INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 17, Issue: 1, Month: 2, Year: 2022
Article Number: 4683, https://doi.org/10.15837/ijccc.2022.1.4683

CCC Publications

Improve the design and testing of fuzzy systems with a set of (almost) simple rules

H.N. Teodorescu

Dedicated to the memory of Prof. Ioan Dzitac

Horia-Nicolai Teodorescu*
Technical University of Iasi
and
Institute of Computer Science of the Romanian Academy
Iasi, Romania,
hteodor@etti.tuiasi.ro

Abstract

Prof. Dzitac used to say, 'The mathematics of fuzzy systems is not fuzzy'. We discuss several limits and potential errors in the design of fuzzy logic systems and how they can be corrected or avoided. Examples from the literature are presented.

Keywords: fuzzy logic, fuzzy logic system, design, interpolation, approximation, input-output function.

1 Introduction

As the much regretted Prof. Dzitac used to say, 'The mathematics of fuzzy systems is not fuzzy', [1]. Yet, many articles on fuzzy logic systems (FLS) and their applications are rather 'fuzzy'. This statement was meant to clarify the meaning of the concept of 'approximate reasoning' used by Zadeh, see for example [2]. Unfortunately, there are papers that include elementary errors or prove a lack of understanding of the principles of FLSs. The design of fuzzy systems could benefit by applying a set of concepts and simple rules rooted in the mathematics of fuzzy systems. We discuss some design limits and provide examples remediating them. The Annex contains brief formal explanations of some of the theoretical foundations of the recommended rules. We discuss only FLSs under min-max logic; the case of FLSs with max-prod is somewhat simpler and more extensively dealt with in the literature.

Throughout the article, we use the following typical abbreviations:
FLS fuzzy logic system
mf(s) membership function(s)
c.o.g. center of gravity (defuzzification method)
PID proportional integral derivative (control)
I/O input-output (function)
CF characteristic function
SISO single input single output (FLS).

2 Some basic design issues

We recall that FLS with c.o.g. defuzzification are nonlinear systems. In addition, they are systems represented by piecewise functions. Hence, they have no transfer function, as the linear systems have. When the input and output membership functions are piecewise linear or low degree polynomials, the input-output functions of Sugeno or Mamdani FLSs are piecewise ratios of integrals of polynomial functions (see P1 in the Annex). When transcendental m.f.s are used, the I/O functions of the FLS become transcendental. These make the treatment of FLSs quite laborious and prone to errors and may impose an exhaustive analysis of the operation; for example, they may need a testing pertaining to Monte Carlo methods.

2.1 Take care of discontinuous derivatives

Typical SISO FLSs with c.o.g. defuzzification and with triangular or trapezoidal input m.f.s have the characteristic function with discontinuous derivatives in a set of points defined by the vertices (peaks) of the input m.f.s. When the derivative of the characteristic function is discontinuous, improper operation may happen in control loops, such as resulting infinite accelerations, which are unrealistic, or jumps in currents, which are impossible. Consider the inverse pendulum control. One cannot create discontinuous forces, because that would mean jumps in currents with electric motors, which have inductances, so electric inertia. Few papers address this issue, be it by showing that in a specific implementation the discontinuity of the derivative of the force does not count because of factors such as the electrical inertia of the actuators or of the driving motor.

2.2 Approximation and interpolation with FLSs – a general perspective

Most applications of FLS are either modeling or control problems. These lead to either approximation or interpolation problems, that is, finding some function $g : R^n \rightarrow R$ that approximates or interpolates a possibly unknown function $f : R^n \rightarrow R$. It turns out that FLSs are well suited to easily interpolate functions, and quite well suited to approximate functions that are intuitively well understood. Indeed, according to a basic theorem, any continuous function in a bounded domain can be (exactly) interpolated through a given finite set of points by a Sugeno or Mamdani type FLS with input and output triangular m.f.s [3] [4]. According to a set of theorems named generically 'universal approximation theorems', FLSs are also able to approximate with whatever small error a continuous function over some compact domain. Even classification problems are amenable to an approximation or interpolation problem, although classification asks for an extra operation of transforming the continuous output into a discrete one, corresponding to the numerical code of the class. These results show that it is crucial to understand the approximation and interpolation by FLSs and to apply this knowledge for obtaining good results.

