

A CAD TOOL FOR SIMULATION OF FUZZY CONTROLLERS

Jean J. Saade and Hassan B. Diab*

*Department of Electrical and Computer Engineering
Faculty of Engineering and Architecture
American University of Beirut, Lebanon*

الخلاصة :

تُقدِّم هذه المقالة برنامجاً سهّل الاستعمال يحوي على قائمة خيارات يحاكي أجهزة التحكم الضبابية (الغامضة) تُعرّف من قبل المستخدم. ويجري استعمال أجهزة التحكم الغامضة (الضبابية) فضلاً عن غيرها في حال عدم توفر نماذج حسابية مناسبة للعملية التي يجري التحكم بها وذلك بسبب تعقيد العملية واحتوائها عمليات لا خطية، إضافة إلى إمكانية تخميص النظام بشكل طبيعي. ولعلّ المرحلة الأكثر صعوبة في تصميم نظام تحكم ضبابي (غامض) هي تعيين وتعديل الوظائف العضوية للإدخال والإخراج وتحريير القواعد الاستنتاجية واختيار إستراتيجية إزالة الغموض المناسبة. ويُمكن برنامج التطبيق مُستخدِمه من تصميم أيّ جهاز تحكم ضبابي (غامض) بشكل منظم ومن ثم تخزين ومحاكاة وفحص وتقييم وتحسين أداء هذا الجهاز وذلك عن طريق اعتماد أسلوب «التجربة والخطأ» السهل التطبيق بحيث يتم ضبط النموذج الضبابي (الغامض) لإخراج الاستجابة المنشودة. هذا وتُقارن هذه المقالة نتائج استخدام استراتيجية جديدة لإزالة التغميض مع نتائج «قاعدة المعدل المثقل لينيماكس» (minimax weighted average formula) التي استُخدمت كطريقة لإزالة الضبابية (التغميض) في الكثير من حالات التحكم (الضبابية).

*Address for correspondence:

Department of Electrical and Computer Engineering
Faculty of Engineering and Architecture
American University of Beirut
P.O. Box 11-0236
Beirut, Lebanon.

ABSTRACT

This paper presents a user-friendly, menu-driven simulation tool for fuzzy user-defined controllers. Fuzzy controllers as well as other fuzzy systems are normally used when adequate crisp mathematical models for the process under control are not available because of process complexity and non-linearities, and when the system variables can naturally be fuzzified. The hardest part in the design of a fuzzy control system lies in the assignment and adjustment of input and output membership functions, the inference rules and also in the selection of a suitable defuzzification strategy. The software package allows the user to systematically design, save the description of, simulate, test, evaluate, and improve the performance of any fuzzy controller. This can be accomplished through an easy to implement *trial-and-error* methodology whereby the fuzzy model can be tuned to yield the desired output response. The paper also compares the results of a new defuzzification strategy with the minimax weighted average formula which has been utilized as a defuzzification method in many fuzzy control situations.

Keywords: Fuzzy sets, CAD, fuzzy logic controllers, inference rules, defuzzification.

A CAD TOOL FOR SIMULATION OF FUZZY CONTROLLERS

1. INTRODUCTION

As a result of publishing a paper [1] in which he introduced the theory of *fuzzy sets*, Professor L.A. Zadeh envisioned the application of his theory in system analysis to cope with vague, imprecise, and uncertain phenomena [2–4]. Fuzzy logic emerged from Zadeh's theory as a tool for modeling human reasoning [4, 5] and thus enabling machines to replace human operators in the execution of tasks which cannot be performed adequately by traditional logic-based machines.

Since the 1970s, fuzzy logic has been applied in various areas. Some applications include fuzzy flip-flops that can memorize fuzzy information [6], aircraft flight control [7], vehicle navigation [8–10], industrial processes [11], maintenance scheduling in transportation systems [12], automated drug delivery in muscle relaxant anaesthesia [13], water plant controllers [14], and gasoline refinery catalytic reformer controllers [15]. In 1990, products such as auto-focusing cameras, washing machines, and air conditioners were manufactured based on fuzzy logic. For instance, more than 70% of all washing machines, manufactured in 1990 by Matsushita Electric Industrial Co. in Osaka were based on fuzzy logic [16].

The CAD package presented in this paper is intended for use by engineers wishing to design fuzzy control systems or other types of systems that follow similar rules. Although the introduction of fuzziness to control systems plays an essential role represented by a reduction in their complexity level; *i.e.*, avoiding the use of complicated mathematical formulas, *etc.*, many problems remain facing the design of fuzzy controllers. These problems mainly relate to the assignment and adjustment of the membership functions that represent the different fuzzy or linguistic values of the input and output variables. They are also attached to the setting of the inference rules and the selection of a suitable defuzzification method.

It is worth noting here that the selection of the shape, degree of overlap and number of membership functions adopted for the input and output variables of a fuzzy controller has a significant effect on the performance of this controller, which could be represented by an input–output control curve or surface. Also, the controller performance is affected by the selection of the inference rules; that is, the correspondence between the antecedent parts of the rules and their consequents. The validation of these presented arguments could easily be obtained from the compositional rule of inference introduced by Zadeh [3].

The present software package allows the user to start the design of fuzzy controllers by considering a preliminary assignment of the fuzzy sets membership functions and rules and observing the corresponding output. The user can then easily introduce modifications to the membership functions and rules so as to modify the output and make it more consistent with his expectations and some reasonable performance criteria. This can be accomplished through the use of the options that are available in the package. The package also contains a new defuzzification strategy which has been recently introduced by the authors [17]. The results offered by this strategy are compared with those obtained using the minimax weighted average formula in some specific fuzzy control application.

2. USE AND APPLICABILITY OF FUZZY LOGIC

Fuzzy logic provides a unique and effective way to draw conclusions (system outputs) from vague and imprecise information (system inputs). Fuzzy logic is based on the concept of a fuzzy set which is characterized by a *membership function* taking values in the interval $[0,1]$ in place of 0 or 1 only as in *crisp sets*. Fuzzy statements, which form the backbone of fuzzy logic, are not, therefore, either true or false. Instead, they assume intermediate truth values represented by *membership grades*.

In general, fuzzy logic is best applied to non-linear, time-variant, ill-defined systems which can be represented by a series of IF–THEN rules. In such systems, fuzziness comes about naturally and conveniently in the system analysis [18]. In system analysis, the advantage of applying fuzzy logic over traditional logic is that using the latter, to completely describe a system, a detailed mathematical formulation that accounts for all the system variables is needed. This could be an impossible and time consuming task when systems as described above are of concern. On the other hand, in fuzzy design, one needs only conceive the relationship between input and output through a series of IF–THEN fuzzy inference rules. Therefore, the simplicity of modeling a control system with fuzzy logic can drastically reduce the complexity of the design and, as a result, require simpler software to implement the same decision-making capability, cutting down much of the design time.

3. ELEMENTS OF A FUZZY CONTROLLER

A fuzzy controller, as shown in Figure 1, accepts crisp inputs, fuzzifies them and performs some processing on the obtained fuzzy quantities to produce either fuzzy or crisp outputs. The *fuzzification* of the inputs of a fuzzy controller

consists of representing them by linguistic values in the form of fuzzy sets or statements with specified membership functions or grades respectively. Generally, the processing of these fuzzy inputs is accomplished through the use of inference rules and some fuzzy set operations.

To clarify the ideas, consider the membership function of the fuzzy set “hot temperature” as shown in Figure 2. A 38°C temperature, for instance, is considered “hot” with degree 0.8; hence, we can say that the *level of assertion* or *truth value* of the statement “the 38°C temperature is hot” is 0.8. In addition to such simple assertion statements, the linguistic logical operators such as “AND”, “OR”, and “NOT” are used to combine simple statements and obtain more complex ones. If we consider two statements denoted X and Y with x and y as their respective truth values, the logical operators can then be interpreted as in Table 1.

