

Reducing the Computational Requirements in the Mamdani-type Fuzzy Control

József Dombi¹, Edit Tóth-Laufer²

¹Faculty of Science and Informatics, University of Szeged, Aradi vértanúk 1, H-6720 Szeged, Hungary, dombi@inf.u-szeged.hu

²Bánki Donát Faculty of Mechanical and Safety Engineering, Óbuda University, Népszínház 8, H-1081 Budapest, Hungary, laufer.edit@bkgk.uni-obuda.hu

Abstract: Fuzzy-based control systems are popular in practical applications where imprecision, subjectivity, and uncertainty can arise in the data and in the evaluation process, and it needs to be addressed. One possible solution is the fuzzy approach. However, computational requirements of these models can be extremely high, mainly defuzzification part of the Mamdani-type control. Whatever great advantages the Mamdani model has, it is closer to the human way of thinking compared to the Sugeno model, which is another popular controller. In the case where a short reaction time is required, the computational needs should be reduced. Here, we propose modified Mamdani models for this purpose. This new model is much faster while all the advantageous properties of the original Mamdani model are retained, and in some ways, it is better.

Keywords: fuzzy control; Mamdani control; arithmetic-based control

1 Introduction

Fuzzy-based control models are frequently used in engineering tasks and in medical related applications [1]. Because of this, they can be built up in a user-friendly way, i.e. we can apply linguistic terms for the hard-to-quantify parameters. The Mamdani control is closer to the human way of thinking, but it has high computational requirements. For this reason, the application of the model in real-time and adaptive systems is limited. Computational requirements of the Takagi-Sugeno system is much lower. For this reason, it is very popular in optimization and adaptive tasks, but its limitation compared to the Mamdani control that intuition can be less built into the system. For the latter reason, it was not examined in detail in this paper, only a brief comparison is performed.

In the literature, there are several different techniques available for reduces the computational requirements in Mamdani-type control [2], such as the HOSVD-

based model, which eliminates the redundancy from the system [3]; genetic algorithms are used for generalization of the antecedents [4]; and, to minimize the number of rules and membership functions of the fuzzy system fuzzy subtractive clustering [5] or the combination of the inputs or antecedent sets [6] are the most popular approaches. Using a hierarchical structure of the model instead of a single-stage system also decreases the computational needs because the number of the rules increases exponentially with the number of the input parameters. In hierarchical systems, this increase is only linear because the problem is broken down into sub-problems. In this way, there are fewer input parameters at the different stages, leading to sub-systems that operate with fewer rules [7].

In this study, two significant modifications of the conventional Mamdani control system are introduced. One of them is the Mamdani-like structure with a discretized output, where the order of the evaluation steps is swapped, while the second is an arithmetic-based model, which is based on a linear combination of the membership function components. The aim of the study is to find an algorithm that can retain the advantages of the conventional control model, yet significantly reduce the computational requirements. The authors examined the proposed methods in a hierarchical patient surveillance system, where physiological parameters are monitored in real-time.

2 Mamdani-Type Control System

In the Mamdani system natural language rules are used, where the rules are given in the following form: *IF condition THEN consequence*. Let the input parameters be $x_1, x_2, \dots, x_n$ from the universe of discourse $X_1, X_2, \dots, X_n$, respectively. In this case, the output parameters $y \in Y$ can be calculated by using the following type of rules:

$$\text{IF } x_1 \text{ is } A_{1,i_1} \text{ and } \dots \text{ and } x_n \text{ is } A_{n,i_n} \text{ THEN } y \text{ is } B_{i_1, \dots, i_n},$$

where A_{k,i_k} is the antecedent i_k belongs to input k , $B_{i_1, \dots, i_n}$ is the fuzzy set that is assigned to the consequent set of the rules, $i_j = 1, \dots, n_j$; and n_j is the number of the antecedent sets belonging to input j . The rule premises can be obtained from all the possible combinations of the fuzzified values of the inputs.

