ARITHMETIC-BASED FUZZY CONTROL

J. DOMBI AND T. SZÉPE

Abstract. Fuzzy control is one of the most important parts of fuzzy theory for which several approaches exist. Mamdani uses α-cuts and builds the union of the membership functions which is called the aggregated consequence function. The resulting function is the starting point of the defuzzification process. In this article, we define a more natural way to calculate the aggregated consequence function via arithmetical operators. Defuzzification is the optimum value of the resultant membership function. The left and right hand sides of the membership function will be handled separately. Here, we present a new ABFC (Arithmetic Based Fuzzy Control) algorithm based on arithmetic operations which use a new defuzzification approach. The solution is much smoother, more accurate, and much faster than the classical Mamdani controller.

1. Introduction

One of the most successful developments of fuzzy reasoning is the design of fuzzy linguistic control systems. A control system is based on a set of 'if-then' rules. Currently, the fuzzy control approach is mainly concerned with model-based methods. The main idea of reasoning is the modus ponens for forward reasoning and modus tollens for backward reasoning. These processes are referred to the literature as inference. The kernel of a linguistic fuzzy model [1, 13] is the rule base consisting of r rules that has the following form:

$$IF\ x_1\ IS\ P_s^1, x_2\ IS\ P_s^2, \dots, x_m\ IS\ P_s^m\ THEN\ y\ IS\ Q_s,$$

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

1.1. Introduction to Defuzzification.

As most modeling 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 are used, such as the parametric BAD (basic defuzzification) distribution and SLIDE

Received: December 2015; Revised: August 2016; Accepted: December 2016
Key words and phrases: Fuzzy controller, Mamdani controller, Defuzzification, Fuzzy arithmetic.

(semi-linear defuzzification) methods of Yager and Filev [7, 27]. 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 [10,20,23]. However, this falls outside the scope of the present paper. Note that in the literature the terms for describing the different defuzzification methods vary from author to author. The terms 'center of gravity' defuzzification, 'center of area' defuzzification and 'center of sum' defuzzification, for instance, refer to different methods by some authors and are used as synonyms by other authors. Therefore, we should focus on the formal definitions of the defuzzification methods rather than their names. In [21, 26], comprehensive overviews are given for defuzzification methods. In this paper, the same terminology is used as in [26].

In the paper [25], the authors present 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 that form a fuzzy partition.

Here, we focus on the computational aspects of the Pliant defuzzification method. When applying the new method, the crisp output y^* of the system will change continuously when the input values vary continuously, which is a desirable property in modeling and control applications. However, the Pliant defuzzification method has a high computational burden [3, 19], which is a significant disadvantage in control and model identification, and in tuning applications. This high cost is often circumvented by introducing new defuzzification methods that seek to approximate the center of gravity [19, 22].

COG defuzzification is the most widely implemented approach, as it produces a very smooth output. To calculate it, the following integration should be performed in a numerical way and it is CPU intensive:

$$z_0 = \frac{\int \mu_i(x) x dx}{\int \mu_i(x) dx},$$

where z_0 is the defuzzified output, μ_i is a membership function and x is an output variable.

The bisector (BIS) defuzzification simply divides the aggregated membership function area into two equal areas at point z_0:

$$\int_{\alpha}^{z_0} \mu_A(x) dx = \int_{z_0}^{\beta} \mu_A(x) dx.$$

The MOM defuzzification method calculates the average value of the locations of the maximum value of the aggregated membership function:

$$z_0 = \frac{\int_{x'} x dx}{\int_{x'} dx'},$$

where $x' = (x, \mu_A(x) = \max(\mu_A))$.

The smallest of maxima (SOM) defuzzification method can be calculated as follows:

$$z_0 = \min(x'),$$

where $x' = (x, \mu_A(x) = \max(\mu_A))$.

The largest of maxima (LOM) defuzzification method can be calculated as follows:

$$z_0 = \max(x'),$$

where $x' = (x, \mu_A(x) = \max(\mu_A))$.

1.2. **Regression and Control.** Here, our approach is different from the usual one, and it is similar to the regression approach. The regression procedure has three main steps. The first step is to find a proper model based on given data such as:

$$\begin{gathered} \text{if } x_1 \text{then } y_1 \\ \text{if } x_2 \text{then } y_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \text{if } x_n \text{then } y_n. \end{gathered} \tag{1}$$

The second step is the application of the developed model.

The third one is if x^* is the input, then we use the model to calculate y^*.

We can see that there is a close correspondence between regression methods and fuzzy control, although we do not wish to translate the rule-based system into regression calculus. Our first observation that (1) is an implication system which describes the case where x_i appears and then y_i also appears i.e. the two events are coupled to each other. We are convinced that a similar case is also true for the fuzzy control system. Our second observation is that a continuous-valued logical expression is in the fuzzy control after 'if', which returns a numerical value instead of an uncertainty condition.