For example, the two statements “the 38°C temperature is hot” and “30% weather humidity is dry” with respective truth values 0.8 and 0.5 can be combined using the “AND” operator to give 0.5 as truth value. The truth value 0.8 is obtained if the “OR” operator is applied. Also, the truth values of the statements “the 38°C temperature is not hot” and “30% weather humidity is not dry” are 0.2 and 0.5 respectively. It is worth noting here that there has been other suggestions for the implementation of the logical “AND” and “OR” operators [19]. For the “AND” operator, the product rule has also been

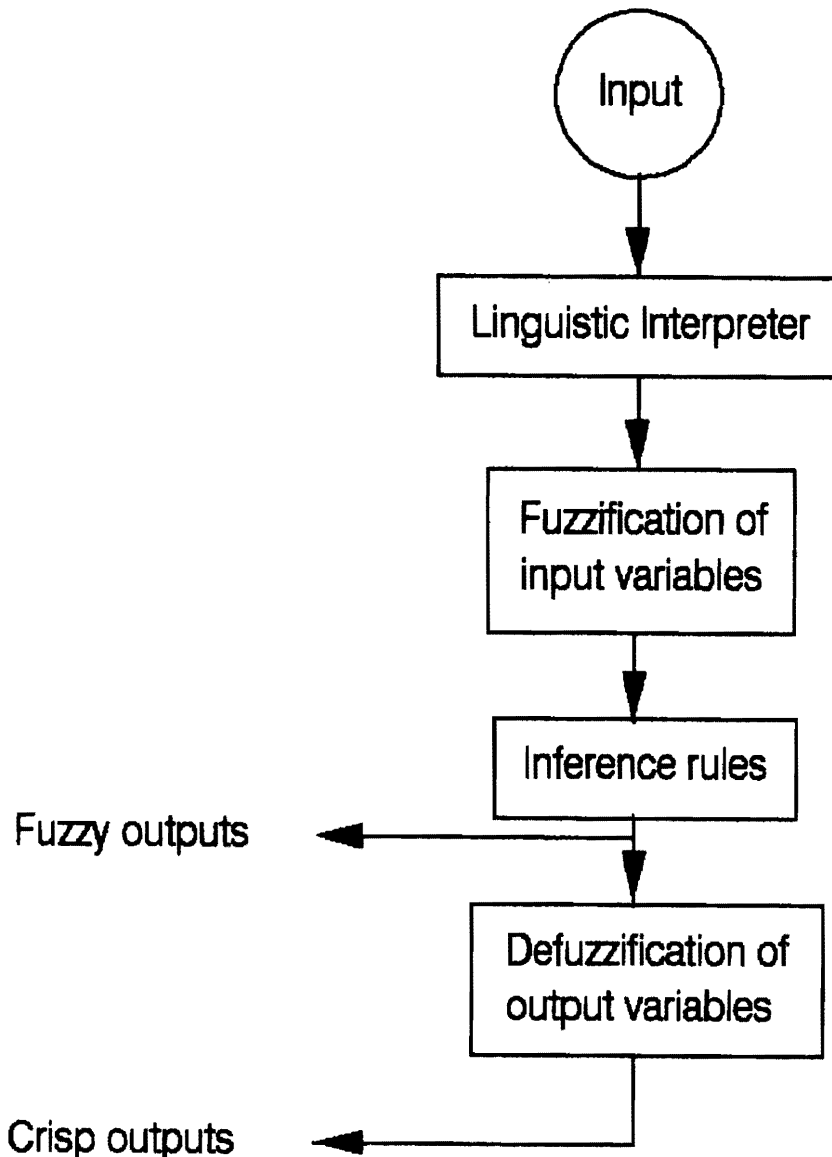


Figure 1. Block Diagram of a Fuzzy Logic-Based Controller.

noted among others as an alternative to the minimum rule. Also, the algebraic sum has been noted as an alternative way for implementing the “OR” operator.

Fuzzy inference rules describe dependent relationships among the system input and output variables by making use of fuzzy sets defined over these variables and the logical operators of Table 1. In the general form, an inference rule can be written as:

$$\text{If } \langle \text{condition} \rangle \text{ then } \langle \text{consequence} \rangle. \tag{1}$$

If, for example, in a temperature-control process, we have a fan to cool a motor, the variables may be the motor temperature and the speed of the fan. The linguistic values related to temperature may be “HOT”, “WARM”, and “COOL”, and those related to fan speed may be “OFF”, “ON-LOW”, and “ON-HIGH”. Actually, the number of linguistic values may be increased or decreased depending on the specific situation and the system designer’s objective. The same applies to the number and structure of the inference rules. Simple control inference rules may be:

Rule 1: *If <temperature is COOL> then <fan speed is OFF>*

Rule 2: *If <temperature is WARM> then <fan speed is ON-LOW>*

Rule 3: *If <temperature is HOT> then <fan speed is ON-HIGH>.* (2)

Table 1. Typical Fuzzy Logic Operators and Their Implementation.

Operation	Operator	Fuzzy Implementation
Intersection	X AND Y	MIN (x,y)
Union	X OR Y	MAX (x,y)
Complement	NOT X	1-x

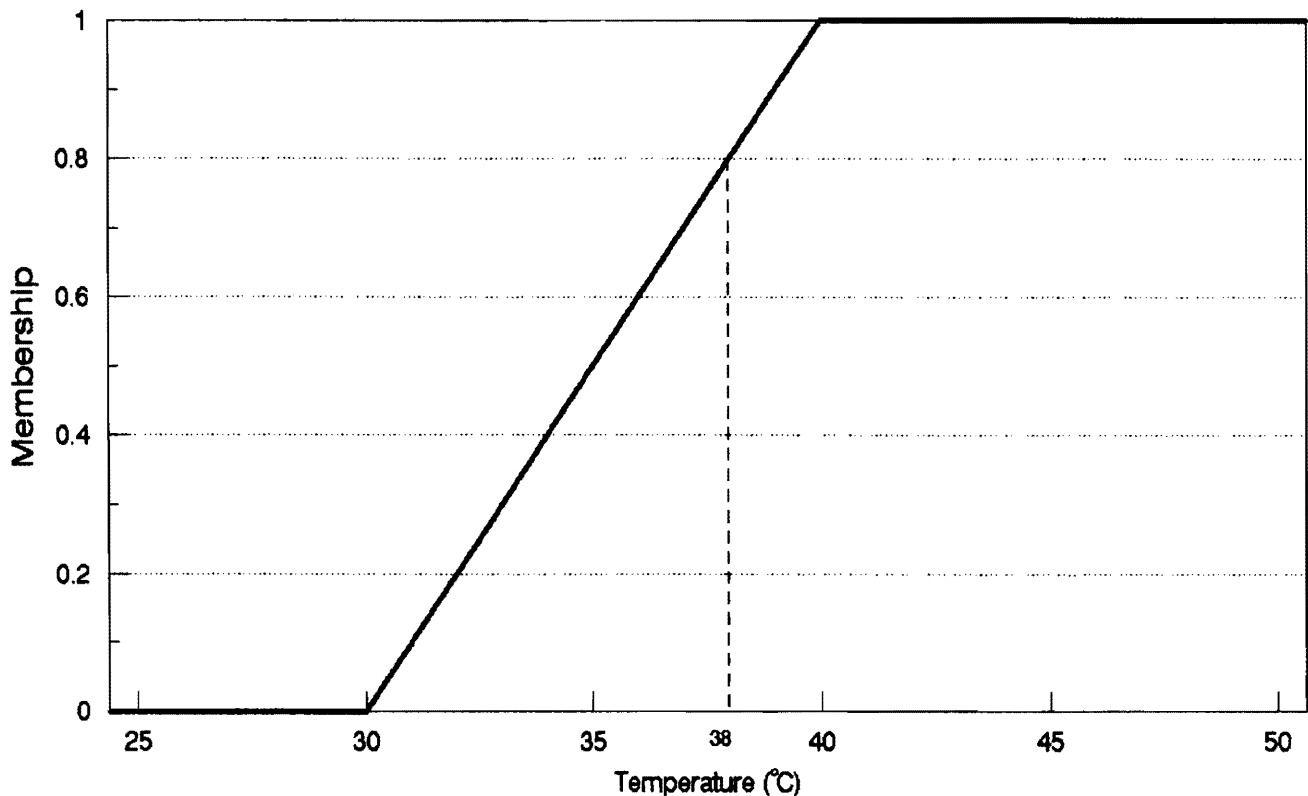


Figure 2. Membership Function of the Fuzzy Set “Hot Temperature”.

The above rules could be adequate when a single system input is considered. Evidently, the inference rules of Equation (2) become more involved if the system is supplied with more than one input.