2.1 Fuzzification

The system inputs are generally obtained as crisp values. In order to handle inherent uncertainty, a fuzzy value should be made from them by applying the fuzzification method. In this step the degree of fit of the input should be

determined on the antecedent set, i.e. the extent to which current inputs belong to the fuzzy sets, which are used to characterize them [8]. In the case of triangular membership functions and crisp inputs, the fuzzified value can be calculated using Eq. (1) and it is illustrated in Fig. 1.

$$\mu_{A_i}(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a \leq x \leq b \\ \frac{c-x}{c-b} & \text{if } b \leq x \leq c \\ 0 & \text{if } c \leq x \end{cases} \quad (1)$$

where a_i , b_i , c_i are the membership function parameters. Visually it looks like this:

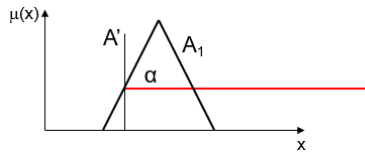


Figure 1

Fuzzification for triangular membership function

2.2 Firing Strength Calculation

During the evaluation the fuzzified values of all the input factors should be taken into account. For this reason, these values should be connected by a fuzzy operator for each rule, based on its antecedent to get the rule strength, which specifies the influence level of the given rule antecedent [9]. The operator selection depends on the task. In the case of a product operator, it can be calculated using Eq. (2) and its result is shown in Fig. 2. The product formula is

$$w_i = \prod_{j=1}^m \mu_{A_{ij}}(x) \quad (2)$$

where m is the number of the inputs, $\mu_{A_i}(x)$ is the fuzzified value of antecedent k of input j . And a visualization of it is

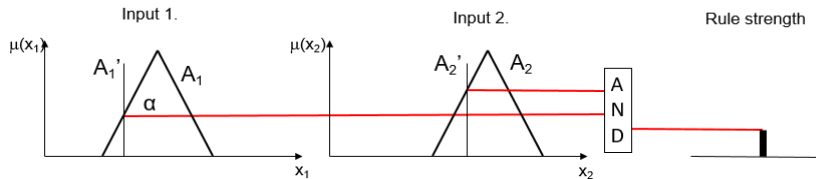


Figure 2

Firing strength calculation using product operator

2.3 Fuzzy Implication

After the rule strength has been determined for each rule we need to find the extent to which the rule-consequent is involved in the final result [10]. Different operators can be used depending on the task. In Fig. 3. the result of a product implication can be seen, which can be calculated using

$$y_{B_i} = w_i \mu_{B_i} \quad (3)$$

where w_i is the firing strength of rule i and μ_{B_i} is the consequent set belonging to rule i . Visually, it looks like this:

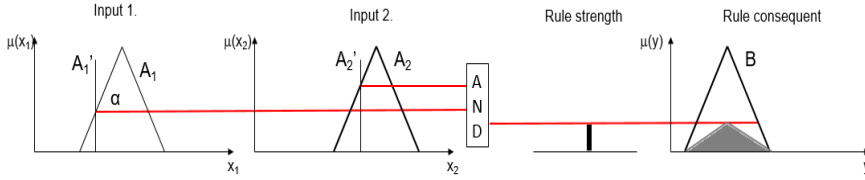


Figure 3

Fuzzy implication using product operator

2.4 Aggregation

After the implication has been calculated for each rule, the partial conclusions are obtained. These fuzzy sets should be aggregated via Eq. (4) to get the system result using an averaging operator [11]. Fig. 4 shows an example using the aggregation method with a sum operator. If the membership functions form a Ruspini-partition, i.e. $\sum_{i=1}^n \mu_i(x) = 1$, then it is not necessary to weight the sets obtained by using the firing strength [12].

Here,

$$y = \sum_{i=1}^n w_i y_{B_i} \quad (4)$$

where w_i is the firing strength of the rule-premise i , y_{B_i} is the result of the implication method for rule i , and n is the number of rules. Visually, it looks like

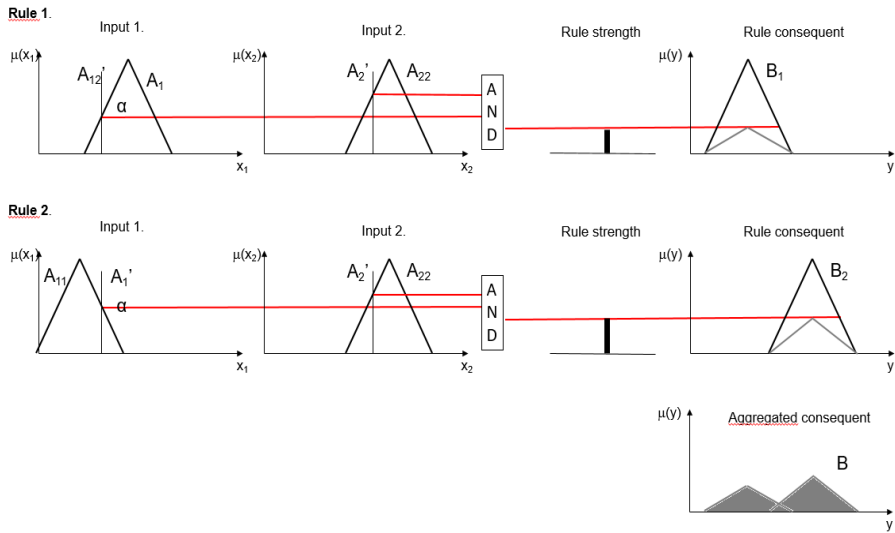


Figure 4
Sum aggregation

2.5 Defuzzification

In the case where a crisp output is needed, which characterizes the system result best, the complex shaped function obtained should be defuzzified. There are several different defuzzification methods available for, of which the centroid method is the most common and physically appealing of all the defuzzification methods [13]. Fig. 5 shows the results of the defuzzification process using the centre of gravity method, which can be defined by