When interpolation is involved, one may simply choose the input m.f.s to be triangular and overlapping two-by-two, as in many elementary examples (including in MATLAB), taking care that the vertices of the triangles are placed at the abscissae of the interpolating points; then, if the system is Sugeno-type, choose the singletons to correspond to the desired values of the interpolated input-output function. That is, if the function to interpolate is $f : R^n \rightarrow R$ and the interpolation points are $\{(x_{1k}, x_{2k}, ..., x_{nk}), y_k\}$, $k = 1, ..., m$, the triangular m.f.s for the variables $x_1, ..., x_n$ should have the vertices at precisely $(x_{1k}, x_{2k}, ..., x_{nk})$, $k = 1, ..., m$ and the singletons should be at y_k. This guarantees that the FLS is a perfect interpolator [3]. For Mamdani FLSs, the difference is that the triangular m.f.s must have the c.o.g.s at y_k. The case of functions $f : R^n \rightarrow R^q$ is similarly treated.

Next comes the issue of approximation. Some of the theorems of universal approximation [3] provide a clarification, essentially saying that the approximation error decreases when the numbers of m.f.s at the input and output increase. This leads to a simple design rule:

Especially when the m.f.s are isosceles triangles, do not use too few rules, because then the characteristic function of the FLS will be too sketchy.

Consider a FLS with triangular m.f.s, where the mfs overlap two by two, moreover the abscissa x_h where a membership function has value 1 is the abscissa where the other m.f.s that overlap with the first have (reach) value 0 (Fig. 1). Then, on any interval of overlapping, the overlapping m.f.s are complementary, $\mu_n(x) = 1 - \mu_{n+1}(x)$. This condition is essential in the discussion. Also, assume that any rule assigns exactly a single output membership function. Then,

Remark. As a result of these conditions, the FLS input-output function of the defuzzified FLS satisfies the conditions: $f(x = (x_{1h}, ..., x_{nh})) = y_h$ where y_h is the corresponding defuzzified value; in addition, the value y_h corresponds to the singleton in the respective rule, for Sugeno systems, respectively the c.o.g. of the single output m.f. selected by the respective rule, for Mamdani f.l.s. with c.o.g. defuzzification. On the other hand, too many rules will make the computation difficult. The computation load is proportional with the product of the numbers of m.f.s of the inputs; having many inputs increases the computation load.

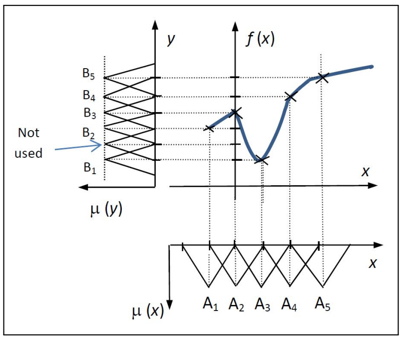

Figure 1: Interpolator with issosceles m.f.s, not all output m.f.s used.

Notice in Fig. 1 that, when forcing the m.f.s to be equal isosceles triangles at the output, some of them may be unused, or they may supperpose imperfectly (irregular superposition), while the input m.f.s may have different bases (A5 in Fig. 1; see also a version of Fig. 1 in [3]). The remarkable easiness of designing a FLS that is an interpolator for some desired control surface does not mean that FLS is also a good approximator for the desired surface. In this respect, always choosing isosceles triangular m.f.s is not good, because in case of far from linear surfaces are poorly approximated by interpolators with such m.f.s. An example is shown in Fig. 2; the Sugeno interpolator with isosceles m.f.s has an approximation error almost twice as large as the shown interpolator with irregular m.f.s, 0.3286 versus 0.1757.

Care is needed in treating the regions close to the boundaries of the universe of discourse. When the boundary m.f.s are constant on an interval or when on a boundary interval of the definition domain of the fls there is a single m.f. active (non-zero) on each input variable, then the output is a constant on that interval. When there are n variables, each with m_k m.f.s, and there is a single m.f. active (non-zero) on each input variable on the boundary intervals, there are 2^n boundary intervals where the output is constant. On those intervals it is not efficient to compute the output, as it is already known. Figure 3 illustrates this effect, which is almost ubiquitous in the literature, unfortunately (the figure is from a student project where all corners had this issue, but similar cases are found in the literature, for example [5], Fig. 6, and [6] Fig. 3). A simple cure is to add complementary m.f.s at the boundaries, even if those m.f.s seem useless – actually they are useful to avoid constant values of the output.