The implementation of the inference rules expressed in Equation (2) or any other set of similar rules using the compositional rule of inference [3], results in a fuzzy output for each particular crisp input. For such an output to become available to the user in a concrete and useful manner, a defuzzification process should be implemented. Defuzzification, which consists of obtaining crisp outputs, could be introduced to the rules before an output is obtained, or could be applied after the fuzzy output has been observed. In the former method integrated into the package, there could be several defuzzification techniques. As described in [16], the use of the minimum rule for the "AND" operator and the maximum rule for the "OR" operator, followed by an averaging process results in the minimax weighted-average formula. The above noted operators arise in the setting of inference rules that account for more than one system input (see Section 7). A detailed description of an improved technique, which has been introduced by the authors in [17], is offered in Section 8.

4. ENVIRONMENT SPECIFICATION

The package runs on an IBM PC XT/AT or compatible computer. It requires about 100KB of memory, an Enhanced Graphics Adaptor (EGA), a mouse, and a color monitor. A single floppy disk is sufficient to run the program.

A user interface was designed to provide the user with a friendly environment to work with. The PC environment was chosen to provide an easy-to-use and cost-effective workstation, which, together with the package's user-friendly menu-driven structure, makes it a powerful CAD tool.

5. METHODOLOGY AND PACKAGE DESCRIPTION

The CAD tool described in this paper is intended for the simulation of controllers using a rule-based fuzzy technique. This package will, therefore, allow the user to design, simulate, and test any fuzzy controller in a systematic way. Upon observing the output of the fuzzy controller provided by the package, in its various forms, the user will be able to evaluate the performance of the control system under study and introduce the necessary changes to the membership functions and rules so that an improvement is achieved. An improved performance can be evaluated by the compatibility of the obtained control curve with some reasonably anticipated one. Section 7 elaborates on this issue.

Initially, the user must enter the input and output variables of the controller under study. Then, the membership functions are entered followed by the inference rules that govern the output behavior of the system. When all the data is entered, the user initiates a simulation run and graphically presents the output results in the form of input-output performance curves. As a result of analyzing the output, the user can make modifications in the rules and/or membership functions and initiates processing of another simulation run. This process can be repeated until the performance curve which is the closest possible to the desired one is attained.

The options provided in the master menu are: File, Variables, Rules, Memberships, Outputs, Help, and Demo. The user makes all his menu selections using a mouse. The File option allows file operations such as opening (loading) a new file, saving a file, and printing a file. A directory list of all *fuzzy files* are displayed with the extension ".CTL". Each file describes a fuzzy control system, namely the inputs, rules, membership functions, and outputs of a fuzzy controller. The Variables option allows the user to define (or edit) all the input and output variables which influence the behavior of the controller under study. The Rules option allows the user to define (or edit) the inference rules (that describe the transfer function of a conventional control system) which will control the output of the fuzzy system being designed. The format of the rules used is as shown in Equation (1). The *condition* field can be a compound statement combining more than one input variable using the linguistic logical operators. The *consequence* field can also be a compound statement combining two or more output variables. In order to overcome the need for an English-sentence linguistic compiler, which will further complicate the package, a simple input/output truth-table is provided to allow the user to describe the rules in a user-friendly manner.

Returning to the temperature control process example, we suppose, for instance, that the rules to be defined are:

- Rule 1: *If <temperature is A and humidity is B> then <fan speed is F>*
- Rule 2: *If <temperature is C and humidity is D> then <fan speed is E>*
- Rule 3: *If <temperature is C and humidity is B> then <fan speed is F>*
- Rule 4: *If <temperature is A and humidity is D> then <fan speed is E>.* (3)

From (3), there are two input variables, temperature and humidity. "A" and "C" are the linguistic values describing temperature, "B" and "D" are the linguistic values describing humidity, and "F" and "E" are the linguistic values that correspond to the speed. These rules can, in the truth-table form, be expressed as in Table 2. In rule 1, "A" and "B" are asserted and the output speed is "F". Therefore, the user must fill an "x" in the first row at columns A, B, and F. The user is able to edit, modify, delete, or add more rules at will.

The Membership option allows the user to enter the membership functions of the controller inputs under design. Membership functions are assumed, for simplicity, to be composed of linear segments (Figure 3). Hence, the user needs only define the coordinates of the end points (or breakpoints) of each line segment. Once the membership functions are defined, the user can visualize the membership function plots by selecting the *SHOW* option. The user can also modify already defined membership functions by adding, deleting, or changing the values of breakpoints.

The Output option allows the user to perform a simulation run of the fuzzy model. The results can be saved in a data file and viewed in the form of 2D graphs. The fuzzy model can use either the minimax weighted-average formula (refer to Section 7.1) or the new defuzzification formula introduced in [17] (refer to Section 7.2) to process the data. The simulation

Table 2. Truth Table of the Inference Rules of Equation (3).

	Inputs				Output	
	Temperature		Humidity		Speed	
	A	C	B	D	E	F
Rule 1:	x	.	x	.	.	x
Rule 2:	.	x	.	x	x	.
Rule 3:	.	x	x	.	.	x
Rule 4:	x	.	.	x	x	.

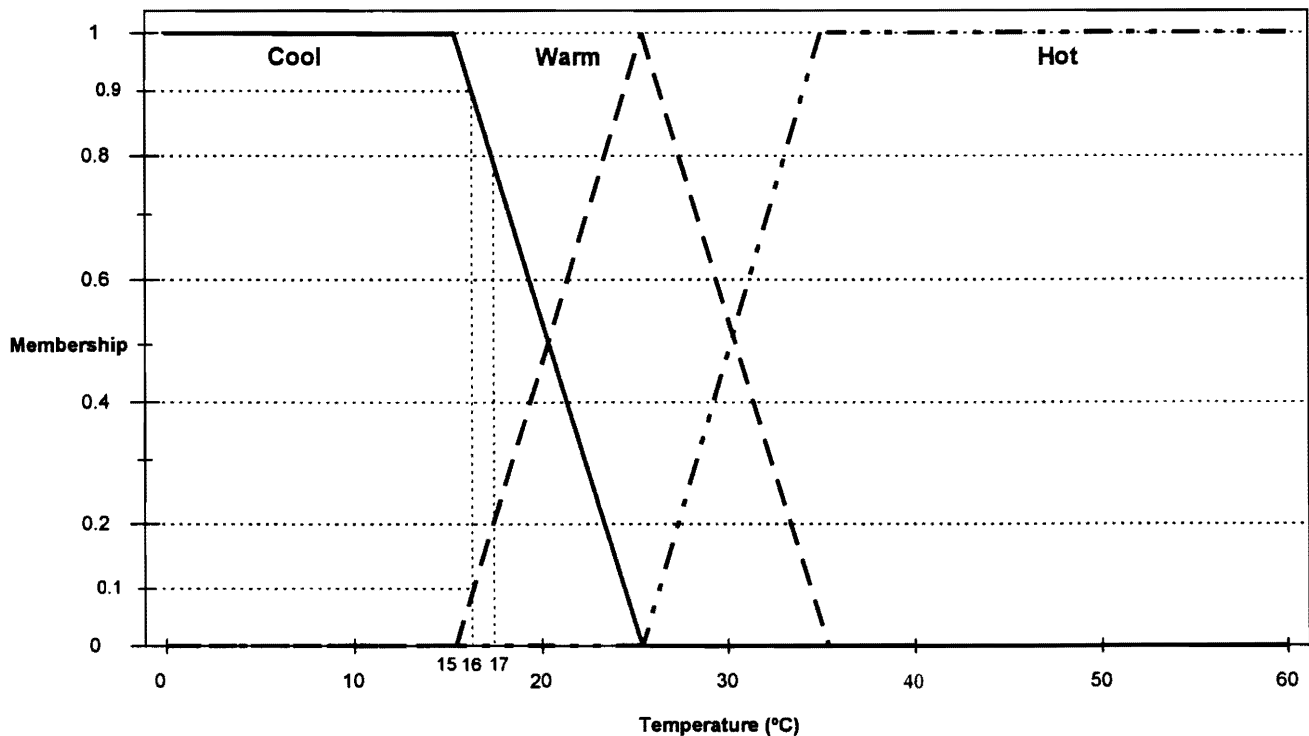


Figure 3. Linear Membership Functions of the Fuzzy Sets Cool, Warm, and Hot Representing the Different Fuzzy States of the Temperature Variable.

starts by opening a file into which the input and output variables, their ranges, the membership functions pertaining to each input variable, and the inference rules — defined by the user — are loaded.

If the minimax defuzzification method is applied, after the minimization is carried out, maximization resolves the competition between the various rules trying to govern the behavior of the same output. The *crisp* output is then computed by using a weighted-average formula.

When the procedure terminates, the results are stored into the data file. On the basis of the obtained results, the user can easily analyze the characteristics of the developed controller resulting from the selection of the specific membership functions and control rules.