$$y_{\text{out}} = \frac{\int_{y \in \mu_B} \mu_B(y) y dy}{\int_{y \in \mu_B} \mu_B(y) dy} \quad (5)$$

where $\mu_B(y)$ is the aggregated consequent set.

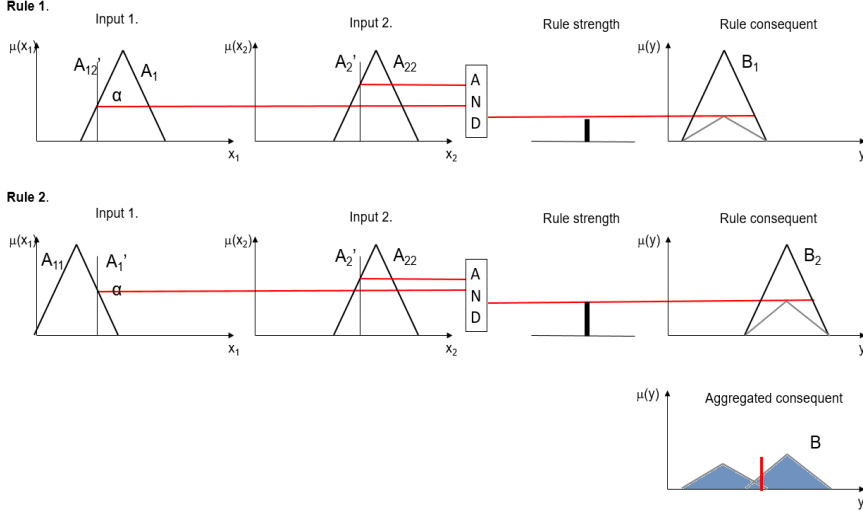


Figure 5

Centre of gravity (COG) defuzzification

3 Mamdani-like System with Discretized Output

3.1 Basic Principles of the Modifications

In the original Mamdani model aggregation and defuzzification are the evaluation steps, which increase the computational needs to the greatest extent. The idea of our reduction approach came from this fact. The first part of the evaluation (i.e. fuzzification, firing strength calculation, fuzzy implication) is the same as that in the original model, but the order of the aggregation and defuzzification is swapped. After the fuzzy implication step, each rule-consequent set is first defuzzified, then it is followed by aggregation. This modification might appear to further increase computational needs, because defuzzification is the computationally most intensive task and in this version, it should be performed as many times for as many rules there are. However, the high computational needs of the defuzzification in the original model is caused by the complex shape of the function, which is obtained after the aggregation. In contrast, in the modified model, simple shaped functions should be defuzzified, whose computational needs are negligible. And, during the aggregation significantly fewer operations are required because in this case crisp values should be aggregated instead of fuzzy sets. Detailed properties and a proof of the equivalency can be found in [17].

3.2 Defuzzification

In this modified version of the original Mamdani-type control, defuzzification is performed for each rule consequent set separately. These sets are piecewise linear (i.e. triangular, trapezoidal) fuzzy sets. As can be seen from Eq. (6) this simplified form significantly reduces the computational needs, for a triangular membership function.

Here,

$$y_i = \frac{a_i + b_i + c_i}{3} \quad (6)$$

where a_i , b_i , c_i , are the membership function parameters.

In the case of a symmetric function, the calculation is even simple, because in the case of triangular function the defuzzified value is just equal to b_i . In the case of symmetric trapezoidal function, it can be calculated using

$$y_i = \frac{b_i + c_i}{2} \quad (7)$$

The defuzzification method for the simplified model is shown in Fig. 6 below.

