

FL-LAB v2: Design and Development of an Easy-to-Use, Interactive Fuzzy Logic Control Software System

Utku Kose^{1,*} and Omer Deperlioglu²

¹Directorate of Computer Center, Usak University, Usak, Turkey

²Afyon Vocational School, Afyon Kocatepe University, Afyonkarahisar, Turkey

Received: 8 Jun. 2014, Revised: 6 Sep. 2014, Accepted: 8 Sep. 2014

Published online: 1 Mar. 2015

Abstract: The Fuzzy Logic or the Fuzzy Logic Control is an Artificial Intelligence approach/technique, which is especially used for designing and developing intelligent controlling systems. It provides an effective and efficient method to simulate the human thinking and behaviors in order to ensure the related intelligent controlling structure. In this context, this paper introduces the FL-LAB v2: a software system, which can be used to design and develop different kinds of Fuzzy Logic inference and controlling systems, by using an easy-to-use, interactive and effective software environment. This software system is the second version of pre-introduced software system and it provides more predefined controls to adjust and realize basic features and functions of a typical Fuzzy Logic inference or control system, without having any foreknowledge about the related subject area. Furthermore, the software system has also been designed on simple but visually improved Windows-form application interfaces. Generally, FL-LAB v2 system is appropriate to be used by most people interested in the Fuzzy Logic and the Fuzzy Logic Control. It ensures an alternative approach to more technical programs based on the related approach/technique.

Keywords: artificial intelligence, fuzzy logic, fuzzy control, software system, engineering applications

1 Introduction

Artificial Intelligence is a popular Computer Science field in which many different scientific works and development are realized. In this sense, Artificial Intelligence approaches and the related techniques are widely used by researchers to design and develop different types of intelligent applications or systems. These applications and systems are too important and critical to allow solving encountered, complex real world-based problems via more accurate and advanced algorithmic and logic approaches. Because of this situation, Artificial Intelligence field is connected with many different types of areas in current modern world. At this point, many different approaches and techniques have been designed and developed in time to provide more and more effective solutions and newer ways of explaining complex, human-thinking based problem-solving methods. It is also attractive that some widely used Artificial Intelligence techniques such as Fuzzy Logic, Artificial Neural Networks, and Genetic Algorithms are often used by researchers to perform intelligent application and system

based works. As parallel, the Fuzzy Logic technique is the research subject area that this work is based on.

The Fuzzy Logic technique and the Fuzzy Logic control approach in this context are important Artificial Intelligence areas, which are especially used to design and develop intelligent, problem-solving applications or systems. With developments realized in time, many different kinds of applications or software systems that are based on Fuzzy Logic have been introduced. These applications or software systems ensure effective and efficient using features and functions that can be used by computer users for working on both basic and advanced/complex Fuzzy Logic oriented problems or research studies. However, although the related applications or software systems come with different using features and functions, they also include some difficulties on ensuring simple usage enabling computer users to realize their works or studies easily and also understand processes passed during design, development and work of a specific Fuzzy Logic system. On the other hand, anyone who wants to learn basics of Fuzzy Logic and also Fuzzy Logic control encounters with some

* Corresponding author e-mail: utku.kose@usak.edu.tr

difficulties on understanding what exactly happens within the related advanced development environments. Because of the related situations, the authors think that some critical, simple and effective using features and functions allowing computer users to have better using experiences, should be provided on the related applications or software systems. Additionally, they should also employ some attractive and interactive features and functions that allow understanding basics of Fuzzy Logic. As it can be understood, designing and developing such a software system environment, which employs effective and efficient using features and functions to improve the using experience while including advantages of strong applications or software systems, may be an interesting approach to express a solution to the related problems or disadvantages.

In order to provide an alternative approach, the authors had developed an educational, virtual laboratory system, which was introduced before [1]. According to the obtained results within the previous work, the developed educational, virtual laboratory system provided a successful working performance. The obtained results have encouraged authors to improve the related system for obtaining a more effective software system and as result, the second version of the software systems has been designed and developed.

In this paper, the FL-LAB v2: a software system, which can be used to design and develop different kinds of Fuzzy Logic inference and controlling systems, by using an easy-to-use, interactive, and effective software environment, is introduced. As mentioned on the previous paragraph, this software system is the second version of pre-introduced software system and it provides more predefined controls to adjust and realize basic features and functions of a typical Fuzzy Logic inference or control system, without having any foreknowledge about the related subject area. Furthermore, the software system has also been designed on simple but visually improved Windows-form application interfaces. Generally, FL-LAB v2 system is appropriate to be used by most people interested in the Fuzzy Logic and the Fuzzy Logic Control. It ensures an alternative approach to more technical programs based on the related approach/technique.

As suitable to the introduction of the work, the rest of the paper is organized as follows: In the second section, foundations of the realized work are described. In this context, basics of the Fuzzy Logic technique and the Fuzzy Logic Control are explained briefly. Right after, fundamentals of the developed software system: FL-LAB v2 and its important using features and functions are explained in detail in the third section. Following the third section, evaluation processes performed to show effectiveness and adequacy of the FL-LAB v2 are explained in the fourth section. Finally, results of the realized work and also some future works related to the developed software system are discussed within the last section.