The HELP option provides information on all the menu options, and a report on the current status of the system. Finally, the DEMO option provides the user with a demonstration on the use of fuzzy logic. The hierarchical structure of the package's menus is shown in Figure 4.

6. COMPARISON OF CAD TOOL WITH OTHER FUZZY TOOLS

There are many fuzzy software tools in the market today, some of which are listed here:

1. **FuziCost:** This is a fuzzy logic cost analysis modeling system. It can be used to build a cost model that combines your knowledge about cost, profit, and strategic performance. The system implements fuzzy logic descriptors of cost behaviors and provides for the integration of sales, distribution, and marketing.
2. **LEVEL5 Quest:** This is an intelligent fuzzy logic based query data mining tool for any database. It uses fuzzy logic engines and concept search maps and searches databases directly, with no preprocessing or indexing of the data required.

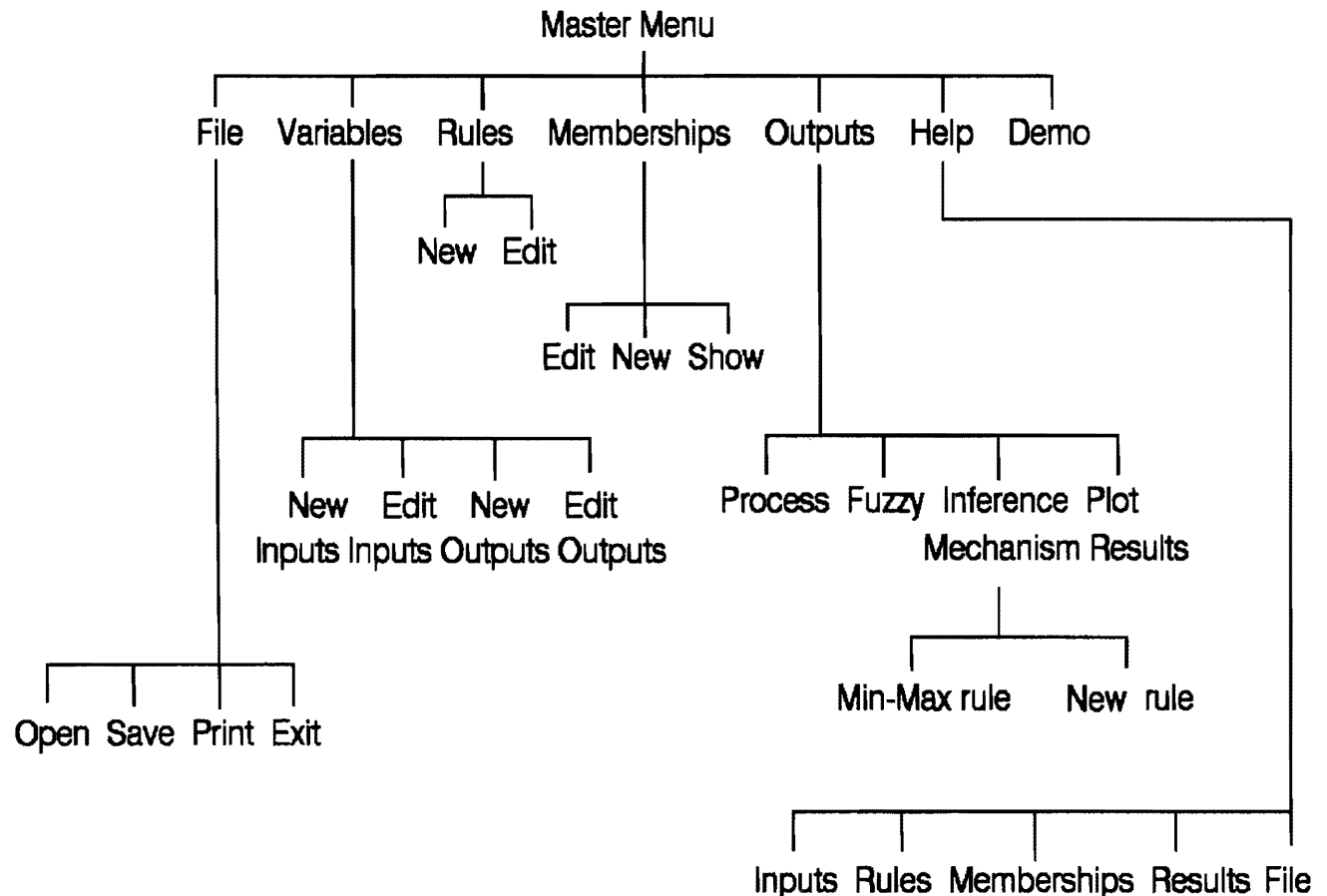


Figure 4. Menu Structure of the Package.

3. **NeuFuz4-C:** This is a software design/development system which combines elements of fuzzy logic with the learning and generalization capabilities of neural networks to learn system behavior, manipulate the information, and develop code that is then implemented in embedded applications.
4. **FuziCell:** This is a fuzzy logic plant scheduling system. It can be used to help managers deal with the uncertain and changing conditions found in the short lead time manufacturing environment. The tool evaluates back orders, new orders, plant load, supply conditions, and inventories using plant rules and experts' knowledge.
5. **FuziQuote:** This is a fuzzy logic quotation system that generates bids. The tool bases its quotation on three areas: (1) historical information on direct costs; (2) key knowledge from the best estimators; and (3) specific insight you apply at the time the quote is needed.
6. **Fuzzy Logic Development Environment for Embedded Systems (FLDE):** This is an application generator which aids in the design and testing of embedded fuzzy logic and C/C++ software.
7. **Fuzzy Logic Inferencing Toolkit (FLINT):** FLINT augments both Prolog and flex, providing tools for the construction of fuzzy expert systems and decision support applications.

The limitation of many fuzzy tools, such as the ones listed above, is that they are intended for specific applications rather than general-purpose. A summary of existing software fuzzy tools is provided in [22]. The CAD tool presented in this paper on the other hand, provides a fuzzy logic platform that is both user-friendly and can be used to design and simulate any user-specified system that can be defined by a set of inputs and outputs as well as inference rules expressing their relationships. Some of the general-purpose fuzzy tools are listed below:

1. **The Manifold Editor:** This is a knowledge acquisition tool that can be used to capture the expert judgements needed to build fuzzy systems. It consists of four major components: (1) The principle (rule) editor which is used to construct the principle-base; (2) the fuzzy set editor to create the fuzzy sets that define the values of the system's inputs and outputs; (3) the Manifold Walker, a built-in simulator which lets you see your design; (4) the Build module, used to create an output file of data that contains the knowledge-base of the designed fuzzy system.
2. **The Fuzzy Logic Toolbox (FLT) for MATLAB:** FLT provides facilities for the construction and evaluation of fuzzy logic control systems. Good for users already familiar with MATLAB.
3. **The Bytronic Fuzzy Control Package (BFCP):** This is a Windows-based development environment, designed to suit the specific needs of a fuzzy controller designer. It offers the user a visual rulebase structure and may be configured to control either internal simulations or external devices.

Being one of the most commonly used general-purpose fuzzy tools today, the FLT for MATLAB (FLTM) is chosen in order to make a comparison between the CAD tool presented in this paper and a similar tool available on the market. Accordingly, Section 6.1 is dedicated to this discussion. Other tools require a minimum platform specification which is not needed for our CAD tool. For example, the BFCP minimum specification is a PC-486 33MHz with 4MB of RAM and Windows 3.1 or higher. While our CAD tool can operate on any PC XT/AT with only 2MB of RAM and under MS-DOS.

6.1. A Comparison Between the CAD Tool and the FLTM

The first comment to be made about the FLTM is that it requires MATLAB vs. 4.21c1 in order to run. The features can be summarized by the following:

1. Define variables, fuzzy rules, and membership functions.
2. Interactively view fuzzy output surface and inference process.
3. Advanced methods such as Adaptive Neuro-Fuzzy Inference Systems (ANFIS), fuzzy clustering (FC), and Sugeno inferencing.
4. Easy to use.