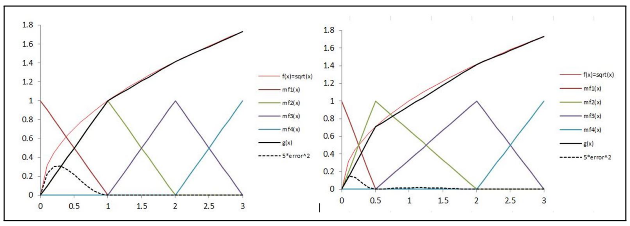

Figure 2: Interpolator with isosceles m.f.s and irregular m.f.s. Approximation errors differ: the error is twice lower in the second case.

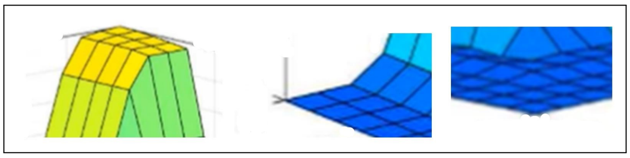

Figure 3: Corner effect: the control surface is flat, meaning that the control is reduced to a constant bias.

2.3 Modeling versus control

Confusion between closed loop control and modeling problems (open loop control) may often occur. Consider a device that is able to eject objects, with specified velocity, at the level of the terrain or of a table, with the aim that the object (ball, puck etc.) stops at a specified point. Such a device could imitate humans playing golf or billiard. Apparently, it is a control problem to send the ball over a terrain with a velocity such that the ball stops at a given point. But there is no control exerted on the ball after throwing it. For the ball stops at the given point, it needs to lose its kinetic energy through friction exactly at the specified point. The ball has an initial energy $mv^2/2$ and the lost energy along a traveled distance d is μmgd where μ is the friction coefficient, m the mass of the ball and g the gravitational constant. Therefore, the ball stops after a distance $d = mv^2/2\mu mg$, while the velocity needed to attain d is $v = \sqrt{2\mu gd/m}$. The "control" of the ball speed turns out to be a simple modeling problem. The uncertainty my come from the approximate estimation of the distance d and from the unknown value of μ. Under uncertainty, it may be justified to estimate the initial (ejecting) velocity v in a fuzzy manner, for example with rules such as

If the distance is large, the ejecting velocity should be large;

or, taking into account the estimation of the roughness (friction coefficient) of the terrain,

If the distance is large AND the roughness is high, the ejecting velocity should be very large.

Different sets of rules depending on the estimated distance should be generated for each estimated value of the roughness; that is, one model velocity vs. distance is generated for every estimation of μ. Essentially, the model requires a fuzzy interpolator for the law v vs. d, for each model in the family of models. Because the law $v = \sqrt{2\mu gd/m}$ is nonlinear, a choice of triangular isosceles membership functions for d is not suitable; instead, non-isosceles triangles should be used. The exmple in Fig. 2 is inspired by this false (closed loop) control problem.

2.4 Are fuzzy models and fuzzy control systems guaranteed to be robust?

The inherent robustness property of fuzzy logic systems is frequently claimed in the literature, e.g., [7]; also see the discussion and references in the early paper [6]. However this statement is not always veryfied and sometimes is is not correctly verified. Moreover, in many cases, the meaning of 'robustness' is not defined, although there are several different definitions in the literature. An essential issue is that FLSs are nonlinear systems, with piecewise defined characteristic functions; therefore they respond differently at different operating points. Even more, FLSs with triangular or trapezoidal functions have discontinuous derivatives of their input-output functions. That makes their response strongly dependent on the amplitude of the input signals at operating points on or close to the boundaries of the domains where their input-output functions are derivables. Consequently, applying the standard unitary step function at the input, as for linear systems, is not enough informative. Linear systems will respond identically for steps of different hights, which is no more true for nonlinear systems. There are numerous papers that use solely unitary steps and a single operating point for testing, without mentioning the above issues. Next, it is necessarily to clarify to what kind of perturbation the system is robust; robustness may relate to changes (or uncertainties) in the parameters of the controlled system, or of the FLS used in control or in modeling. The input-output function of the FLSs can be very sensitive to the change of a membership function, as shown in Fig. 2.