2 Foundations

The FL-LAB v2 employs appropriate Fuzzy Logic algorithm and program code structures to overcome the mentioned functional solutions and objectives. In this sense, the whole algorithmic working mechanism of the FL-LAB v2 and its infrastructure regarding to the Fuzzy Logic technique and the Fuzzy Logic Control have been written via C# programming language. At this point, it is important that basics of the Fuzzy Logic technique and the Fuzzy Logic Control should be explained before introducing the software system, in order to enable readers to dominate the subject area/field that is touched upon.

2.1 Basics of the Fuzzy Logic Technique and the Fuzzy Logic Control Approach

The Fuzzy Logic technique is a powerful Artificial Intelligence approach, which was developed based on the Fuzzy Set Theory. The Fuzzy Set Theory was first introduced by L. A. Zadeh in 1965 [2]. Following Zadeh's works within the related subject, more researchers have also introduced and expressed newer approaches, methods, and ideas about the Fuzzy Set Theory [3,4,5,6]. With these studies, The Fuzzy Set Theory has become a remarkable and important approach among many people in the related field. Finally, Fuzzy Logic was developed from the Fuzzy Set Theory to simulate the human-thinking and behaviors and represent knowledge in an operationally powerful form via this way.

At this point, explaining differences between the Crisp Logic and the Fuzzy Logic can enable readers to have better idea about basics of Fuzzy Logic. In this context, Fuzzy Logic is known as an extension of the Crisp Logic. Within the Crisp Logic, variables are defined as only with true (1) or false (0) values. So, a member can be included in only one set. For example; for the relationship with the set X , the characteristic function gives us true (1) if a member named m is the member of the set X or false (0) if the m is not the member of the set X [4].

At this point; as different from Crisp Logic, Fuzzy Logic evaluates a member as included in one or more sets provided. In order to explain this new approach, Fuzzy Logic uses membership degree values to express membership situation of any member. Associated with the membership approach for Fuzzy Logic, the most important subject is using membership functions.

2.1.1 Membership functions in Fuzzy Logic

In the Fuzzy Logic, the mentioned membership degree of a member is expressed via different types of functions called as membership functions.

In a mathematical manner, the relationship between the set X and the member m can be defined as in the Equation 1.

$$\mu_X(m) : F \rightarrow [0, 1]; \quad \mu_X(m) \in [0, 1] \quad (1)$$

Within the Equation 1; the set F is the fuzzy set of $X(m)$ [4].

In the context of the expressions, each membership function can be defined with different curves using different mathematical calculations to return the membership degree of any member (In this way, the main definition of a membership function can also be obtained: A membership function is a typical curve converting the numerical value of input within a range from 0 to 1 [7,8,9,10]).

It is extremely important that a typically developed Fuzzy Logic model/system is based on thinking of the person, who has an expert experience, knowledge, and ideas about the related problem subject. So, membership functions for a designed and developed Fuzzy Logic model/system should be determined with the support and help of an expert person [11]. In the related concept, this is an effective way to introduce the knowledge of experts for an accurate and proper control approach [1,5,12,13,14,15,16].

2.1.2 Using linguistic rules in Fuzzy Logic

Linguistic rules are used to realize working mechanism (evaluating the input variables to obtain output values associated with output variables) of a developed Fuzzy Logic system/model. A typical linguistic rule is formed by using an IF ... (AND/OR) ... THEN rule sentence. In this context, some example rules for a three inputs – one output Fuzzy Logic system/model, can be written as below:

–IF (weight IS LOW) AND (height IS NORMAL)
AND (age IS YOUNG) THEN (result IS HEALTHY)

–IF (weight IS VERY LOW) AND (height IS SHORT)
AND (age IS VERY OLD) THEN (result IS ILL)

–IF (weight IS VERY LOW) OR (height IS VERY
HIGH) AND (age IS OLD) THEN (result IS
CRITICAL)

... etc.

With the help of this approach, different linguistic rules can be formed for a typical Fuzzy Logic system/model. In this way, the expert defines behaviors of the related system for different types of inputs. Furthermore, in a typical Fuzzy Logic system/model, the necessary linguistic rules can also be combined in a table called as the rule table. Each output for different input

variables can be determined and observed via this table. As it can be understood from previous explanations, content of the table is formed with an expert knowledge.

2.1.3 Defuzzification process

After determining the linguistic rules, the developed Fuzzy Logic system/model is ready to work. Briefly, the working mechanism can be expressed as follows: the developed system/model obtains the output value(s) associated with output variable(s) from fed input variable(s) [via input value(s)]. Generally, this process is called as Fuzzification. At this point, the obtained output is defined as in a fuzzy set form. Because of this situation; in order to use the related output value in the real-world, the fuzzy value is translated into real values via different mathematical approaches [15,16]. This translation process is called as Defuzzification [17]. In the literature, there are many different approaches that are used for the Defuzzification process [18,19,20]. Some widely used ones are Centroid, Centre of Sums, and Mean of Maxima.

2.1.4 Determining a Fuzzy Logic model

Typically, the Fuzzy Logic technique is a collection of different mathematical approaches, methods, and calculations. For instance, there are different types of calculation approaches that have been introduced for functions performed within a Fuzzification and Defuzzification process. At this point, the related approaches result different Fuzzy Logic models that are used to obtain results from different mathematical aspects. In this sense, there are four different Fuzzy Logic models used within the related works. In the literature, Mamdani and Takagi-Sugeno type Fuzzy Logic models are widely used for the related works and studies [7,21,22,23].