The concept is similar to the Takagi-Sugeno model [24]. The main difference is that we do not have a single value as an aggregated consequence, but a fuzzy number. Here in this article on one hand we keep the Takagi-Sugeno concept for computational efficiency and on the other hand we use the Mamdani extension, i.e. on the right hand side they are fuzzy numbers.

Let $\mathcal{L}_i(\mathbf{x})$ be the i-th logical expression. Then a rule system can be formalized in the following way:

$$\begin{gathered} \text{if } \mathcal{L}_1(\mathbf{x}) \text{then } y_1 \\ \text{if } \mathcal{L}_2(\mathbf{x}) \text{then } y_2 \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ \text{if } \mathcal{L}_n(\mathbf{x}) \text{then } y_n. \end{gathered} \tag{2}$$

The application of this rule system is the following. For any given $\mathbf{x}^*$ input, the α_i can be calculated as $\mathcal{L}_i(\mathbf{x}^*) = \alpha_i$.

$$\begin{gathered} \alpha_1,\ y_1 \\ \alpha_2,\ y_2 \\ \cdots\cdots \\ \alpha_n,\ y_n. \end{gathered} \tag{3}$$

But what is α_i? In the Mamdani model α_i is an α-cut at the consequence of the i-th rule. The interpretation of α_i is the 'applicability' of i-th rule. Here in this article we normalize α_i in the following way:

$$w_i = \frac{\alpha_i}{\sum\limits_{i=1}^{n} \alpha_i} \qquad If \sum \alpha_i \neq 0. \tag{4}$$

If $\sum\limits_{i=1}^{n} \alpha_i = 0$, then we cannot infer anything.

The output in this case is

$$y^* = \sum_{i=1}^{n} w_i y_i. \tag{5}$$

1. The value of y^* is the generalized arithmetic mean of y_i-s and the weights are the truth values of the antecedent.
2. We can interpret y^* as an expected value, where the y_i-s are the values and the w_i-s are the probability values which are determined by the rules and express its probability or applicability.
3. In fuzzy control, y^* can be interpreted as the COG based on the terminology used in Mamdani defuzzification.

We notice in the above-mentioned method that there is no room for implication. We only use arithmetic operations! Now, let us turn to the Mamdani model. The only change that we can make is that the consequent part is not a value as in (1) and (2), but fuzzy sets i.e. y_i are membership functions. To generalize the approach, the weighted arithmetic mean of the membership functions has to be calculated. Later on we will see it is either a triangular or trapezoid function, and this calculation is very simple.

In our new ABFC method, the linear combination of the consequent of the rules will be used, where the weights are the normalized values of the antecedents.

To achieve this, first an effective algorithm needs to be developed to calculate the linear combination of the membership functions.

The outline of the paper is as follows: A novel approach of Fuzzy arithmetical operations will be introduced in Chapter 2. Here we use the advantages of the new formulas for the membership functions, which result in a simplification of the linear combination of fuzzy numbers. Based on this fuzzy arithmetic concept, a new defuzzyfication method will be introduced in Chapter 3, where a direct calculation can be applied. Lastly, the advantages of this approach will be elaborated on in Chapter 4.

2. Fuzzy Arithmetic: A New Concept

The suggestion that fuzzy quantities could be arithmetically combined based on the laws of fuzzy set theory is due to Zadeh [29]. Soon after, several researchers worked independently along these lines, like Jain [9], Mizumoto and Tanaka [15, 16], Nahmias [17], Nguyen [18], Dubois and Prade [4]. It was only later realized that the mathematics of fuzzy quantities is an application of possibility theory, an extension of interval analysis and extension of the algebra of many-values quantities (Young [28]). Fuzzy interval theory extends and updates the overview of Dubois and Prade [5]. Theoretical details and applications can be found e.g. in monographs by Kaufmann and Gupta [11,12], and Mares [14]. In 1987, there was a special issue of Fuzzy Sets and Systems. Dubois and Prade ([6]) focused on the fuzzy intervals domain, and later another one appeared (Fullér and Mesiar [8]). Fuzzy arithmetic-based α-cuts are where the result of an α-cut is represented by an interval. The arithmetics can be understood as an interval arithmetics of the α-cuts.

In fuzzy arithmetic theory we deal with fuzzy numbers, which 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 interval arithmetic principle to get the resulting value for each α value.

Instead of dealing with intervals, we deal with left and right-hand sided soft inequalities that define the interval. Here we present a new calculation procedure of arithmetic, where these soft inequalities have certain properties (i.e. the inequality is represented by a strict monotonously increasing function). We show that the results of linear combinations are also linear (i.e. they are closed under linear combination). We give the result of other operations as well. The soft inequalities define an interval by using a proper conjunctive operator and disjunctive operator. Here, we will present these operations as well.