All of the above features are found in our CAD tool except for item 3, which allows the use of ANFIS for shaping membership functions automatically by training them with input data, and applies FC for specific applications such as pattern recognition, and finally facilitates the Sugeno inferencing method for the creation of hybrid fuzzy systems. The CAD tool presented, however, only deals with *pure* fuzzy logic systems without the use of other techniques such as neural networks or genetic algorithms. Some of the comments pertaining to the enhancements and bug fixes for the FLTM vs. 1.0

are included on the Internet in the “Fuzzy Logic Toolbox User’s Page” whose URL is <http://www.mathworks.com/support/updates/fuzzyupdate.html>

The presented CAD tool allows the evaluation and comparison of all of the possible solutions and variables associated with complex real-life design problems, in a simple intuitive manner. The CAD tool developed has many features that makes it attractive. This includes the fact that a very simple platform is required for the tool without the need for any additional software modules as well as the low cost user-friendly environment it offers. The main limitation of the tool is that membership functions can only be linearly-shaped, *i.e.*, triangular and trapezoidal, *etc.* Hence, observing and testing the effect of changing the shape of the membership functions with the presented CAD tool cannot be done with the same degree of simplicity.

Although the fuzzy inference system can be built by command lines, it is much easier to do it graphically. In the FLTM, there are five Graphical User Interface (GUI) tools for building, editing, and observing the fuzzy inference system. All the GUI tools have the same menu that allows saving, opening, and editing a fuzzy inference system using any of the GUI tools. They also have a help button that gives some information about how the GUI tool works and a close button that dismisses the window. In addition, a status bar describes the most recent operation in the window. The following represent the FLTM GUI tools which are also facilitated by the presented CAD tool.

1. The Fuzzy Inference System (FIS) Editor

The FIS editor displays general information about a fuzzy inference system. A diagram at the top shows the names of each input variable and each output variable along with a block diagram showing the inference system. Below the diagram is the name of the system and the type of inference used. The type of inference could be a Mamdani or Sugeno inference method.

2. The Membership Function Editor

This allows the display and editing of all the membership functions of the fuzzy inference system, including both input and output variables. A specified membership function is selected by clicking on it with the mouse. The user can change any membership function’s attributes, including name, type, range, and numerical parameters.

3. The Rule Editor

This window contains a large editable text field for displaying, editing, and parsing the rules.

4. The Rule Viewer

This is a window that displays the whole inference process. It contains a single figure window with several small plots nested in it. In the lower left corner, there is a text field where the user can enter a specific input value if desired. He can thus see, the result or the output corresponding to this specific value. The rule viewer is important in showing how the shape of the membership functions is influencing the overall result. It performs well with up to 30 rules and as many as 7 variables. However, because it plots every part of each rule, for all rules, it becomes unwieldy for a large number of rules.

5. The Surface Viewer

Upon opening the surface viewer, the user is presented with a two-dimensional or three-dimensional curve that represents the mapping from input to output. In a single-input, single-output system, the entire mapping can be seen in one two-dimensional plot, and in a two-input, one-output system, it can be seen in one three-dimensional plot. In moving beyond this limit, the surface viewer has pop-up menus that let the user choose any two inputs and one output for plotting.

7. CASE STUDY

This section presents a case study on a fuzzy controller for a fan and specifically emphasizes the results obtained by the weighted-average formula and the new formula introduced in [17], as two alternative defuzzification methods integrated in the CAD tool. As noted previously (Section 5), the first step involves defining the input and output variables of the system. Let the input variables be the temperature and humidity, and the output variable be the speed of the fan.

The used fuzzy sets or linguistic values are “COOL”, “WARM”, and “HOT” for the temperature, “DRY”, “MOIST”, and “WET” for the humidity, and “OFF”, “ON-LOW”, and “ON-HIGH” for the speed. As a result of selecting the variables and

their associated linguistic values, the corresponding membership functions can be assigned. It is important to note here that overlapping fuzzy sets, which reflect the different fuzzy states of a single input variable, permit a control system to operate smoothly [19]. Therefore, choosing the shape and inflection points of each membership function constitutes a major part of the design process and this is available in the CAD tool. In one case, overlapping of the "COOL" and "WARM" membership functions leads to 20°C being cool with grade 0.5 and warm with grade 0.5 (Figure 3).

Now the inference control rules are defined for the particular case we are considering. These rules, which are of the form described in Equation (1), are as follows:

- Rule 1: *If <temperature is COOL and humidity is DRY> then <fan speed is OFF>*
 Rule 2: *If <temperature is COOL and humidity is MOIST> then <fan speed is ON-LOW>*
 Rule 3: *If <temperature is COOL and humidity is WET> then <fan speed is ON-LOW>*
 Rule 4: *If <temperature is WARM and humidity is DRY> then <fan speed is ON-LOW>*
 Rule 5: *If <temperature is WARM and humidity is MOIST> then <fan speed is ON-HIGH>*
 Rule 6: *If <temperature is WARM and humidity is WET> then <fan speed is ON-HIGH>*
 Rule 7: *If <temperature is HOT> then <fan speed is ON-HIGH>.* (4)

The processing applied on the inputs involves a defuzzification algorithm. The following describes the options facilitated by the package for the defuzzification process.

7.1. Minimax Weighted Average Formula

The minimax rule consists of applying the minimum operator if AND is used in the condition part of the rule. Among each set of rules controlling the same output variable, the rule that is satisfied to the highest grade (*i.e.* maximum) is chosen. For example, assume that the input temperature is 16°C and humidity is 27.5%. Then, as shown in Figure 3, the temperature is 0.9 cool, 0.1 warm, and 0.0 hot, while the humidity (Figure 5) is 0.75 dry, 0.25 moist, and 0.0 wet. Therefore, rule 1 of Equation (4) will be satisfied with degree $\text{MIN}(0.9, 0.75) = 0.75$. Table 3 shows the results (for this example) of all the rules of Equation (4).

From Table 3, it can be noted that rules 2, 3, and 4, for example, both compete for controlling the same output value (ON-LOW). To resolve this matter, the rule that is satisfied to the highest grade is selected. This is equivalent to OR-ing or taking the maximum in the ON-LOW column, which in this case is $\text{MAX}(0.1, 0.25, 0.0) = 0.25$. Therefore, rule 2 is selected.

Now in order to defuzzify the output (obtain crisp results), a common technique that can be applied is as follows: Let the fan speed be defuzzified as indicated below:

$$\begin{aligned} S\text{-OFF} &= S_o\% \text{ of the maximum speed} \\ S\text{-ON-LOW} &= S_l\% \text{ of the maximum speed} \\ S\text{-ON-HIGH} &= S_h\% \text{ of the maximum speed.} \end{aligned} \quad (5)$$

Let the truth values obtained by the minimax algorithm be μ_o for OFF, μ_l for ON-LOW, and μ_h for ON-HIGH; then the fan speed can be obtained by applying a weighted-average formula:

$$\text{fan speed} = \frac{(\mu_o * S_o) + (\mu_l * S_l) + (\mu_h * S_h)}{\mu_o + \mu_l + \mu_h} \quad (6)$$

In our example,

$$\begin{aligned} \mu_o &= 0.75 \\ \mu_l &= \text{MAX}(0.25, 0.0, 1.0) = 0.25 \\ \mu_h &= \text{MAX}(0.0, 0.1, 0.0) = 0.1. \end{aligned} \quad (7)$$

Thus, if we assume $S_o = 0\%$, $S_l = 40\%$, and $S_h = 100\%$, as used in the package, then:

$$fan\ speed = \frac{(0.75 * 0.0) + (0.25 * 40\%) + (0.1 * 100\%)}{0.0 + 0.25 + 0.1} = 57.14\%. \tag{8}$$

Figure 6 shows a plot of the fan speed vs. temperature for an input humidity of 25%. This plot is indicated as the minimax formula curve. The membership functions for the temperature and humidity values are as shown in Figures 3 and 5 respectively. Figure 7 shows a plot of the speed vs. temperature for an input humidity of 70%.

Table 3. Results of the Rules of Equation (4) with Truth Values as in Section 7.1.