2.1.5 Fuzzy Logic Control – Fuzzy Control

As a result of combining theoretical infrastructure of Fuzzy Logic and real-world based control problems, Fuzzy Logic Control (Fuzzy Control) systems are designed and developed in parallel. These intelligent control systems allow researchers to solve control problems with more accurate, effective, and efficient logical and mathematical approaches. A typical Fuzzy Logic Control – Fuzzy Control system schema is shown in Fig. 1 [24].

2.1.6 Current innovations in Fuzzy Logic

Today, the Fuzzy Logic technique is widely used for forming hybrid intelligent models aiming to provide more

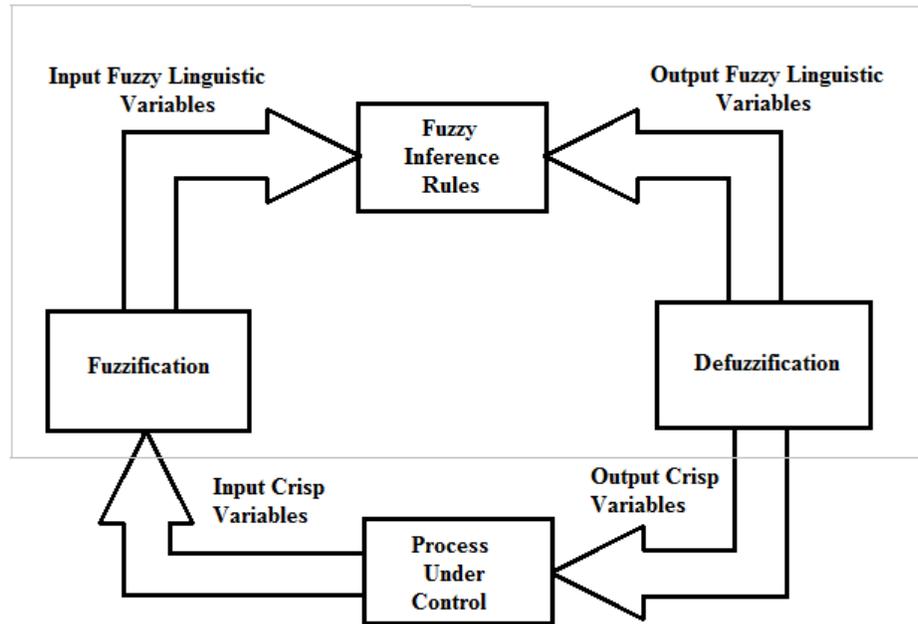


Fig. 1: A typical Fuzzy Logic Control – Fuzzy Control system schema [24].

effective solution approaches. In this sense, especially newer Artificial Intelligence techniques and algorithms attract researchers to design more effective problem-solving methods. But it is also an effective approach to combine different Artificial Intelligence techniques to get hybrid models for problem solving process. Additionally, adjusting different mathematical parameters of an Artificial Intelligence technique by using other Artificial Intelligence techniques is another popular approach applied by researchers (Similarly, the parameters of the Fuzzy Logic technique can also be adjusted with the last mentioned way). As result (according to the expressed innovations), it seems that the improvements within the research field and the related literature will be continued in the future.

3 FL-LAB v2: Design and Development of an Easy-to-Use, Interactive Fuzzy Logic Control Software System

The FL-LAB v2 comes with effective using features and functions that enable computer users to work better on the Fuzzy Logic technique and so Fuzzy Logic Control approach, thanks to an easy-to-use, interactive software system environment. In order to learn more about FL-LAB v2 and its usability, both its programmatic infrastructure and using features and functions should be explained. Before explaining using features and functions of the FL-LAB v2, the programmatic infrastructure will be introduced.

3.1 Programmatic Infrastructure of the FL-LAB v2

The FL-LAB v2 has been designed and developed by using an object oriented programming language called C#. The first development of FL-LAB has been coded on the Microsoft Visual Studio 2008 platform whereas newer additions and also adjustments for the second version: FL-LAB v2 has been done on the Microsoft Visual Studio 2010.

Generally, programmatic infrastructure of the FL-LAB v2 can be examined under two different categories: Interface Designs and Fuzzy Logic Codes.

3.1.1 Interface designs

Interfaces of the FL-LAB v2 has been designed meticulously via Visual Studio 2008 and also 2010 environment, in order to provide effective and efficient enough using experiences for computer users. At this point, special C# programming language components has been employed to realize the related objectives. Button, TextBox, ListBox, Group View, and all of other remaining objects/components have been designed and located on the related software system interfaces carefully. Each interfaces of the related software system is associated with a Windows-form window platform. Briefly, FL-LAB v2 interfaces can be listed as below:

–Main Software System Interface,

- Input – Output Variable Definition Interface,
- Linguistic Rules Definition Interface,
- Rule Table Definition Interface,
- Rule Applicator Interface,
- Control Feeding Interface,
- Incoming Connector Interface,
- Outgoing Connector Interface,
- Active System/Model Info Interface,
- Software System Options Interface,
- Fuzzy Logic Support Web Interface.

Design definitions of the listed FL-LAB v2 interfaces has been done via/under .cs files provided within the software system source code (project solution). The related interfaces and basic using features and functions that can be performed via these interfaces are explained within the next sections.