However, the calculation of fuzzy operations with the α-cut is tedious and often impractical. Here a new, efficient method is presented which is equivalent to the α-cut. More detailed properties and proofs can be found in ref [2]

2.1. **Arithmetics Based on Inverse Functions.** Fuzzy numbers are often composed of two strictly monotone functions, i.e. the left-hand side denoted by μ_l, and the right-hand side denoted by μ_r of the fuzzy number. Fuzzy operations can be carried out by first applying them to the left-hand sides then to the right hand sides of the operands.

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

Next, we introduce a new concept called the distending function.

Definition 2.1. A distending function is a measurement that expresses the truth of an inequality. That is,

$$\delta_a(x) = truth(a < x).$$

Lemma 2.2. *Let $\mu_1, \mu_2, \ldots, \mu_n$ ($n \geq 1$) be strictly monotone functions representing fuzzy inequalities and let F be an n-ary fuzzy operation over them. If*

$$\mu = F(\mu_1, \mu_2, \ldots, \mu_n),$$

then

$$\begin{aligned} \mu(z) &= \left(F\left({\mu_1}^{-1}, {\mu_2}^{-1}, \ldots, {\mu_n}^{-1}\right)\right)^{-1}(z) \\ \mu(z) &= \sup_{F(x_1,x_2,\ldots,x_n)=z} \min\left\{\mu_1(x_1), \mu_2(x_2), \ldots, \mu(x_n)\right\}. \end{aligned} \tag{6}$$

Proof. It can be readily verified that the method is equivalent to the α-cut. □

We can state a theorem for the properties of fuzzy operations.

Theorem 2.3. *Let $\mu_1, \mu_2, ..., \mu_n$ $(n \geq 1)$ be strictly monotone functions representing fuzzy inequalities and let F be an n-ary fuzzy operation over them. If*

$$F(\mu_1^{-1}, \mu_2^{-1}, ..., \mu_n^{-1})$$

is strictly monotone, then F has the same properties as its non-fuzzy counterpart.

Proof. This can be derived using (2). □

2.2. **Calculation of the Linear Combination.** We will suppose that the consequent is a triangle or a trapezoid membership function.

The equations of the left- and right-hand sides have the following forms:

$$y = m_l(x - a_l) + \frac{1}{2}$$
$$y = m_r(x - a_r) + \frac{1}{2}, \tag{7}$$

where $m_l > 0$, $m_r < 0$ and $-\infty < a_l < a_r < \infty$.
If the trapezoid is a rectangle, then $m = \infty$ and the equation is

$$y(x) = \begin{cases} \frac{1}{2}, & \text{if } x = a \\ 0, & \text{if } x < a \\ 1, & \text{if } x > a \end{cases} \tag{8}$$

The interpretation of equations (7) and (8) is $truth(a < x)$, i.e. the membership function is a soft inequality.

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-hand lines and then add their right-hand lines together. Lemma (2.2) allows one to derive a general formula for adding lines.

Definition 2.4. We say that a line $l_a^{(m)}(x)$ is given by its mean value if

$$l_a^{(m)}(x) \quad = \quad \max\left(\min\left(m(x-a) + \frac{1}{2}, 0\right), 1\right) = [a <_m x].$$

Here, we introduce a new notation:

$$[x] = \max\left(0, \min(x, 1)\right) = \begin{cases} 1 & \text{if } , x > 1, \\ x & \text{if } , 0 < x < 1, \\ 0 & \text{if } , x > 0. \end{cases}$$

The following properties holds.

$$if a < x, \ then \ [a <_m x] > \frac{1}{2},$$

$$if a = x, \ then \ [a <_m x] = \frac{1}{2},$$

$$if a > x, \ then \ [a <_m x] < \frac{1}{2}.$$

The interpretation of $l_a^{(m)}(x)$ is

$$l_a^{(m)}(x) = truth(a <_m x) = [a <_m x].$$

The inverse of $l_a^{(m)}(x)$ denoted by $l^{-1}(y)$ can be calculated using the formula:

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

When we apply an arithmetic operation to Pliant inequalities we need to ensure that the operation is meaningful, i.e. the Pliant inequalities must represent the same sides of the fuzzy numbers. The following criteria encapsulates this requirement.

$$sgn(m_1) = sgn(m_2) = ... = sgn(m_n).$$

2.3. **Addition Based on Inverse Function.** In this section, we will show how to add two linear functions using the $\oplus$ operator.