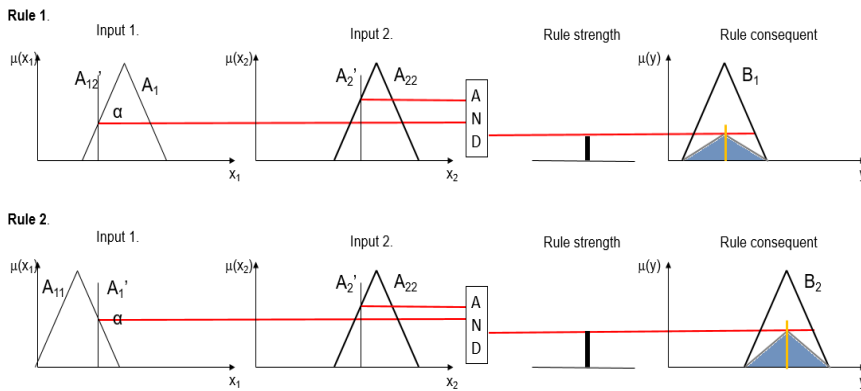


Figure 6

Defuzzification in the Mamdani-like system with discretized output

3.3 Aggregation

After defuzzified values have been obtained for all the rules, the final conclusion of the system should be drawn. For this reason, these crisp values should be aggregated as a weighted sum of them, where the weight factor is the firing strength of the rules, as follows

$$y_{out} = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \quad (8)$$

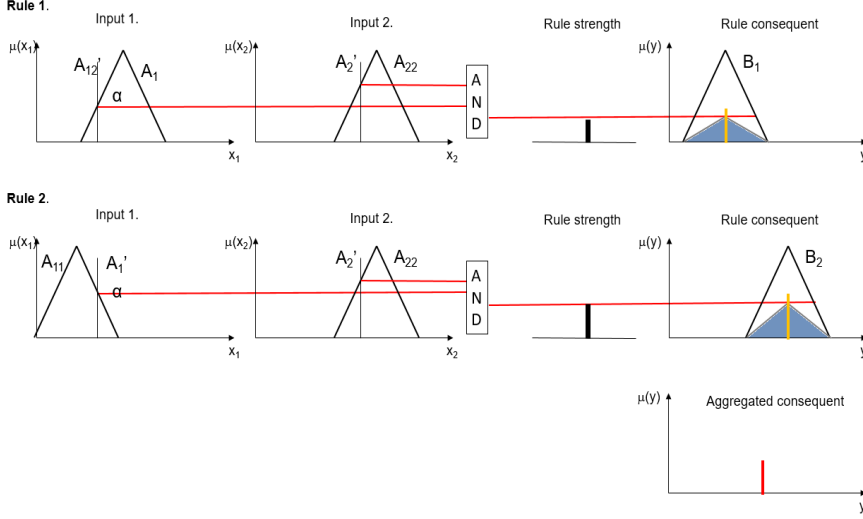


Figure 7

Aggregation in the Mamdani-like system with discretized output

4 Arithmetic-based Control

4.1 Basic Principles of the Method

The traditional Mamdani-type control is again used here as a basis for our simplification approach. Even the order of the evaluation steps is the same as in the traditional model (Subsection 2.1-2.5); only the step of fuzzy implication is omitted, while aggregation is modified. The basic idea is to manage the left and right-hand sides of the functions instead of aggregating the entire consequent sets. In this case, the result of the aggregation can be got as the linear combination of these parts of the functions, left and right-hand sides separately [14]. Consequently, the computational needs can be significantly reduced, due to using crisp parameters to represent the membership functions. The proposed method is equivalent to the α -cut, but it is more effective. Detailed properties and a proof can be found in [15].

4.1 Aggregation

After the firing strength calculation (2), the values obtained should be normalized like so:

$$w_{i_n} = \frac{w_i}{\sum_{i=1}^n w_i} \quad (9)$$

The left- and right-hand side of the aggregated consequent set can be represented by the following pair of equations

$$y_L = m_L(x - a_L) + \frac{1}{2} \quad (10)$$

$$y_R = m_R(x - a_R) + \frac{1}{2} \quad (11)$$

where m_L and m_R are the slopes of the membership function, $m_L > 0, m_R < 0$, $a \in X$, and $f(a) = \frac{1}{2}$.

A linear combination of the rule consequences can be obtained, for both sides of the functions, as follows:

$$a = \sum_{i=1}^n w_{i_n} a_i \quad (12)$$

$$\frac{1}{m} = \sum_{i=1}^n \frac{w_{i_n}}{m_i} \quad (13)$$

After the aggregation, the simple-shaped function obtained can be defuzzified using Eq. (6). The way of aggregation and defuzzification work is shown in Fig. 8 below.