In [5], the authors correctly apply steps with several heights to check the robust operation of their control system; the figures they provide (Figs. 9 and 11) show clearly different responses for different operation points, althogh the description details do not clarify enough the operation (for example, the membership functions are defined on [-1,1] in Fig. 5, but they use steps of values 2, 4 and 8, incompatible with the [-1,1] range). Serra and Quelas [8] also intuitively apply a correct testing method, by generating input steps with random amplitudes; Figures 7 and 9 in their paper clearly show that the system has different responses for different heights of the steps. Concluding, one should apply caution to robustness statements when they are not correctly verified, or when the meaning of 'robustness' is not defined. Providing an explanation on the sampling frequency on the loop including the FLS is also essential when checking the stability and the robustness. The lack of data on the operation point of the FLS where a perturbation is applied, or of the amplitude of perturbation can desqualify any claim of robustness of the FLS.

In case of FLSs, after carefully choosing a meaning for robustness, one should check it by generating random inputs with various amplitudes and for varied operating points of the system. When testing with sinusoidal or harmonic systems one has to consider that the nonlinearity and the discontinuity of the derivative of the CF of the FLS produces harmonics and that the amplitudes of the harmonics depend on the amplitude of the testing signal and on the operation point. Beyond the above discussion, there is at least one sense of understanding the robustness of FLSs, namely that given in the FLSs' property:

Property of low sensitivity of FLSs to the change in input m.f.s. Consider a SISO FLS with input m.f.s reaching value 1 in a single point and overlapping at most two by two for values larger than 0, with the vertex of one m.f. being a point where adjacent m.f.s have value zero, and with *symmetric* output m.f.s overlapping at most two by two in the sense above and reaching value 1 in a single point. All m.f.s are considered to have values larger than zero on compact intervals. Then, *on any interval defined by the vertices of two successive input m.f.s, the characteristic function of the FLS remains in the same output interval whatever is the choice of the input m.f.s as long as their vertices do not change.*

The property conditions are illustrated in Fig. 4 (where the two branches of the m.f.s are assumed to be polynomials). The shape of the m.f.s in Fig. 4 does not change the respective interval of the output. The property guarantees us that errors in the choice of shapes of input m.f.s affect in a limited manner the output. The property is easily extended to multiple input systems respecting the same conditions for the m.f.s.

The above leads to the following criterion of robustness for FLSs:

Robustness with respect to the shape of m.f.s. A FLS with the m.f.s satisfying the given conditions is robust to the change of the shape of the membership functions in the sense that the output remains confined to the same intervals, defined by the vertices of the m.f.s, as long as the input(s) remain in the intervals defined by the vertices of the input m.f.s and the intersection points of the adjacent m.f.s.

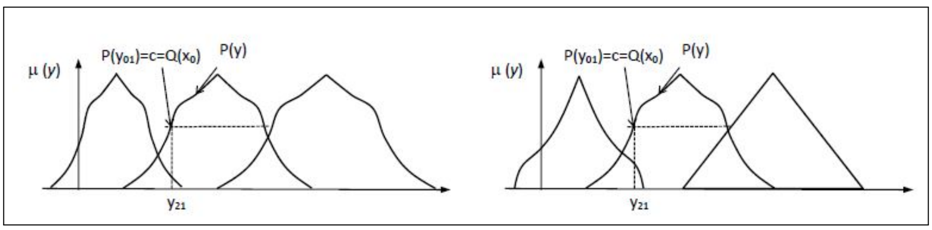

Figure 4: Case of symmetric m.f.s overlapping at most two by two. The defuzzified output is between y_{cog1} and y_{cog2}.

Some authors claim robustness but seem to imply the robustness of the stability of the loop. This kind of statement should be carefully pondered. Assume a control loop with fuzzy control of at least one parameter of the loop. Denote by u that parameter, with the main system in the loop $y = f(x), x = g(y; u, \dot{u}, \ddot{u})$, g depending on $\ddot{u}$ as $\ddot{u}^k, k > 0$, and $u = h(y)$ expressed by a SISO FLS. Assume that the FLS is a Sugeno system, or a Mamdani system with triangular input m.f.s, unequal or not isosceles triangles. Then, the FLS with c.o.g. has the CF non-derivable at certain points. Consequently, the system is unstable, because $\ddot{u}$ is infinite for some values of y, namely at the limits of the intervals defining the piecewise function. Fortunately, real systems have mechanical or electrical inertia that limit the values of any involved $\ddot{u}$ to finite values. It is recommendable to use low-pass filters in loops involving FLSs to further limit the frequency spectrum of the signals in the loop during operation and thus increase the prospect of stability.