Theorem 2.5. *Let $l_i(x) = m_i(x - a_i) + \frac{1}{2}$ ($i \in \{1, \ldots, n\}$) be lines given by their mean values. The fuzzy sum of l_i lines denoted by l is also a line and may be defined as*

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

where

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

Proof. Using Lemma 2.2, we have

$$\begin{aligned} l^{-1}(y) &= \left(l_1{}^{-1}(y) + \ldots + l_n{}^{-1}(y)\right) \\ &= \sum_{i=1}^{n} \left(\frac{y - \frac{1}{2}}{m_i} + a_i \right) = \sum_{i=1}^{n} \left(\frac{y - \frac{1}{2}}{m_i} \right) + \sum_{i=1}^{n} a_i = \\ &= \left(y - \frac{1}{2} \right) \sum_{i=1}^{n} \frac{1}{m_i} + \sum_{i=1}^{n} a_i \end{aligned}$$

From this, we find that

$$l(x) \quad = \quad \frac{1}{\sum_{i=1}^{n} \frac{1}{m_i}} \left(x - \sum_{i=1}^{n} a_i \right) + \frac{1}{2}.$$

Substituting $\frac{1}{m}$ and a into the equation, we get the desired result. Namely,

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

□

2.4. **Linear Combination Based on an Inverse Function.** Now we will compute a linear combination of two linear functions.
The following general theorem is valid.

Proposition 2.6. *The linear combination of the left (right)-hand sides is linear. The result is*

$$y = m(x-a) + \frac{1}{2}, \tag{9}$$

where

$$a = \sum w_i a_i, \tag{10}$$

and

$$\frac{1}{m} = \sum_{i=1}^{n} \frac{w_i}{m_i}. \tag{11}$$

Proof. The proof is similar to those for the above theorems. □

3. New Defuzzification Method

The following proposition is valid.

Proposition 3.1. *The linear combination of a triangle or a trapezoid function is also a triangle or a trapezoid function.*

Proof. We have proved that the linear combinations of the left- and right-hand sides of the membership functions are also linear functions. Now, let the intersection of this function be at x_1, y_1. We have to prove that $y_1 > 0$. Let $\mu_A(x_{A_1}) = \alpha$, $\mu_A(x_{A_2}) = \alpha$, and $x_{A_1} < x_{A_2}$, and let $\mu_B(x_{B_1}) = \alpha$, $\mu_B(x_{B_2}) = \alpha$ and $y_{B_1} < y_{B_2}$.
$\mu(x_1^*) = \alpha$, $\mu(x_2^*) = \alpha$, where

$$x_1^* = w_1 x_{A_1} + w_2 y_{B_2} < w_1 x_{B_1} + w_2 x_{B_2} = x_2^*$$

and w_1 and w_2 are coefficients of this linear combination. □

After calculating the linear combination of the left- and right-hand sides of the resulting (aggregated) trapezoid, we get the following cases:

- The intersection is above 1; see Figure (1),
- The intersection is below 1; see Figure (2),
- The intersection is equal 1; see Figure (3),

The intersection cannot be below zero, and for why this is so, see the propositions above.
The main advantage of using the arithmetic concept is that the result generalizes a trapezoid-like function. In our model, we use a new type of defuzzification method. Here, two defuzzification methods are proposed.

1. **Let x_M be the intersection of the aggregated left- and right-hand sides. We call it the highest value of the trapezoid.**

$$x_M = \frac{m_1 a_1 - m_2 a_2}{m_1 - m_2}. \tag{12}$$

See Figures (1, 2, 3).

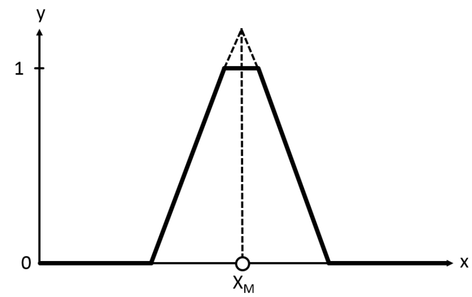

Figure 1. Trapezoid Membership

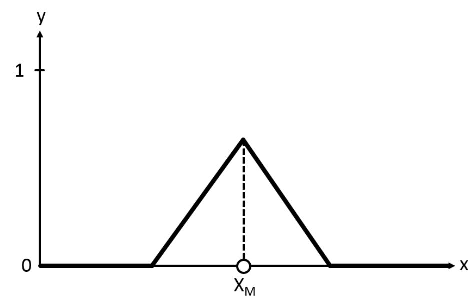

Figure 2. Triangle Function (Not Normalized) with $\max \mu(x) < 1$

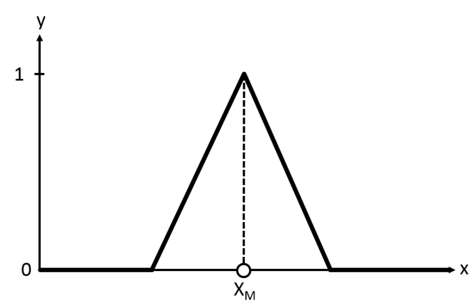

Figure 3. Triangle (Normalized) i.e.: max $\mu(x) = 1$

2. **Let x_S be the center point of the aggregated trapezoid. We call it the stable point of the trapezoid.**