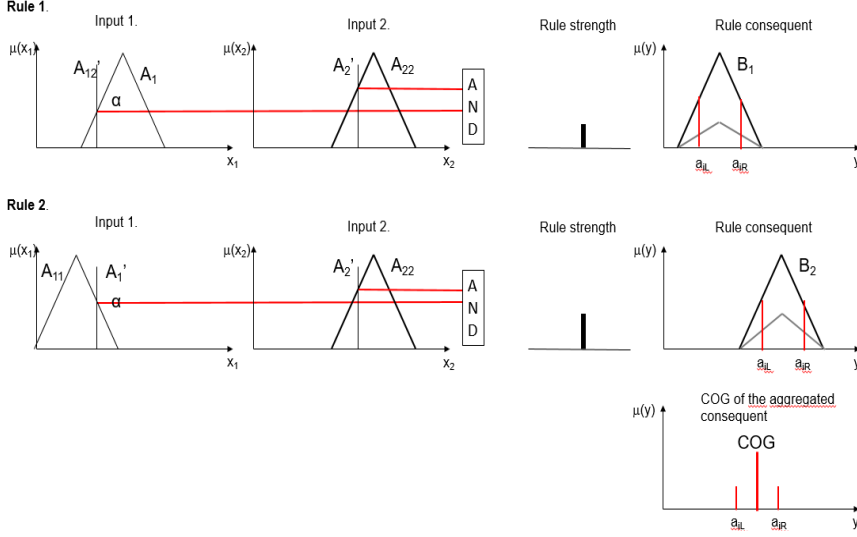


Figure 8

Aggregation and defuzzification in the arithmetic-based control process

5 Comparison of the Computational Needs

5.1 Mamdani-type Control vs. Mamdani-like System with a Discretized Output

When comparing the computational needs of these methods, it should be kept in mind that the first steps of the evaluation (i.e. fuzzification, firing strength calculation, implication) are the same. Consequently, their computational needs do not have to be taken into account because they are the same. This significant reduction can be achieved due to the swapping of the order of the aggregation and defuzzification.

In the traditional Mamdani-type model, fuzzy sets should be aggregated, then the complex shaped function obtained should be defuzzified. In practice, this means that aggregation must be performed with an equidistant division of the input range. Let Y be the input domain $[y_{min}, y_{max}]$, the set, which contains its equidistant base points $Y_i = [y_1; y_1 + \Delta; y_1 + 2\Delta; \dots; y_N]$, where N is the number of the base points of the equidistant division and the distance between two points is $\Delta = (y_{max} - y_{min}) / (N - 1)$. Aggregation should be performed for

these points using Eq. (4), thus the number of operations requirements of this step is $N * n$ multiplicative and $n - 1$ additive operations. During the defuzzification, the trapezoidal rule is applied like so:

$$\int_{y_{\min}}^{y_{\max}} f(y) dy \approx T_n(f) = \frac{y_{\max} - y_{\min}}{2N} \left[f(y_1) + 2 \sum_{i=2}^N f(y_i) + f(y_{N+1}) \right] \quad (24)$$

where $y_i = y_1 + (i - 1) \frac{y_{\max} - y_{\min}}{N}$, $i = 1, \dots, n+1$. The number of operations needed for the COG method, which is defined in (5) is $N+1$ additive and 3 multiplicative operations (since aggregated values are given in the base points). This is equal to the computational needs of the denominator of the formula, while in the nominator $N+1$ additive and $N+3$ multiplicative operations are needed. Hence, the overall formula requires $2(N+1)$ additive and $N+6$ multiplicative operations.

In the Mamdani-like structure with discretized output, the centre of gravity of each simple-shaped rule consequent is calculated. Using Eq. (6) the number of operations needed is 2 additive, and 1 multiplicative operation for each rule consequent. Then, these crisp values should be aggregated as their weighted sum, where the weight factor is the firing strength of the rules defined by Eq. (8). Its computational needs are $2(n - 1)$ additive, and n multiplicative operations.

A summary of the above is shown in Table 1.

Table 1
Comparison of the computational needs for the traditional and discretized Mamdani-type control processes

Operation		Traditional Mamdani	Discretized Mamdani
Aggregation	Additive	$n-1$	$2n$
	Multiplicative	$N*n$	n
Defuzzification	Additive	$2(N+1)$	$2(n-1)$
	Multiplicative	$N+6$	n
Sum	Additive	$n-1+2(N+1)$	$4n-2$
	Multiplicative	$N*(n+1)+6$	$2n$

Although the Takagi-Sugeno system was not addressed, a brief comparison was made for the sake of completeness regarding the computational requirements. The results are shown in Table 2. Takagi-Sugeno is a standard method, where the rule consequents can be generated as a function of input values using $g_{i_1, \dots, i_n}(x_1, \dots, x_n)$, and the rules can be represented in the following form [18]:

$$x_1 \text{ is } A_{i_1} \text{ and } \dots \text{ and } x_n \text{ is } A_{i_n} \text{ THEN } y \text{ is } g_{i_1, \dots, i_n}(x_1, \dots, x_n).$$

As a result of the above the computational requirements of the function should be examined, instead of the defuzzification step, which is omitted.