	SPEED		
	OFF	ON-LOW	ON-HIGH
Rule 1: MIN(0.9, 0.75)	0.75	-	-
Rule 2: MIN(0.9, 0.25)	-	0.25	-
Rule 3: MIN(0.9, 0.0)	-	0.0	-
Rule 4: MIN(0.1, 0.75)	-	0.1	-
Rule 5: MIN(0.1, 0.25)	-	-	0.1
Rule 6: MIN(0.1, 0.0)	-	-	0.0
Rule 7: MIN(0.0)	-	-	0.0

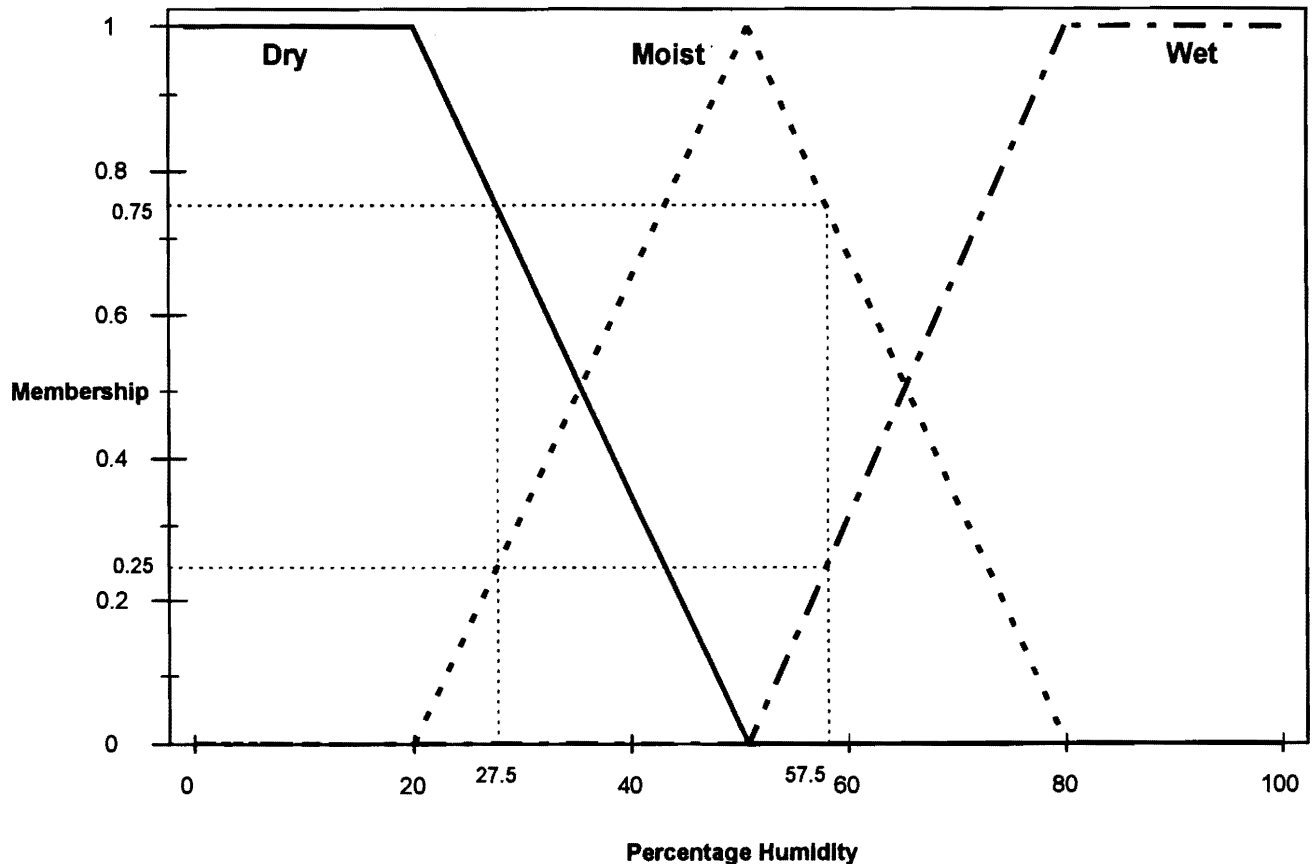


Figure 5. Linear Membership Functions for the Fuzzy Sets Dry, Moist, and Wet Representing Different Fuzzy States of the Humidity Variable.

It is to be noted here that the input–output control curves of the fuzzy controller shown in Figures 6 and 7 can be easily modified using the described CAD package by changing the spread and/or the number of membership functions shown in Figures 3 and 5 and by changing the inference rules expressed in Equation (4). Since, by the virtue of this change the grades of membership of the same crisp temperature and humidity values would change. As a result, μ_o , μ_t , and μ_h would also change and formula (6) would lead to a different output value.

This change in the control curve can be interpreted as a change in the performance of the fuzzy controller. The CAD tool provides a convenient trial and error methodology by which the controller performance can be improved and made closer to some desired curve. Such a desired control curve could have some prescribed characteristics which need to be satisfied, or specified using a set of desired input–output data points made available by an expert. If we look at Figure 7, for instance, the portion of the minimax formula curve where the fan speed decreases for an increase in temperature does not look reasonable in this particular application.

8. THE NEW DEFUZZIFICATION FORMULA

In the minimax-weighted average method discussed in the previous section, the undesired decrease in the output fan speed as temperature increases, for a constant humidity (Figure 7), seems related to the application of the minimax formula and particularly the maximum rule. The maximization in the column of a single output variable (see Table 3) to resolve the conflict between competing rules is questionable. Is it truly the maximum operation that needs to be thought about in this instance or another kind of operation? This does not seem to be a simple matter to solve since it relates to the suitability of the operation used to translate the “OR” operator (that is introduced between the rules). Such an issue has been extensively addressed by Zimmermann in [19] and no solutions that would accommodate all fuzzy control situations have been offered.

All what can be said in this instance is that the maximization considers the contribution of only one rule to a specific output value and disregards contributions from other rules to this output. Besides, the division by the sum ($\mu_o + \mu_t + \mu_h$) in

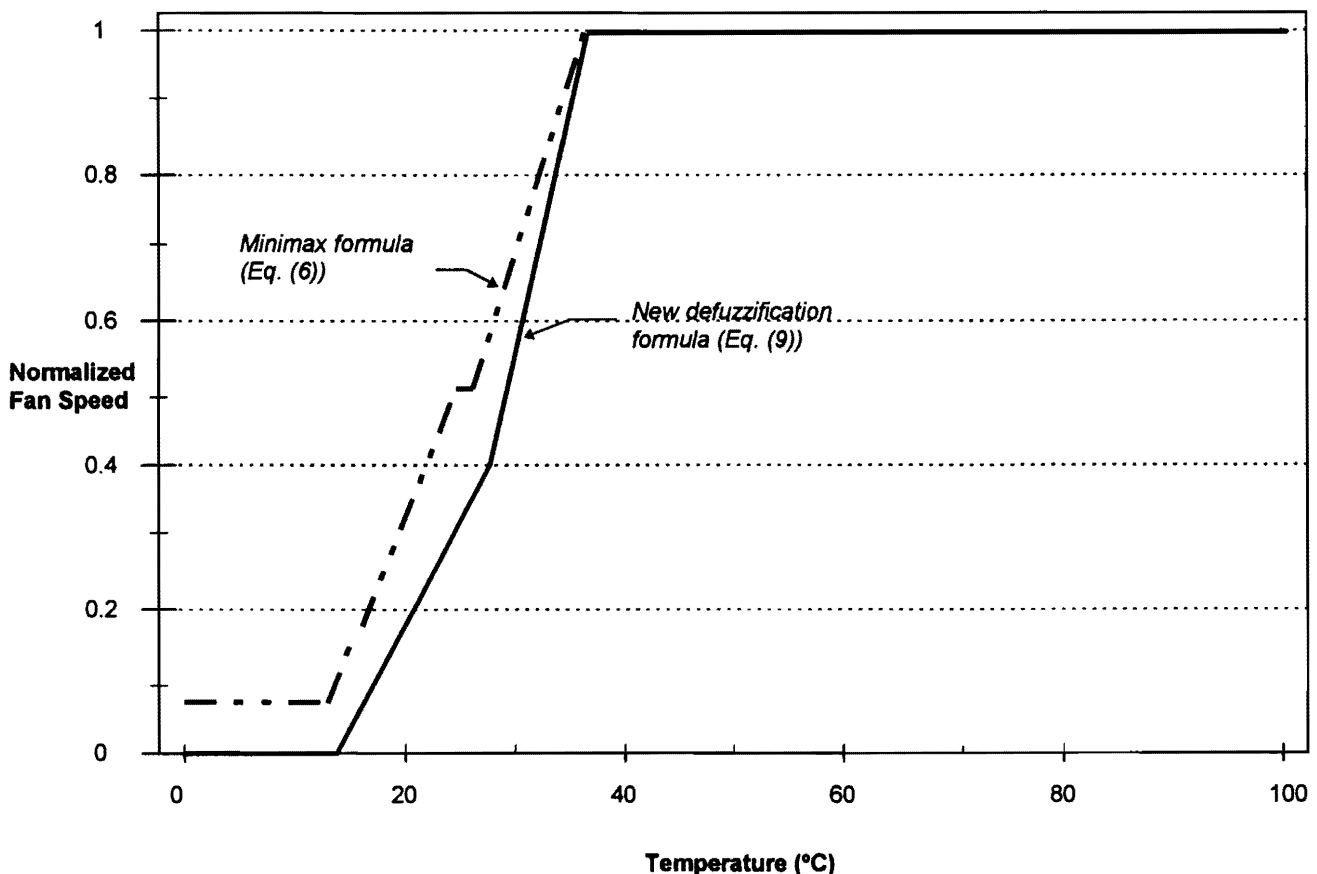


Figure 6. Fan Speed vs. Temperature for a Humidity of 25% and Input Memberships as in Figures 3 and 5.