$$x_S = \frac{1}{6}(c_1 - c_2)\frac{c_1 + 3b + c_2}{c_1 + 2b + c_2}. \tag{13}$$

The meaning of $a,\ b,\ c_1,\ c_2,\ x_S$ can be understood in Figure (4) .

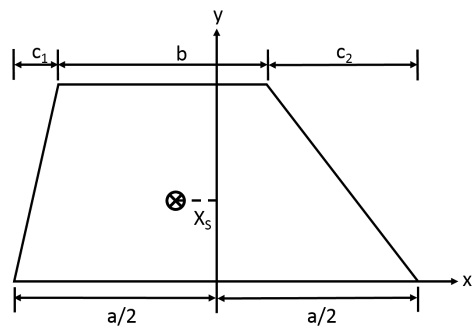

FIGURE 4. Stable Center of the Trapezoid

4. **Advantages of the New Method**

For an experimental comparison, a classical control demo was used to highlight the better performance of the ABFC method. The task to be solved consisted of a water tank with one pipe at the bottom and another at the top. The upper pipe is also equipped with a controllable valve that is capable of increasing the water level of the tank, while the pipe at the bottom only serves as an outlet. The purpose of the control is to maintain the desired water level in the tank by opening and closing the valve. This demo can be found in the Matlab Fuzzy Logic Toolbox. This controller is a classical Mamdani type. Here, the output was examined with different numerical resolutions. We observed significant degradation when the resolution decreased, but the speed increased slightly. The classical fuzzy controller was compared with the new ABFC method, as shown in Figure (5), where the two models are identical, except the controller box.

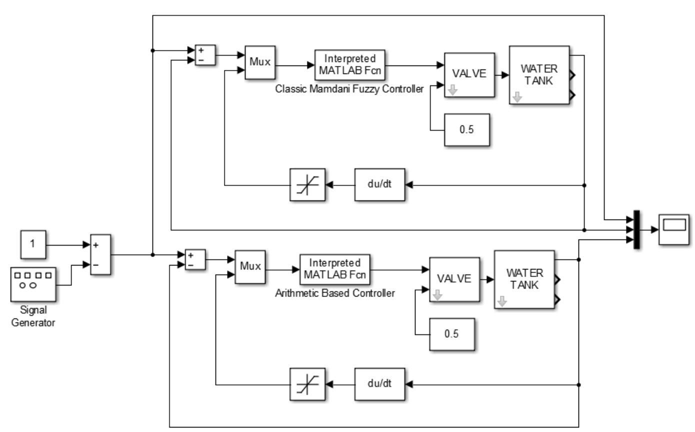

FIGURE 5. Matlab Simulink Model for the Mamdani and ABFC Methods

A more detailed analysis of the results can be found later on.

4.1. **Range Independence.** Range dependency is a general problem in the case of the classical Mamdani method with COG defuzzification, and it often leads to design errors. Here, the meaning of the range is the universe of discourse of the membership function of the consequence i.e. it should be limited in the Mamdani implementation. For example, this error occurs when the center of the triangular membership function is defined at the edge of the range. In the case of the ABFC method, the output is not affected by the range, hence the triangle will be precisely defuzzified. In Figure (6), the results of applying the ABFC method (•) and Mamdani methods (∘) have been plotted.

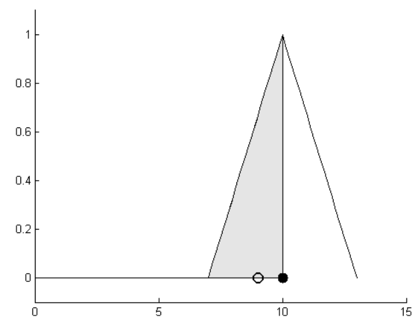

FIGURE 6. Range Independence

In the classical Mamdani method, a controversial case can also occur when two membership functions are separated on the output. The classical defuzzification method depends on the range of the control variable, as seen in Figure (7), where one of the membership functions is partly outside of the output range. The results of applying the ABFC method are denoted by • and the Mamdani COG method by ∘, and the aggregated output function by the grey region. The central trapezoid (dashed line) is a linear combination of the two output membership functions used by the ABFC method.