3.1.2 Fuzzy Logic codes

The most important infrastructure part of the FL-LAB v2 is the programming code structure associated with the category name: Fuzzy Logic Codes. The Fuzzy Logic technique and also Fuzzy Logic Control – Fuzzy Control infrastructure of the FL-LAB v2 has been coded in detail in order to provide a fast and flexible software system (working mechanism). The foremost object oriented programming approaches (via C#) has been used by authors to realize the mentioned objectives. The program code structure of the first version FL-LAB has been written flexible enough for future improvements, so it was easy for authors to improve using features and functions of the FL-LAB in the second development. Still, the programmatic code structure of the FL-LAB v2 is flexible enough for future additions and developments.

Briefly, the Fuzzy Logic codes of the FL-LAB v2 has been written as appropriate for the common working mechanism flow of the Fuzzy Logic technique. In order to improve compiling and working performance, pipeline approaches has been used while writing the Fuzzy Logic codes. As similar to the interfaces, the code definitions of the Fuzzy Logic have been done via/under .cs files. Briefly, the Table 1 shows the related .cs files and their functions.

As being associated with the mentioned programmatic infrastructure, using features or functions of the FL-LAB v2 (on the other words; using FL-LAB v2) is explained as follows:

3.2 Using Features or Functions of the FL-LAB v2 – Using FL-LAB v2

It is possible to examine and explain using features and functions of the FL-LAB v2 under three titles: Designing and Developing Fuzzy Logic Systems/Models, Working on an Active Fuzzy Logic System/Model, Connection

Table 1: Fuzzy Logic Codes – The .cs files and their functions.

The .cs file	Function
FuzzySystem.cs	General functions associated with designing and developing a Fuzzy Logic system-model.
MamdaniFuzzySystem.cs	Design, development and control of Mamdani type Fuzzy Logic systems-models.
TakagiSugenoFuzzySystem.cs	Design, development and control of Takagi Sugeno type Fuzzy Logic systems-models.
FuzzyVariable.cs	Forming and adjustments of Fuzzy Logic input and output variables.
MembershipFunction.cs	Determining appropriate membership functions and the related parameters for defined input or output variable(s).
FuzzyRule.cs	Creation and adjustments of the related linguistic rules for an active system-model.
RuleParser.cs	Parsing linguistic rules during the working process of the active system-model.
DefuzzifyMethod.cs	The related working mechanisms defined for the defuzzification methods.

with Other Applications and Additional Features or Functions. Combination of these titles defines a typical working session that can be performed via FL-LAB v2.

3.2.1 Designing and developing Fuzzy Logic systems/models

After executing the FL-LAB v2 application file, the Main Software System Interface is opened. As it can be understood, this is the main software system window, where the computer user can perform or reach to all necessary processes associated with the Fuzzy Logic technique. The interface comes with simple and fast controls ensuring an interactive, easy-to-use working experience. A screenshot from the Main Software System Interface of the FL-LAB v2 is shown in Fig. 2.

With the controls provided on the Main Software System Interface, it is easy to perform simple File operations (New, Open, Save... etc.), examine and adjust current features (input or output variable(s), linguistic rules, model options, feeding, controlling, making connection... etc.) of the designed Fuzzy Logic system/model. The Main Software System Interface also allows computer users to adjust main using features and

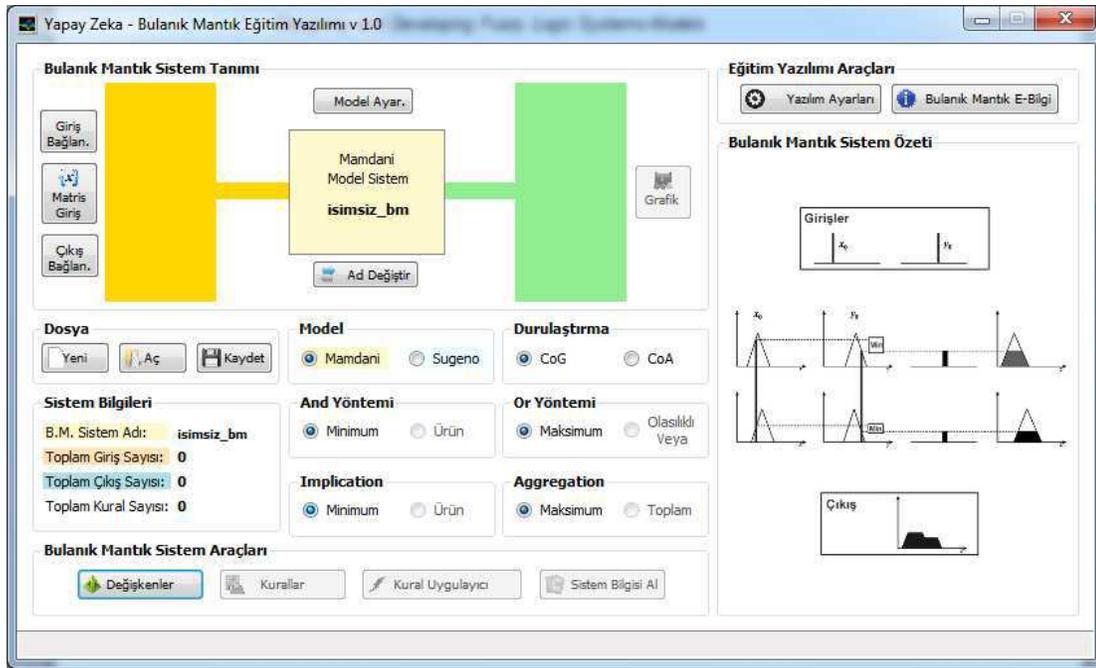


Fig. 2: The Main Software System Interface of the FL-LAB v2.

functions of the FL-LAB v2 (Software System Options Interface) and also reach some additional resources (Fuzzy Logic Support Web Interface).