Equation (6) seems to be adopted in order to transform the possibility distribution represented by the membership grades μ_o , μ_t , and μ_h , which respectively correspond to the output values S_o , S_t , and S_h , into a probability distribution that goes along with the averaging process. This type of transformation is also questionable since the problem is inherently non-probabilistic.

As a result of the above-noted discussion, we consider a more reasonable alternative approach to the defuzzification problem. The defuzzification method that is described here is the one offered in [17] with a complete justification. It has the objective of avoiding the competition between rules controlling the same output and hence the application of the maximum operation; and also avoiding the transformation of the problem from its possibilistic nature to a probabilistic one. We define the following correspondence between temperature and fan speed: *cool* $\rightarrow S_{co}\%$, *warm* $\rightarrow S_{wa}\%$, *hot* $\rightarrow S_{ho}\%$. Similarly, the correspondence between humidity and fan speed is: *dry* $\rightarrow S_{dr}\%$, *moist* $\rightarrow S_{mo}\%$, *wet* $\rightarrow S_{we}\%$. To account for all combinations of the different input values, the output can, in a way, be obtained by taking the average of temperature and humidity pairs.

For example, if temperature is cool and humidity is moist, the speed will be $\frac{(S_{co} + S_{mo})}{2}\%$. Table 4 shows the results for all input combinations.

Now, instead of taking the minimum rule as before, for combining two or more inputs, the product of their membership grades is applied. For example, if a crisp temperature value is μ_{t1} cool, μ_{t2} warm, and μ_{t3} hot and a crisp humidity value is μ_{h1} dry, μ_{h2} moist, and μ_{h3} wet, then combining both temperature and humidity by the product, Table 5 is obtained.

Requiring that the sum of the membership grades of any crisp input value in the different fuzzy sets defined for the same input be 1 (see Figures 3 and 5), then $\mu_{t1} + \mu_{t2} + \mu_{t3} = 1$ and also $\mu_{h1} + \mu_{h2} + \mu_{h3} = 1$. Hence, the sum of the membership grades listed in Table 5 is always 1. The weighted average formula can now be applied in an alternative and improved form relevant to the noted specifications to give:

$$\begin{aligned}
 \text{fan speed} = & \mu_{11} * \frac{S_{co} + S_{dr}}{2} + \mu_{12} * \frac{S_{co} + S_{mo}}{2} + \mu_{13} * \frac{S_{co} + S_{we}}{2} \\
 & + \mu_{21} * \frac{S_{wa} + S_{dr}}{2} + \mu_{22} * \frac{S_{wa} + S_{mo}}{2} + \mu_{23} * \frac{S_{wa} + S_{we}}{2} \\
 & + \mu_{31} * \frac{S_{ho} + S_{dr}}{2} + \mu_{32} * \frac{S_{ho} + S_{mo}}{2} + \mu_{33} * \frac{S_{ho} + S_{we}}{2} .
 \end{aligned} \tag{9}$$

There is no need to divide here by the sum of membership grades (as in Equation (6)) since it is 1. Thus, the probabilistic averaging in Equation (6) has been avoided. If we consider the particular case of Figure 3 where the temperature value is

Table 4. Fan Speeds for all Different Combinations of Temperatures and Humidities.

Temperature	Humidity	Fan Speed
Cool	Dry	$(S_{co} + S_{dr})/2\%$
Cool	Moist	$(S_{co} + S_{mo})/2\%$
Cool	Wet	$(S_{co} + S_{we})/2\%$
Warm	Dry	$(S_{wa} + S_{dr})/2\%$
Warm	Moist	$(S_{wa} + S_{mo})/2\%$
Warm	Wet	$(S_{wa} + S_{we})/2\%$
Hot	Dry	$(S_{ho} + S_{dr})/2\%$
Hot	Moist	$(S_{ho} + S_{mo})/2\%$
Hot	Wet	$(S_{ho} + S_{we})/2\%$

17°C and Figure 5 with humidity value as 57.5%, then the membership entries in Table 5 become 0.0, 0.6, 0.2, 0.0, 0.15, 0.05, 0.0, 0.0, and 0.0 respectively. The fan speed entries of the same Table become 0%, 20%, 50%, 20%, 40%, 70%, 50%, 70%, and 100% respectively. These latter values result from the assumption that $S_{co} = 0\%$, $S_{wa} = 40\%$, $S_{ho} = 100\%$, $S_{dr} = 0\%$, $S_{mo} = 40\%$, and $S_{we} = 100\%$. Therefore, the fan speed corresponding to the temperature of 17°C and 57.5% humidity is:

$$\begin{aligned} fan\ speed &= (0*0) + (0.6*20) + (0.2*50) + (0*20) \\ &+ (0.15*40) + (0.05*70) + (0*50) + (0*70) + (0*100) = 31.5\% \end{aligned} \tag{10}$$

Table 5. Memberships and Fan Speeds for all Combinations of Temperatures and Humidities.

Combination	Membership	Fan Speed
Cool and dry	$\mu_{i1} * \mu_{h1} = \mu_{11}$	$(S_{co} + S_{dr})/2\%$
Cool and moist	$\mu_{i1} * \mu_{h2} = \mu_{12}$	$(S_{co} + S_{mo})/2\%$
Cool and wet	$\mu_{i1} * \mu_{h3} = \mu_{13}$	$(S_{co} + S_{we})/2\%$
Warm and dry	$\mu_{i2} * \mu_{h1} = \mu_{21}$	$(S_{wa} + S_{dr})/2\%$
Warm and moist	$\mu_{i2} * \mu_{h2} = \mu_{22}$	$(S_{wa} + S_{mo})/2\%$
Warm and wet	$\mu_{i2} * \mu_{h3} = \mu_{23}$	$(S_{wa} + S_{we})/2\%$
Hot and dry	$\mu_{i3} * \mu_{h1} = \mu_{31}$	$(S_{ho} + S_{dr})/2\%$
Hot and moist	$\mu_{i3} * \mu_{h2} = \mu_{32}$	$(S_{ho} + S_{mo})/2\%$
Hot and wet	$\mu_{i3} * \mu_{h3} = \mu_{33}$	$(S_{ho} + S_{we})/2\%$