2.5 Building analytic expressions for FLSs may be impossible

A tradeoff in designing FLSs is frequently found in the literature, namely the time spent with deriving analytic expressions for the operation of systems involving FLSs and the time required for simulations. Many designers present the principles of the FLS, but do not pursue the derivation of analytic expressions, because the CFs of the FLSs are piecewise and should be determined on each multidimensional interval where the CF has a single expression. This is justified especially when the number of multidimensional intervals (and correspondingly the number of rules) is very large. For example, if there are three input variables and seven triangular membership functions (verlapping at most two by two) for each variable, there are 8^3 3-dimensional elementary intervals where the CF has to be expressed.

However, the simple case described above is not depicting the more complex situation situation when the CF has no analytic expression. The issue is discussed below. We will assume that all m.f.s attain value 1 in a single point; at left and right of that point the m.f.s are represented by polynomials and are intersecting at most two by two. Consider a Mamdany FLS with a single input variable, with the input m.fs represented at left of the vertex by the polynomial $Q(x)$. Let the input value be x_0 and $c = Q(x_0)$. According to the respective rule, the truncation of the corresponding output m.f. described at left by $P(x)$ leads to the equation $P(y) = c$ and results in a single solution, y_{01}; a second solution is produced by the right section of the m.f, y_{02}. There are no more solutions as we assume that all α-cuts are compact. The solutions are algebraic functions of c if P has degree 4 or lower; for higher degrees, Abel-Ruffini theorem let us know that the solutions are not algebraic in general (not algebraic in almost all cases). Therefore, for m.f.s represented by higher degree polynomials, we may be unable (except simple cases) to find an expression for y_{01}, y_{02}. But these expressions are needed to express the c.o.g.:

$$y_{cog} = \frac{... + \int_a^{y_{01}(Q(x_0))} yP(y)dy + ...}{... + \int_a^{y_{01}} P(y)dy + ...}$$

where a is the left limit of the m.f. Above, we have shown only one of the defined integrals in the expression of the defuzzified output. Other integrals have as limits similar solutions of polynomials of the output m.f.s, $P_j(y) = Q_h(x_0)$, where P_j are polynomials of output m.f.s and Q_h are polynomials of input m.f.s; or the integrals have limits solutions of $P_i(y) = P_j(y)$ of m.f.s that intersect. When all these polynomials of the output m.f.s gave the degrees less than 5, the solutions can be represented by algebraic expressions (i.e., using only powers, radicals and + and ×. Thus, the denominator and nominator are polynomials of algebraic expressions of x_0; therefore, in the stated case we are guaranteed that we can find an expression for the CF of the FLS. Notice that the input m.f.s are unessential (they can have a higher degree than 4), because they contribute only the constant c. On the other hand, when the polynomials of the output m.f.s have higher degree, in most cases the solutions cannot be found as algebraic expressions (and we do not have today a method to find the solutions).

The above discussion also leads to the nice property of FLS given in the Annex, P2.

3 Conclusions and future work

Despite their intuitive grasping and their ease of description in a basic manner, FLSs are complex systems that are not easy to master and adapt, and their properties and operation are frequently difficult to predict, because of their nonlinearity. As a consequence, many designs of control and modeling based on FLSs are suboptimal and sometimes erroneous. Applying a few simple rules may help avoiding significant errors. Many of the presented rules remain valid in more elaborate settings, including under different versions of fuzzy logics, see for example results in [9], or in neurofuzzy systems as in [10].

Acknowledgments. I am indebted to the graduate students in my class of Fundamentals of A.I. for teching me several potential design errors; their errors in the projects have been often most illuminating and sometimes they were motivational for further research.

References

[1] I Dzitac, I; Filip, FG, MJ Manolescu (2017), Fuzzy logic is not fuzzy: World-renowned computer scientist Lotfi A. Zadeh. *International Journal of Computers, Communications & Control* . v. 12, n. 6, p. 748-789, Dec. 2017. ISSN 1841-9844.

[2] Teodorescu, H.-N.L. (2006). On the meaning of approximate reasoning. An unassuming subsidiary to Lotfi Zadeh's paper dedicated to the memory of Grigore Moisil. *International Journal of Computers Communications & Control* 6 (3), 577-580.