Generally, each control provided on the Main Software Interface defines a specific mechanism related to the general working mechanism of the FL-LAB v2. At this point, in order to make the paper easier to read and give more emphasis on more important and critical using features and functions, details about some simple controls are not explained. But readers may understand roles of some of these controls while reading next sub-sections.

It is too easy to design and develop a new Fuzzy Logic system/model via FL-LAB v2 controls. Firstly, a new system/model creating command can be given to the software system with the provided File controls. After this, other remaining design and development processes are related to typical Fuzzy Logic system/model forming operations, which were explained indirectly under the Foundations section.

For defining input – output variable(s), memberships functions of the related variables and linguistic rules of a system/model and also realize the related operations, it is possible to use the related controls included within two group panel controls called as the Fuzzy Logic Software System Tools and the Fuzzy Logic System/Model Definitions. The Fuzzy Logic System/Model Definitions panel can also be used for displaying some information (active system/model name, model type, defined variables... etc.) about the active system/model, thanks to some interactive and visual tools. On the other hand, it is also possible to perform other specific operations like

realizing controlling process or ensuring connection with other applications, via the Fuzzy Logic System/Model Definitions panel. The Fig. 3 shows a regional screenshot from the Fuzzy Logic System/Model Definitions panel.

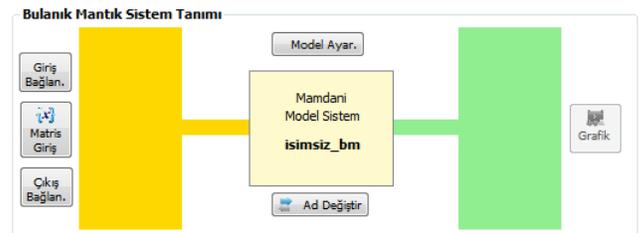


Fig. 3: The Fuzzy Logic System/Model Definitions panel.

On the Main Software System Interface, the button named Variables can be used to open the Input – Output Variable Definition Interface, where the related input – output variable(s) and their membership functions can be defined, adjusted or examined. Fig. 4 shows a screenshot from the Input – Output Variable Definition Interface.

Within the interface, there are two different tab controls that can be used for displaying the related controls for defining, adjusting or examining input – output variable(s). Briefly, the computer user can organize variables and related membership functions of any variable, thanks to the provided controls. Additionally, it

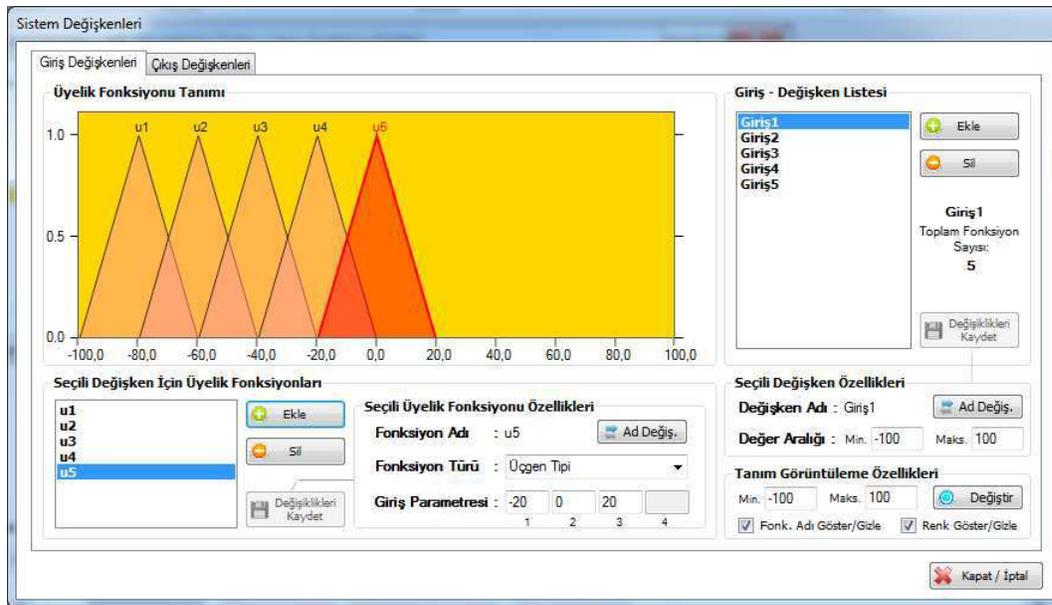


Fig. 4: The Input – Output Variable Definition Interface.

is also possible to adjust value range of each variable and parameters or type of each membership function via the related controls. While examining membership functions of a variable, their visual definitions are shown automatically. At this point, each different type membership function is displayed as in different visual shapes. Names of the related membership functions and value range of the variable are also displayed with these visual tools. After choosing a membership function for displaying it or adjusting its features, its visual shape is also displayed in red color. Visual shapes of some important membership functions in FL-LAB v2 are represented in Fig. 5.