Table 2

Comparison of the computational needs for the Takagi-Sugeno and discretized Mamdani-type control processes

Operation	Takagi Sugeno	Operation type	Discretized Mamdani	Operation
Aggregation	$2(n-1)$	Additive	$2n$	Aggregation
	n	Multiplicative	n	
Output function	$2n$	Additive	$2(n-1)$	Defuzzification
	$3n$	Multiplicative	n	
Sum	$4n-2$	Additive	$4n-2$	Sum
	$4n$	Multiplicative	$2n$	

The results in the table clearly show that the number of operations needed for the traditional Mamdani control system is significantly higher than in the case of its discretized version, for both additive and multiplicative operations.

5.2 Mamdani-like Structure with Discretized Output vs. Arithmetic-based Control

Comparing the two methods, it should be mentioned that in the arithmetic-based algorithm fuzzy implication is not a separate step, and the firing strength is used during the aggregation process to weight the parameters of the functions. For this reason, the number of operations needed for fuzzy implication cannot be defined for this approach. Another important issue is defuzzification. In both cases, simple-shaped functions should be defuzzified. As a consequence, the number of operations needed for defuzzification do not have to be taken into account because their computational needs are the same.

In the case a Mamdani-like system with a discretized output, during the implication the above-defined equidistant division is used. The function values should be multiplied by the firing strength of the given rule (3), hence it requires $N \cdot n$ multiplicative operations. In the arithmetic-based method, we also need to aggregate crisp values as in the discretized Mamdani model. However, here

$$a \in X, f(a) = \frac{1}{2} \quad (12), \text{ and the slope of the functions (13) should be}$$

aggregated. Based on these equations, the computational needs of the aggregation are $4(n-1)$ additive, and $4n$ multiplicative operations.

After summarizing the operational needs of the methods, it can be seen that the arithmetic-based method provides a slightly worse result for additive operations,

but discretized Mamdani structure has significantly higher computational needs for multiplicative operations. It may be deduced from the above that the arithmetic-based method has fewer computational needs overall.

Table 3

Comparison of the computational needs of the discretized Mamdani-type and arithmetic-based control

Operation		Discretized Mamdani	Arithmetic-based
Implication	Additive	-	-
	Multiplicative	$N \cdot n$	-
Aggregation	Additive	$2n$	$4(n-1)$
	Multiplicative	n	$4n$
Sum	Additive	$2n$	$4(n-1)$
	Multiplicative	$(N+1)n$	$4n$

6 A Case Study

The above-described methods were examined in a patient surveillance system, where physiological parameters were monitored in real-time. In these kinds of systems, a short reaction time is essential to avoid any serious consequences caused by the inappropriately chosen form, or intensity of the motion [16]. In real-time systems use of conventional Mamdani control is limited, due to its high computational requirements. However, with a modified algorithm one can retain its advantages, and also ensure a faster reaction time.

6.1 Model Structure

In the system, input parameters are selected patient-by-patient, based on a personal profile. This means that the number of the monitored parameters and their ranges can be defined specifically for patients and sports. A schematic structure of the model is shown in Fig. 9. The aim of the system is to assess the risk level of the current activity, based on the measured physiological values in the “Medical condition” group; the characteristics of the sport activity are its duration, frequency, and intensity under the “Activity” group; and in the case of an outdoor activity, the environmental conditions are listed in the third main group. An evaluation is performed hierarchically, i.e. risk levels of the different groups are calculated separately for each group, then these values form the input parameters of the summarizing group at the next level, where the overall risk is obtained. Due to the hierarchical structure, the system can be easily modified or expanded, where necessary.

