantage for this technique is it enables simpler fuzzy structure for easier development and maintanance.

VI. ACKNOWLEDGEMENTS

This research was partially supported by the TÁMOP-4.2.2/08/1/2008-0008 program of the Hungarian National Development Agency.

REFERENCES

[1] S. Assilian. *Artificial intelligence in the control of real dynamical systems*. Ph.D. Thesis, London University, Great Britain, 1974.

[2] E. Van Broekhoven and B. De Baets. Fast and accurate center of gravity defuzzification of fuzzy system outputs defined on trapezoidal fuzzy partitions. *Fuzzy Sets and Systems*, Vol. 157, pp. 904–918, 2006.

[3] J. Dombi. Pliant Arithmetics and Pliant Arithmetic Operations. *Acta Polytechnica Hungarica*, 6:5, 19–49, 2009.

[4] D. Driankov, H. Hellendoorn and M. Reinfrank. *An Introduction to Fuzzy Control*. Springer, Berlin, 1993.

[5] D. Dubois and H. Prade. Operations on fuzzy numbers. *Int. J. Systems Science*, 9:613–626, 1978.

[6] D. Dubois and H. Prade. Fuzzy members: An overview. *Analysis of Fuzzy Information*, Vol. I., CRC Press, Boca Raton, FL, 3–39, 1987.

[7] D. Dubois and H. Prade. Special Issue on Fuzzy Numbers. *Fuzzy Sets and System*, 24(3), 1987.

[8] D. Filev and R.Yager. A generalized defuzzification method via BAD distributions. *Internat. J. Intell. Systems*, Vol. 6, pp. 689–697, 1991.

[9] R. Fullér and R. Mesiar. Special Issue on Fuzzy Arithmetic. *Fuzzy Sets and System*, 91(2)., 1997

[10] R. Jain. Tolerance analysis using fuzzy sets. *Int. J. System Science*, 7: 1393–1401, 1976.

[11] T. Jiang and Y. Li. Generalized defuzzification strategies and their parameter learning procedures. *IEEE Trans. Fuzzy Systems*, Vol. 4, pp. 64–71, 1996.

[12] A. Kaufmann and M. M. Gupta. *Introduction to Fuzzy Arithmetic-Theoryand Applications.* Van Nostrand Reinhold, New York, 1985.

[13] A. Kaufmann and M. M. Gupta. *Fuzzy MAthematical Models in Engineering and Management Science.* North-Holland, Amsterdam, 1988.

[14] W. Van Leekwijck and E. Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, Vol. 108, pp. 159–178, 1999.

[15] E. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Proc. IEE*, Vol. 121, pp. 1585–1588, 1974.

[16] M. Mares. *Computation Over Fuzzy Quantities.* CRC Press, Boca Raton, FL, 1994.

[17] M. Mizumoto and K. Tanaka. The four operations of arithmetic on fuzzy numbers. *Syst. Comput. Controls*, 7(5): 73–81, 1976.

[18] M. Mizumoto and K. Tanaka. Algebraic properties of fuzzy numbers. *Proc. Int. Conf. On Cybernetics and Society*, Washington, DC, 559–563, 1976.

[19] S. Nahmias. Fuzy variables. *Fuzzy Sets and System*, 1:97–110, 1978.
[20] H. T. Nguyen. A note on the extension principle for fuzzy sets. *J. Math. Anal. Appl.*, 64:369–380, 1978.
[21] A. Patel and B. Mohan. Some numerical aspects of center of area defuzzification method. *Fuzzy Sets and Systems*, Vol. 132, pp. 401–409, 2002.
[22] S. Roychowdhury and B.-H.Wang. Cooperative neighbors in defuzzification. *Fuzzy Sets and Systems*, Vol. 78, pp. 37–49, 1996.
[23] S. Roychowdhury and W. Pedrycz. A survey of defuzzification strategies. *Internat. J. Intell. Systems*, Vol. 16, pp. 679–695, 2001.
[24] A. Sakly and M. Benrejeb. *Activation of trapezoidal fuzzy subsets with different inference methods*. In: T. Bilgi, B. De Baets, O. Kaynak (Eds.), Fuzzy Sets and Systems - Proc. IFSA 2003, Lecture Notes in Artificial Intelligence, Vol. 2715, Springer, Berlin, 2003, pp. 450–457.
[25] Q. Song and R. Leland. Adaptive learning defuzzification techniques and applications. *Fuzzy Sets and Systems*, Vol. 81, pp. 321–329, 1996.
[26] R. Yager and D. Filev. SLIDE: a simple adaptive defuzzification method. *IEEE Trans. Fuzzy Systems*, Vol. 1, pp. 69–78, 1993.
[27] R. C. Young. The algebra of many-valued quantities. *Math. Ann.*, 104:260–290, 1931.
[28] L. A. Zadeh. The concept of a linquistic variable and its application to approximate reasoning. *Information Sciences*, Part I:8, 199–249, Part II:8, 301–357, Part III:9, 43–80, 1975.