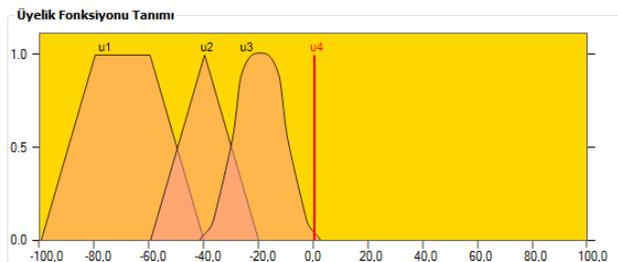


Fig. 5: Visual shapes of some important membership functions in FL-LAB v2.

The FL-LAB v2 allows defining many different types of membership functions. Additionally, the FL-LAB v2 enables the computer user to define a membership

function manually or automatically. In this sense, manual definition can be used to define all necessary parameters of the membership function whereas automatic definition creates a new membership function in which the necessary parameters are adjusted in a suitable, symmetric parameters flow. Of course, it is also possible to change the automatically defined parameters manually.

Following the definition of the related variable(s) and membership function structures, the linguistic rules of the active system/model can be defined by using the Linguistic Rules Definition Interface. This interface can be opened via the Linguistic Rules button provided on the Main Software System Interface. The mentioned linguistic rule definitions can be made easily by using simple controls displayed on the Linguistic Rules Definition Interface. Furthermore, the FL-LAB v2 also allows computer users to define the linguistic rules automatically by using the crossing approach. This approach can also be called as the symmetric rule matching approach for the systems/models having two inputs and one output variables. Fig. 6 shows a screenshot from the Linguistic Rule Definition Interface.

As being associated with defining linguistic rules, it is also possible to define simple rules by using the rule table tool provided on the Rule Table Definition Interface. Briefly, it is possible to define simple linguistic rules by clicking on the related controls on the related rule table shape. At this point, the FL-LAB v2 automatically creates a 2D rule table or some kind of 3D (cubic) rule table structure according to numbers of variables defined. Fig. 7 represents a screenshot from the Rule Table Definition Interface.

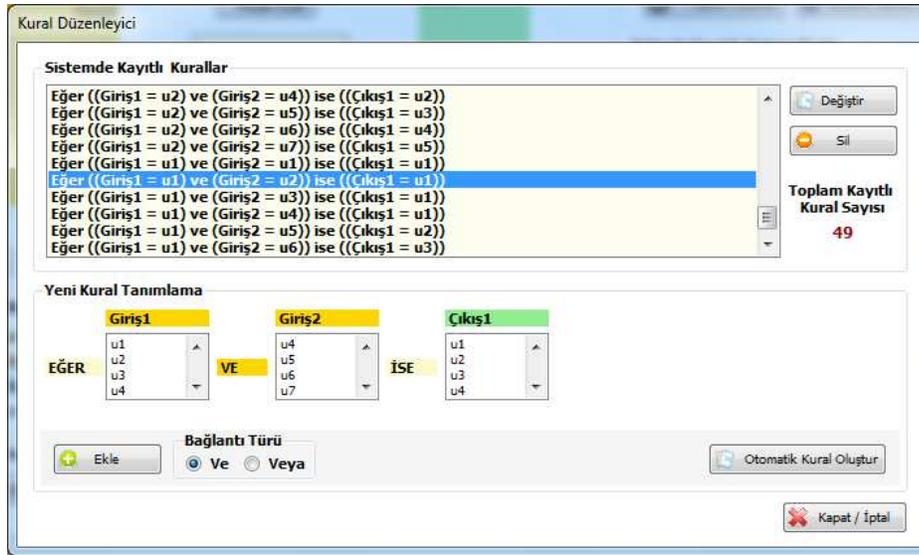


Fig. 6: The Linguistic Rules Definition Interface.

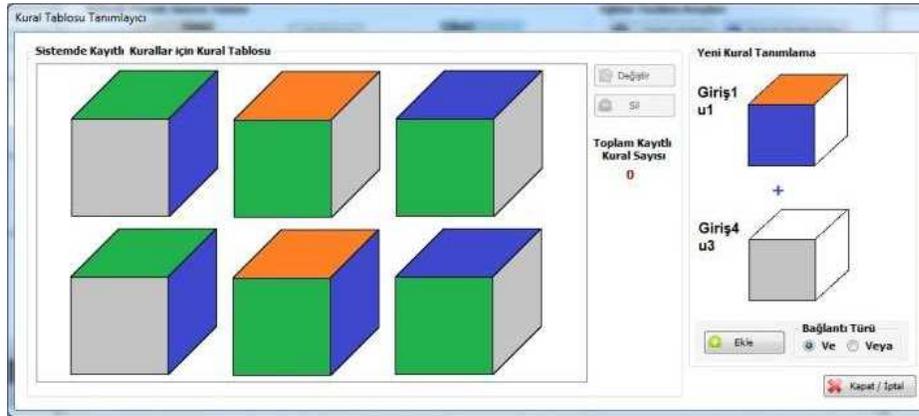


Fig. 7: The Rule Table Definition Interface.

3.2.2 Working on an active Fuzzy Logic system/model

Generally, by performing the related operations explained under the previous sub-section, a typical Fuzzy Logic system/model is ready to be used. At this point, it is possible to use the Rule Applicator Interface for feeding the active system/model with different values to see results (output values of the system/model). In order to achieve this, the related interface can be opened by using the Rule Applicator button on the Main Software System Interface (This using feature/function is similar to the related feature/function provided within MATLAB – Fuzzy Logic Toolbox). On the related interface, the computer user can not-only feed the active system/model but also obtain a wide list of all possible output values by clicking on the provided Output List button. In this way, it is possible to use the obtained values on different

platforms to evaluate them from different perspectives. Fig. 8 shows a screenshot from the Rule Applicator Interface.