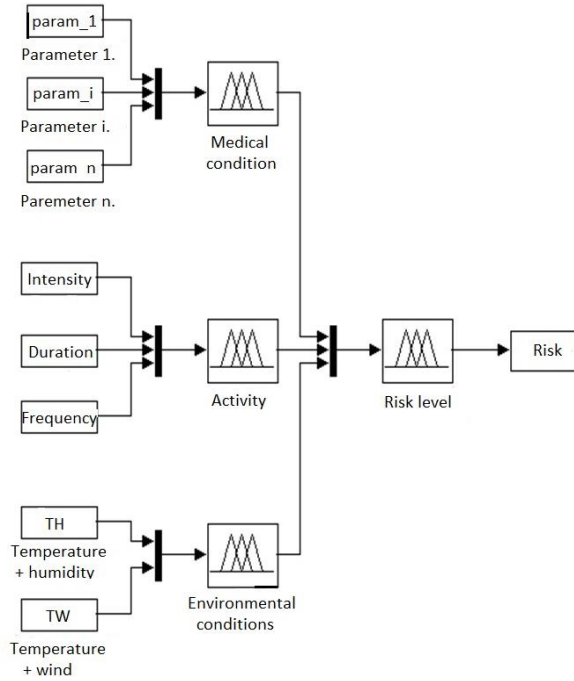


Figure 9
Model structure

6.2 A Comparison of the Control Models

During the study the heart rate, systolic and diastolic blood pressure of the patients were examined for 576 different input value combinations, for a 50-year old female patient.

First, the results of the different algorithms were compared to demonstrate the interchangeability of each. The compatibility of the conventional Mamdani control and the Mamdani-like control with a discretized output has already been shown mathematically [17]. For the two discretized solutions, the results of Mamdani-like structure and arithmetic-base model are shown in Fig. 10. Quantifying the comparison of the result: the average difference, the maximum difference, the standard deviation, and the correlation-coefficient were also calculated (see in Table 4). The figure and the metrics tell us that the output values in both systems are almost the same, the difference between them being negligible, supporting the former mathematical proof.

Here, the computational speed of the different methods is also considered in a standard desktop environment. The algorithms were implemented in a RAD

Studio multi-platform, multi-device environment. The running time for the 576 different cases are as follows: 8.017 s for the conventional Mamdani control, and 4.309 s for Mamdani-like control, 4.283 s for the arithmetic-based model. This means that the average evaluation time of the current input for the different algorithms is 14 ms, 7.5 ms, and 7.4 ms, respectively. From the above, we may infer that the computational speed is approximately the same as in the case of the Mamdani-like control and arithmetic-based models, while the conventional Mamdani control requires almost twice as much CPU time. As can also be seen from the mathematically described computational requirement (Table 1, 3).

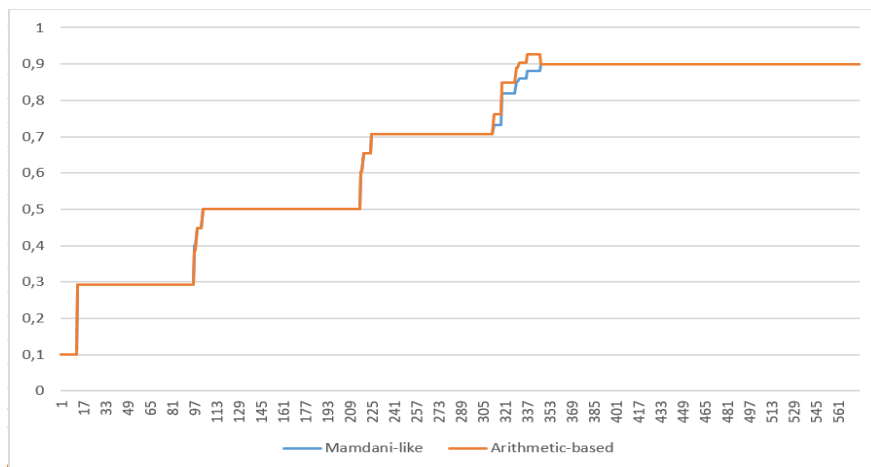


Figure 10

Comparison of the Mamdani-like and arithmetic-based control results

Table 4

Comparison of the results of the discretized Mamdani-type and arithmetic-based control for 576 different cases

Average difference	0.0022
Maximum difference	0.0455
Standard deviation	0.0091
Correlation-coefficient	0.9993

Conclusions

Fuzzy control systems are widely used in medical-related applications because of their advantageous properties. That is, linguistic terms can be incorporated into the model, it can handle the subjectivity in the data and in the evaluation process, and its operation is very close to human thinking. However, the applicability of the conventional Mamdani control is limited by its high computational requirements in real-time and adaptive systems. In this study, possible modifications of the conventional Mamdani model were suggested, such as the

Mamdani-like structure with a discretized output and the arithmetic-based model. The main contribution of the paper is that the proposed models maintain the basic “philosophy” of the Mamdani-type controller while utilizing the properties of the triangular and trapezoidal membership functions technical modifications were suggested that drastically reduce the computational needs in comparison with that of the original method that is formulated for general form membership functions. Computational requirements of the different approaches were compared mathematically in general, for any-shape function. Besides that computational speed was also compared in a patient monitoring system. Based on the analysis both the Mamdani-like system and arithmetic-based model have lower computational needs than the conventional Mamdani model. There is no significant difference between the two discretized models, and both can enhance the features of the conventional model by decreasing the computational requirements and CPU time required.