In the case of the classical Mamdani method, the output is not affected by the rightmost part of the aggregated function, so the output of the COG will shift to the left. The ABFC method does not need the range parameter, hence both trapezoids are treated in the same way during the defuzzification procedure.

4.2. **Effect of Fuzziness.** In Figure (8), we have a typical Mamdani situation where membership functions with different level of fuzziness need to be defuzzified. The value resulting from the classical Mamdani method (∘) is close to the center of the left triangular membership function, despite the fact that the triangle on the right-hand side is completely valid. The ABFC method intuitively considers the validity of both membership functions, so the result (•) is closer to the right triangle. The aggregated output function is denoted by the grey region, while the

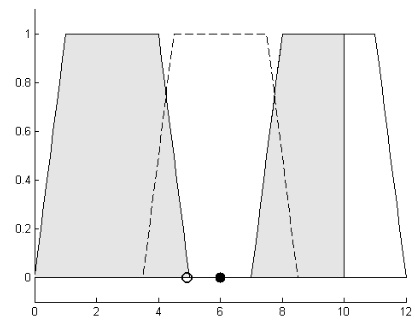

Figure 7. Controversial case

central triangle (dashed line) is a linear combination of the two output membership functions used by the ABFC method.

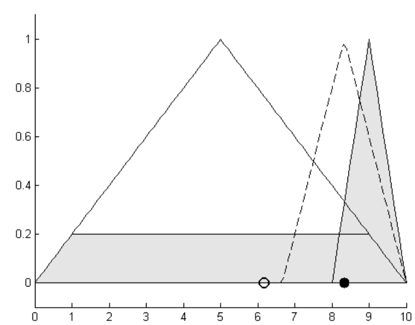

Figure 8. Effect of Fuzziness

4.3. **Efficiency of the Calculations.** In the classical Mamdani method, instead of an analytical integration, a numerical calculation is performed. The classical and the ABFC methods were compared in Figure (9) using different n numerical resolutions.

We can see that in our water tank example the difference between the approximate solution and the analytic solution increases as the number of numerical base values decreases. We compute the errors in the following way: first we choose a very high numerical resolution, i.e. we choose 100000. We suppose that this gives an ideal solution and also its value will be the reference for the calculation errors.

In Figure (9A), $n = 1000$, $\epsilon \approx 10^{-4}$,
In Figure (9B), $n = 100$, $\epsilon \approx 10^{-2}$,
In Figure (9C), $n = 25$, $\epsilon \approx 2$,
In Figure (9D), $n = 15$, $\epsilon \approx 8$.

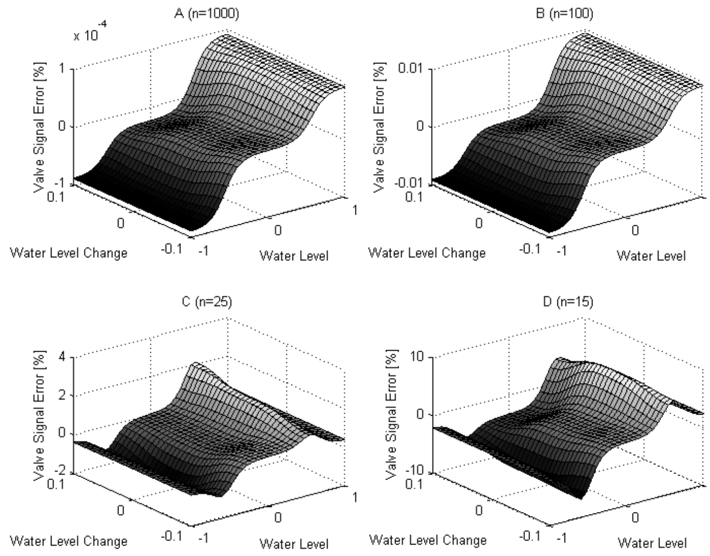

FIGURE 9. Membership Function Comparison

It seems to be valid that if $n \to \infty$, we get the same result as that got using the ABFC method. In this case, we can switch between the Mamdani and ABFC methods using the same rule base.

4.4. **Quality of the New Method.** We tested the water tank demo control, as described in Figure (5). In Figure (10) a typical testing procedure can be seen in response to a rectangular reference signal. The dashed line shows the classical Mamdani response with an $n = 15$ numerical resolution, while the solid line shows the response of the presented ABFC method with the same five rules. We observe that the overshooting error is much smaller using the ABFC method than that using the classical Mamdani case. The speed of the calculation is still 2.5 times faster than that using the ABFC method.