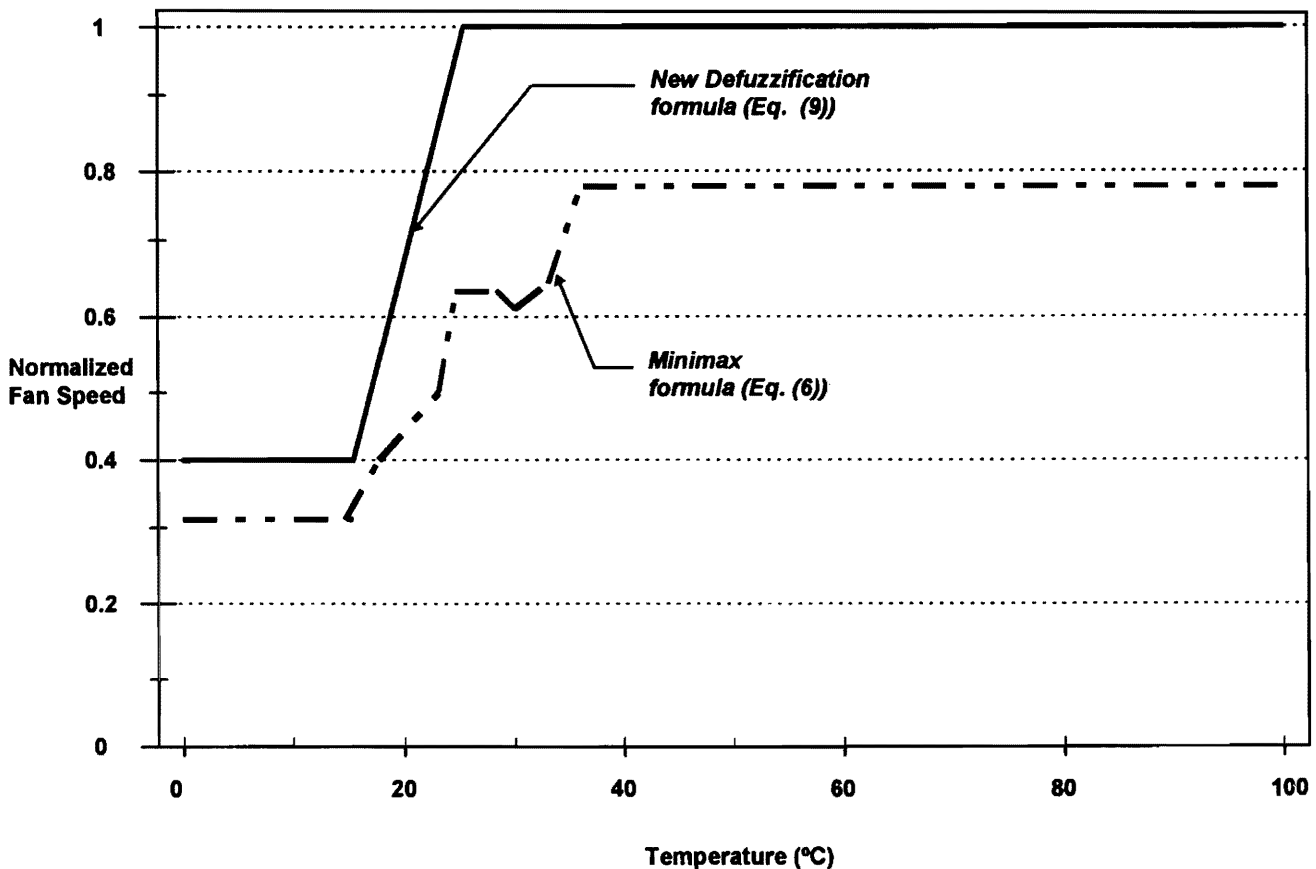


Figure 7. Fan Speed vs. Temperature for a Humidity of 70% and Input Memberships as in Figures 3 and 5.

The plots of fan speed vs. temperature for humidity values of 25% and 70% are shown in Figures 6 and 7 respectively. As seen in Figure 7, the disadvantage related to the application of the minimax weighted average formula (Equation (6)) is not present when the new formula expressed in Equation (9) is used. For increasing temperature, fan speed always increases. The new defuzzification formula (9) also allows for an adjustment of the input–output control curve by changing the input membership functions of the fuzzy controller. This change can easily be accommodated by the options available in the presented CAD tool (see Section 5).

9. CONCLUSION

This paper has presented a CAD tool for fuzzy logic controllers. The tool, mainly allows the design, simulation, and evaluation of the performance of the controller through graphical illustrations. It also enables the designer to modify and improve the performance of a fuzzy control system using the available package options, which permit rapid changes in the input membership functions and inference rules. Performance improvement can also be obtained as shown in Figure 7 using the described new defuzzification method (Section 8) which is also integrated in the CAD package.

The package presents, therefore, a very convenient and powerful methodology for designing controllers for complex, ill-defined and non-linear processes in a simple and user-friendly manner. All that is needed in order to check on the performance of the fuzzy control system simulated with the aid of the package, is some type of description of the process input–output characteristic or available input–output data points. These can be obtained from a human expert who is assumed to be familiar with the process under control.

Also, the tool provides the designer with a fast simulation turnaround time. This is a very important aspect when the classical non-linear model of the controller is too complex, demanding extensive computing power and, therefore, involving expensive hardware resources to carry out the simulation.

A natural extension to this work may include a linguistic interpreter [21] to allow the user to enter English language phrases, and a graphical interface tool to define the membership functions graphically. Further research could also address defuzzification methods applied to fuzzy outputs, modification of the package to cope with adaptive fuzzy systems, and specific procedures for performance evaluation. This latter could include the introduction of some performance error index whose minimization results in an optimum performance. It could also consist of building an interface for controller verification through the use of a closed loop in which the process to be controlled is included.

ACKNOWLEDGEMENTS

The authors would like to thank Ramzi Labban, Hala Rachid, and Younes Younes for some of the computations they have carried out.

REFERENCES

- [1] L.A. Zadeh, "Fuzzy Sets", *Information and Control*, **8** (1965), pp. 338–353.
- [2] L.A. Zadeh, "Toward a Theory of Fuzzy Systems", *Report No. ERL-69-2*. Electronics Research Lab., University of California, Berkeley, 1969.
- [3] L.A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", *IEEE Trans. Systems, Man, and Cybernetics*, **SMC-3(1)** (1973), pp. 28–44.
- [4] L.A. Zadeh, "Making Computers Think Like People", *IEEE Spectrum*, August 1984, pp. 26–32.
- [5] L.A. Zadeh, "Fuzzy Logic and Approximate Reasoning", *Synthese*, **30** (1975), pp. 407–428.
- [6] K. Hirota and K. Ozawa, "The Concept of Fuzzy Flip-Flop", *IEEE Trans. on Systems, Man, and Cybernetics*, **19(5)** (1989), pp. 980–997.
- [7] L. Larkin, "A Fuzzy Logic Controller for Aircraft Flight Control", *Proceedings of 23rd Conf. on Decision and Control, Las Vegas, NV*, December 1984, pp. 894–897.
- [8] R. Hogle and P. Bonissone, "A Fuzzy Algorithm for Path Selection in Autonomous Vehicle Navigation", *Proceedings of 23rd Conf. on Decision and Control, Las Vegas, NV*, December 1984, pp. 898–900.
- [9] M. Sugeno and K. Murakami, "Fuzzy Parking Control of Model Car", *Proceedings of 23rd Conf. on Decision and Control, Las Vegas, NV*, December 1984, pp. 902–903.
- [10] U. Kramer, "On the Application of Fuzzy Sets to the Analysis of the System Driver-Vehicle Environment", *Automatica*, **21** (1985), pp. 101–107.

- [11] P. King and E. Mamdani, "The Application of Fuzzy Control Systems to Industrial Processes", *6th IFAC Congress on Control Technology in the Service of Man, Boston, MA*, August 1975, pp. 235–242.
- [12] M. Etschmaier, "Fuzzy Controls for Maintenance Scheduling in Transportation Systems", *Automatica*, 1980, pp. 255–264.
- [13] D. Linkens, "Self-Organising Fuzzy Logic Control and Application to Muscle Relaxant Anaesthesia", *IEE Proceedings-D*, **138**(3) (1991), pp. 274–284.
- [14] W. Kickert and H. Van Nauta Lemke, "Application of a Fuzzy Controller in a Warm Water Plant", *Automatica*, **16** (1980), pp. 301–308.
- [15] W. Bare, R. Mulholland, and S. Sofer, "Design of a Self-Tuning Rule Based Controller for a Gasoline Refinery Catalytic Reformer", *IEEE Trans. on Automatic Control*, **35**(2) (1990), pp. 156–164.
- [16] K. Self, "Designing with Fuzzy Logic", *IEEE Spectrum*, November 1990, pp. 42–44.
- [17] J.J. Saade and H. Diab, "Defuzzification Techniques for Fuzzy Controllers", submitted to *IEEE Trans. Systems, Man, and Cybernetics*.
- [18] K. Sugiyama, "Rule-Based Self-Organizing Controller", in *Fuzzy Computing Theory, Hardware and Applications*. ed. M.M. Gupta and T. Yamakawa. Amsterdam: Elsevier Science Pubs., 1988, pp. 341–353.
- [19] H.J. Zimmermann, *Fuzzy Set Theory and its Applications*, 2nd edn. Dordrecht: Kluwer Academic Pubs., 1991.
- [20] K. Anderson, "Control Systems Sample Life in the Fuzzy Lane", *Personal Engineering and Instrumentation News*, October 1992, pp. 78–81.
- [21] M. Braae and D.A. Rutherford, "Theoretical and Linguistic Aspects of the Fuzzy Logic Controller", *Automatica*, **15** (1979), pp. 553–577.
- [22] W. Pedrycz, *Fuzzy Sets Engineering*. Boca Raton: CRC Press, 1995, Ch. 7.

Paper Received 5 October 1996; Revised 15 June 1997; Accepted 13 December 1997.