An important feature/function of the FL-LAB v2 is based on ensuring a typical Fuzzy Logic Control (Fuzzy Control) process. In this sense, the FL-LAB v2 provides the Control Feeding Interface for intelligent control operations. It is possible to reach to the mentioned interface by clicking on the Matrix Input button on the Main Software System Interface. As it can be understood; in the context of the related feature/function, the FL-LAB v2 allows computer users to define controlling parameters via state-space analysis methods. The related controls for defining parameters are automatically created by the system according to the structure of the active Fuzzy Logic system/model.

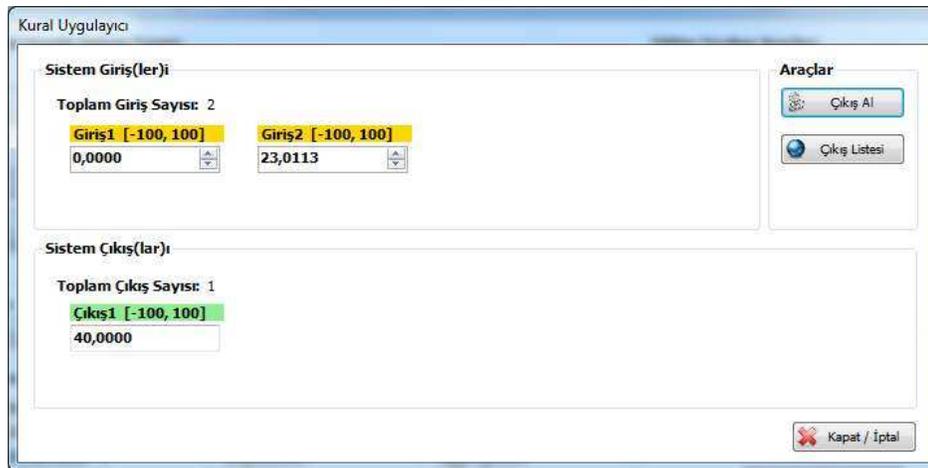


Fig. 8: The Rule Applicator Interface.

For instance, the FL-LAB v2 can allow defining the control solution in the $A.x + B.u$ equation form (the simplest form). In this sense, state-space equation of this system/model can be defined by using the following expressions:

$$\dot{x} = A.x + B.u \quad y = C^T .x \quad \text{and} \quad \dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} \quad (2)$$

Fig. 9 shows a screenshot from the Control Feeding Interface in which the related controls are automatically created as being suitable to the related solution expressed within Equation 2 (As it can be seen from the Fig. 9, the computer users can define A , B , C , and x matrixes, u value, reference value, and the error limit via this interface, before starting to the control process).

Following the definitions for all parameters, control process for the active system/model can be started by clicking on the Start Control Process button on the interface. The process may be continued until the controlling limits (objective or ending point) are reached. At the end of the control process, graphics for the error value, y value, u value, and the system/model output (du value) are automatically viewed by the FL-LAB v2. Fig. 10 represents screenshots from the related graphics viewed after a control process (94 iterations).

3.2.3 Connection with other applications

One of the most attractive, effective, and important using feature/function of the FL-LAB v2 is ensuring a connection environment. At this point, the FL-LAB v2 allows computer users to form hybrid systems with other Artificial Intelligence techniques by using incoming or outgoing connection type of the software system. This using feature/function is the biggest improvement for the

FL-LAB software system after the first development. Because of this development, the code structure of the FL-LAB has been revised and rewritten for ensuring the related approaches.

The incoming and outgoing connection approach of the FL-LAB v2 is realized by using a special parameters communication and connection library, which was designed and developed by the authors. The library currently includes many C# programming language-based codes, so it is better to explain general structure and working mechanism of the library, in order to provide more idea about the related using feature/function.

In the context of incoming connections, the FL-LAB v2 reads hierarchically or sequent organized file types. In this sense, it can read for example XML or Excel type (.xls, .csv... etc.) files. Provided data within these files are automatically directed into the core of the communication (connection) library. The related data then evaluated by the library as some kind of parameters and according to the work flow, the active Fuzzy Logic system/model is used immediately. Of course, the related data must be organized as the form of the mentioned files. Most of Fuzzy Logic applications or software systems within the literature allow computer users to obtain these kinds of files. On the other hand, it is also possible to convert any other file types used by the related applications or software systems to the supported file types, thanks to the file converter applications.

The incoming connection operation of the FL-LAB v2 is realized by using the Incoming Connector Interface, which can be opened by using the Incoming Connection button provided on the Main Software System Interface. Fig. 11 represents a screenshot from the related interface.