4.5. **Computational Speed.** We have seen that the quality of the new approach is better. Now we turn to the question of the efficiency. For a comparison, all values of the table are expressed in seconds. Here, all measurements were computed 10^7 times in a standard desktop environment. The algorithms were all implemented in C. First, the classical Mamdani methods (i.e. COG, BIS, SOM/LOM and MOM) were compared. Their speed depends on the numerical resolution, and not on the number of rules. Since the COG gives the best quality of calculation, it was used as a benchmark. Table 1 lists the calculation speeds of implication (imp), aggregation (aggr) and COG defuzzification with a different number of rules (3, 6, 9), and different numerical resolutions ($n = 101$, $n = 1001$, $n = 10001$). The

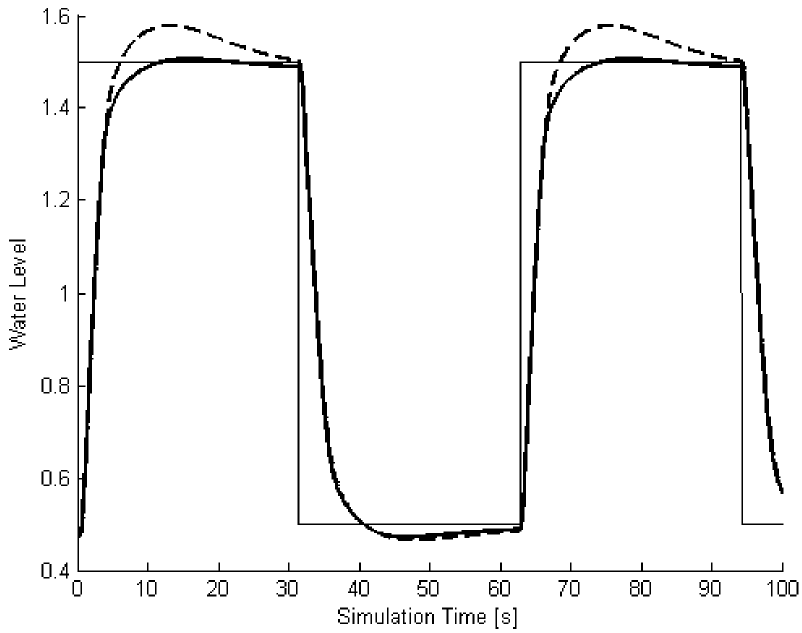

Figure 10. Water Tank Response of the Rectangular Pulse, with the Default Mamdani Controller (- -) and the ABFC Method (—)

	$n = 101$			$n = 1001$			$n = 10001$		
	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules
imp + aggr	0.154	0.326	0.481	1.688	3.852	5.885	24.268	47.711	66.952
COG	0.034	0.034	0.034	0.336	0.339	0.334	3.387	3.347	3.347
imp + aggr + COG	0.188	0.360	0.515	2.024	4.191	6.219	27.656	51.059	70.300

Table 1. Comparison of the Classical Defuzzification Approach

numerical resolution is the number of the elements in the universe of discourse, and the integral will be approximated based on it. This latter influenced all of the measured operation speeds, while the COG did not depend on the number of rules applied.

Here, we proposed two different defuzzification techniques for the ABFC method. Namely,

1. The maximal probability (Mprob); see equation (12).
2. The maximal stability (Mstab); see equation (13).

In Table 2, two approaches were compared with a number of rules (3, 6, 9). The calculation speed of Mprob and Mstab did not depend on the number of rules, which is similar to the case of classical defuzzification techniques. However, both proposed methods easily outperformed the classical COG.

	3 rules	6 rules	9 rules
ABFC Mprob	0.0008706	0.00086079	0.00086061
ABFC Mstab	0.00078055	0.00077053	0.00077054

Table 2. Comparison of the Two New Methods

	$n = 101$			$n = 1001$			$n = 10001$		
	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules	3 rules	6 rules	9 rules
imp + aggr + COG	0.188	0.360	0.515	2.024	4.191	6.219	27.660	51.059	70.300
ABFC aggr + MStab	0.012	0.022	0.033	0.011	0.022	0.033	0.012	0.022	0.033
(imp + aggr + COG) / (ABFC aggr + MStab)	**15.960**	**16.543**	**15.716**	**187.801**	**192.212**	**189.665**	**2344.781**	**2343.707**	**2142.582**

Table 3. The Speed-up of the ABFC Method Compared to That for the Classical Mamdani Concept

In Table 3, the speed-up of the new ABFC method versus the classical Mamdani concept is summarized. Here, the calculation speed of the whole ABFC method depends only on the number of rules. As a result, the ABFC method increasingly outperforms the classical solution as the numerical resolution increases and it always gives more accurate results. If the resolution is low ($n = 101$), the speed is 15 times faster. If the resolution is medium ($n = 1001$), the speed is 190 times faster. If the resolution is high ($n = 10001$), the speed is about 2300 times faster.