[3] Teodorescu, H. N. (2007) *Sisteme Fuzzy si Aplicatii*. Ed. Politehnium, Iasi, 2007.

[4] HNL Teodorescu, (2013) On the characteristic functions of fuzzy systems. *International Journal of Computers Communications & Control* 8 (3), 469-476.

[5] K. Sharma and D. K. Palwalia, (2017) Robust controller design for DC-DC converters using fuzzy logic, 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), 2017, pp. 477-481, doi: 10.1109/ISPCC.2017.8269726.

[6] J. Wang and J. R. Jordan, "The robust step performance of PID and fuzzy logic controlled SISO systems," Proceedings of 1995 IEEE International Conference on Fuzzy Systems., 1995, pp. 325-330 vol.1, doi: 10.1109/FUZZY.1995.409699.

[7] S. Lihan, M. Jie and Y. Baoqing, (2020) "Fuzzy PID Design of Vehicle Attitude Control Systems," 2020 Chinese Control And Decision Conference (CCDC), 2020, pp. 1826-1830, doi: 10.1109/CCDC49329.2020.9164275.

[8] M. Serra and J. Quelas, "Robust fuzzy gain schedulling PID implementation for gimbal stabilization system," 2017 XVII Workshop on Information Processing and Control (RPIC), 2017, pp. 1-6, doi: 10.23919/RPIC.2017.8214309.

[9] Teodorescu, M.; Teodorescu, H. N. (2021) Noncommutative Logic Systems with Applications in Management and Engineering. *International Journal of Computers, Communications & Control* . Feb2021, Vol. 16 Issue 1, p1-19.

[10] Teodorescu HN, Taylor and bi-local piecewise approximations with neuro-fuzzy systems. *Studies in Informatics and Control* 21 (4), 367-376.

ANNEX

P1. Proof of the property of low sensitivity of FLSs to the change in input m.f.s
Because of the symmetry of the input m.f.s, their c.o.g. will remain the same whatever the truncation value is, see Fig. 4. Because the input m.f.s also overlapp at most two by two, when the input has the value of a vertex of an input m.f. only a rule is fired and a single output m.f. is selected.

Because the output m.f.s are symmetric (the case of singletons is included), whatever are the truncation values produced by an input value confined between two successive vertices, the defuzzified values remain between the vertices of the output m.f.s., which are the abscissae of their c.o.g.s, for example y_{cog1} and y_{cog2} in

Fig. 4. This guarantees the property.

P2. Algebraic expressions of the I/O function of several classes of FLSs

As already discussed, if the input and output membership functions are piecewise linear or polynomial of degree less than 5, the input-output functions of Mamdani FLSs with c.o.g. defuzzification are piecewise ratios of polynomial functions of variables represented algebraically.

Next assume that the limits a of the output m.f.s are algebraic numbers, moreover that all the intersection points of the output m.f.s are at points represented by algebraic numbers. In addition, assume that the coefficients of the input and output polynomials are algebraic numbers. Then the Mamdani FLS maps the algebraic numbers of the input interval into a set of algebraic numbers.

The proof is qualitative: Because the input m.f.s are polynomials with algebraic coefficients, for any x_0 algebraic number, $c = Q(x_0)$ is algebraic.

Because the integrals $\int_a^{y_{01}(Q(x_0)} yP(y)dy$, $\int_a^{y_{01}(Q(x_0)} P(y)dy$ after integration generate polynomials and because the limits of the integrals are algebraic numbers, the results of integration are algebraic numbers.

Thus, an input algebraic number is mapped into an output algebraic number.

The last property remains true when the input polynomials have degree larger than 4, but maintain algebraic coefficients. The property is true for Sugeno FLS with singletons at algebraic numbers.

Horia-Nicolai Teodorescu's research interests cover aspects of artificial intelligence, including neuro-fuzzy systems, chaos technology, speech technology, image processing, and medical applications of AI, and analysis and prediction of social networks. He has co-edited more than 10 books with major publishers and authored or co-authored more than 250 papers.

Cite this paper as:
Teodorescu H.-N.(2022). Improve the design and testing of fuzzy systems with a set of (almost) simple rules, *International Journal of Computers Communications & Control*, 17(1), 4683, 2022.
https://doi.org/10.15837/ijccc.2022.1.4683