References

- [1] T. Obeidi, C. Larbes, A. Ilinca, G. F.T. Kebir, Fuzzy Logic-based Maximum Power Point Tracking for a Solar Electric Vehicle, *Acta Polytechnica Hungarica*, Vol. 15, No. 7, pp. 133-156, 2018
- [2] A. Gegov, “Complexity management in fuzzy systems,” *Chapter Rule Base Reduction Methods of Fuzzy Systems of Series in Studies in Fuzzyness and Soft Computing*, Springer, Heidelberg, pp. 17-31, 2007
- [3] R-E. Precup, “On the combination of tensor product and fuzzy models,” In *Proc. of the IEEE Int. Conference on Automation, Quality and Testing, Robotics*, Cluj-Napoca, 22-25 May, pp. 48-53, 2008, DOI: 10.1109/AQTR.2008.4588792
- [4] F. Herrera, “Genetic fuzzy systems: taxonomy, current research trends and prospects,” *Evolutionary Intelligence*, Vol. 1, No. 1, pp. 27-46, Springer-Verlag, 2008
- [5] S. Chopra, R. Mitra, V. Kumar, “Reduction of fuzzy rules and membership functions and its application to fuzzy PI and PD type controllers,” *Int. Journal of Control, Automation, and Systems*, Vol. 4, No. 4, pp. 438-447, 2006
- [6] S. Ding, H. Zhao, et. al, “Extreme Learning Machine: Algorithm, Theory and Applications”, *Artificial Intelligence Review*, Vol. 44, No. 1, pp. 103-115, June 2015
- [7] Y. Chen, A. Abraham, “Hierarchical fuzzy systems,” *Tree-Structure based Hybrid Computational Intelligence*, Springer, Heidelberg, Vol. 2, 2010, pp. 129-147
- [8] T. J. Ross, *Fuzzy Logic with Engineering Applications*, Second Edition, John Wiley & Sons, 2004

- [9] R. Czabanski, M. Jezewski, J. Leski, Introduction to Fuzzy Systems, Theory and Applications of Ordered Fuzzy Numbers, Studies in Fuzziness and Soft Computing, Vol. 356, pp. 23-42, 2017
- [10] Kyung Whan Oh, Wyllis Bandler, Properties of Fuzzy Implication Operators, International Journal of Approximate Reasoning, Vol. 1, No. 3, pp. 273-285, 1987
- [11] R. Fullér, Aggregation Operators – Tutorial, Institute for Advanced Management Systems Research Department of Information Technologies Abo Akademi University, 2010
- [12] A. R. Várkonyi-Kóczy, “Model Based Anytime Soft Computing Approaches in Engineering Applications.” In Balas, V., J. Fodor, A. R. Várkonyi-Kóczy (eds.), Soft Computing Based Modeling in Intelligent Systems (Ser. Studies in Computational Intelligence), Springer Verlag, Berlin, Heidelberg, 2009, pp. 63-92, DOI: 10.1007/978-3-642-00448-3_4
- [13] Oļegs Užga-Rebrovs, Gaļina Kuļšova, Comparative Analysis of Fuzzy Set Defuzzification Methods in the Context of Ecological Risk Assessment, Information Technology and Management Science, Vol. 20, No. 1, pp. 25-29, 2018
- [14] J. Dombi, T. Szépe, Arithmetic-based Fuzzy Control, Iranian Journal of Fuzzy Systems, Vol. 14, No. 4, pp. 51-66, 2017
- [15] J. Dombi, Pliant arithmetics and pliant arithmetic operations, Acta Polytechnica Hungarica, Vol. 6, No. 5, 2009, 19-49
- [16] Ábel Garai, István Péntek, Attila Adamkó, Revolutionizing Healthcare with IoT and Cognitive, Cloud-based Telemedicine, Acta Polytechnica Hungarica, Vol. 16, No. 2, pp. 163-181, 2019
- [17] E. Tóth-Laufer, I. J. Rudas, M. Takács, Operator Dependent Variations of the Mamdani-type Inference System Model to Reduce the Computational Needs in Real-Time Evaluation, International Journal of Fuzzy Systems, Vol. 16, No. 1, pp. 57-72, 2014
- [18] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on System, Man and Cybernetics, Vol. 15, pp. 116-132, 1985