5. Conclusions

In this study, we presented a new arithmetic-based control approach. This new method is based on simple arithmetic operations instead of logical implication, which calculates the linear combination of the trapezoid membership functions. Also, two new defuzzification operations were introduced. These together form a new fuzzy controller, which outperforms the classical approach in terms of accuracy and speed. It is also range independent and intuitively handles different levels of fuzziness.

References

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

[2] J. Dombi, *Pliant arithmetics and pliant arithmetic operations*, Acta Polytechnica Hungarica, **6(5)** (2009), 19–49.

[3] D. Driankov, H. Hellendoorn and M. Reinfrank, *An introduction to fuzzy control*, Springer Science & Business Media, Berlin, 2013.

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

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

[6] D. Dubois and H. Prade, *Special issue on fuzzy numbers*, Fuzzy Sets and System, **24 (3)**, 1987.

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

[8] R. Fullér and R. Mesiar, *Special issue on fuzzy arithmetic*, Fuzzy Sets and System, **91(2)** (1997).

[9] R. Jain, *Tolerance analysis using fuzzy sets*, International Journal of Systems Science, **7(12)** (1976), 1393–1401.

[10] T. Jiang and Y. Li, *Generalized defuzzification strategies and their parameter learning procedures*, IEEE Trans. Fuzzy Systems, **4** (1996), 64–71.
[11] A. Kaufmann and M. M. Gupta, *Introduction to fuzzy arithmetic: theory and applications*, Van Nostrand Reinhold, New York, 1985.
[12] A. Kaufmann and M. M. Gupta, *Fuzzy mathematical models in engineering and management science*, North-Holland, Amsterdam, 1988.
[13] E. Mamdani, *Application of fuzzy algorithms for control of simple dynamic plant*, Proc. IEE, **121** (1974), 1585–1588, .
[14] M. Mares, *Computation over fuzzy quantities*, CRC Press, Boca Raton, FL, 1994.
[15] M. Mizumoto and K. Tanaka, *The four operations of arithmetic on fuzzy numbers*, Systems Comput. Controls, **7(5)** (1976), 73–81.
[16] M. Mizumoto and K. Tanaka, *Algebraic properties of fuzzy numbers*, Proc. Int. Conf. On Cybernetics and Society, Washington, DC, (1976), 559–563.
[17] S. Nahmias, *Fuzy variables*, Fuzzy Sets and System, **1** (1978), 97–110.
[18] H. T. Nguyen, *A note on the extension principle for fuzzy sets*, J. Math. Anal. Appl., **64** (1978), 369–380.
[19] A. Patel and B. Mohan, *Some numerical aspects of center of area defuzzification method*, Fuzzy Sets and Systems, **132** (2002), 401–409.
[20] S. Roychowdhury and B.-H.Wang, *Cooperative neighbors in defuzzification*, Fuzzy Sets and Systems,**78** (1996), 37–49.
[21] S. Roychowdhury and W. Pedrycz, *A survey of defuzzification strategies*, Internat. J. Intell. Systems, **16** (2001), 679–695.
[22] A. Sakly and M. Benrejeb, *Activation of trapezoidal fuzzy subsets with different inference methods*, International Fuzzy Systems Association World Congress, Springer Berlin Heidelberg, (2003), 450–457.
[23] Q. Song and R. Leland, *Adaptive learning defuzzification techniques and applications*, Fuzzy Sets and Systems, **81** (1996), 321–329.
[24] M. Sugeno, *Industrial Applications of Fuzzy Control*, Elsevier Science Publishers, New York, 1985.
[25] E. Van Broekhoven and B. De Baets, *Fast and accurate centre of gravity defuzzification of fuzzy system outputs defined on trapezoidal fuzzy partitions*, Fuzzy Sets and Systems, **157** (2006), 904–918.
[26] W. Van Leekwijck and E. Kerre, *Defuzzification: criteria and classification*, Fuzzy Sets and Systems, **108** (1999), 159–178.
[27] R. Yager and D. Filev, *SLIDE: a simple adaptive defuzzification method*, IEEE Trans. Fuzzy Systems, **1** (1993), 69–78.
[28] R. C. Young, *The algebra of many-valued quantities*, Math. Ann., **104** (1931), 260–290.
[29] L. A. Zadeh, *The concept of a linquistic variable and its application to approximate reasoning*, Information Sciences, **1(8)** (1975), 199–249.

József Dombi, Institute of Informatics, University of Szeged, Szeged, Hungary
E-mail address: `dombi@inf.u-szeged.hu`

Tamás Szépe*, Department of Technical Informatics, University of Szeged, Szeged, Hungary
E-mail address: `szepet@inf.u-szeged.hu`

*Corresponding author