With the help of the Incoming Connector Interface, incoming data is automatically read by the FL-LAB v2 and fed into the active system/model. As it was mentioned before, different types of hybrid systems

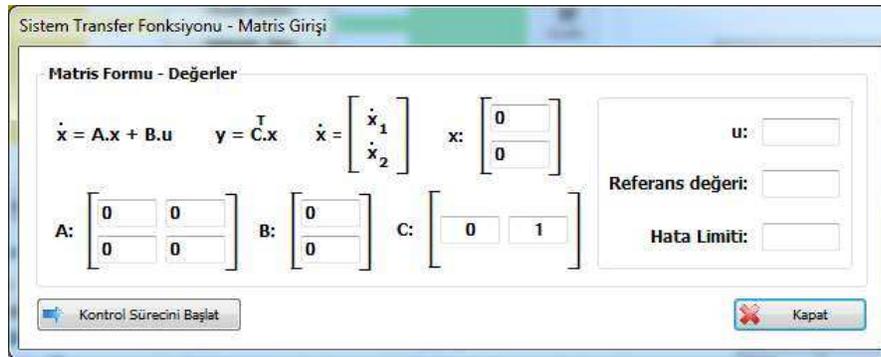


Fig. 9: The Control Feeding Interface for the solution within Equation 2.

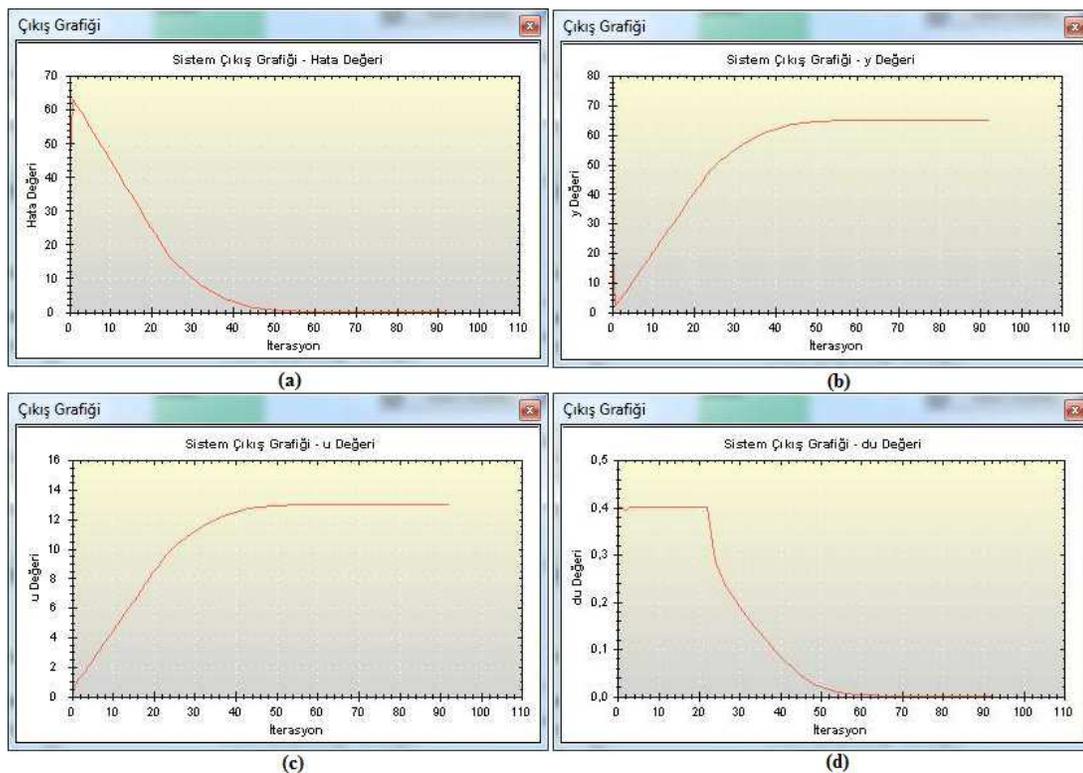


Fig. 10: Screenshots from the graphics viewed after a control process (94 iterations), a) error value, b) Output of system y value, c) The total system/model output u value, d) The change of system/model output du value.

including the Fuzzy Logic technique or the Fuzzy Logic Control approach are formed easily. For example, it is possible for the computer user to form an ANFIS (Adaptive Neuro – Fuzzy Inference System) by using the related using feature/function. In order to provide an example solution, the authors have formed an ANFIS by using the FL-LAB v2 and an Artificial Neural Networks tool, which was designed and developed before. The related Artificial Neural Networks tool was designed and

developed by the authors and introduced in [25]. Fig. 12 shows a screenshot from the related tool.

The other connection interface; Outgoing Connector Interface is used to obtain the related Fuzzy Logic controller outputs to feed other Artificial Intelligence techniques. At this point, the FL-LAB v2 exports a file containing the obtained output values for iterations during the performed control process. Then, the file [XML or Excel-type (.xls, .csv...etc.)] can be used to feed the other application or software system associated with an

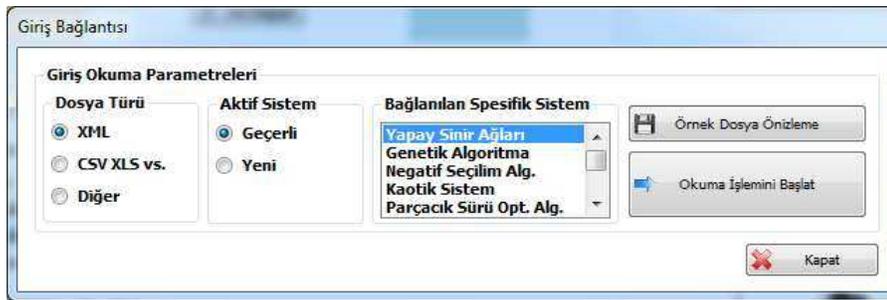


Fig. 11: The Incoming Connector Interface.

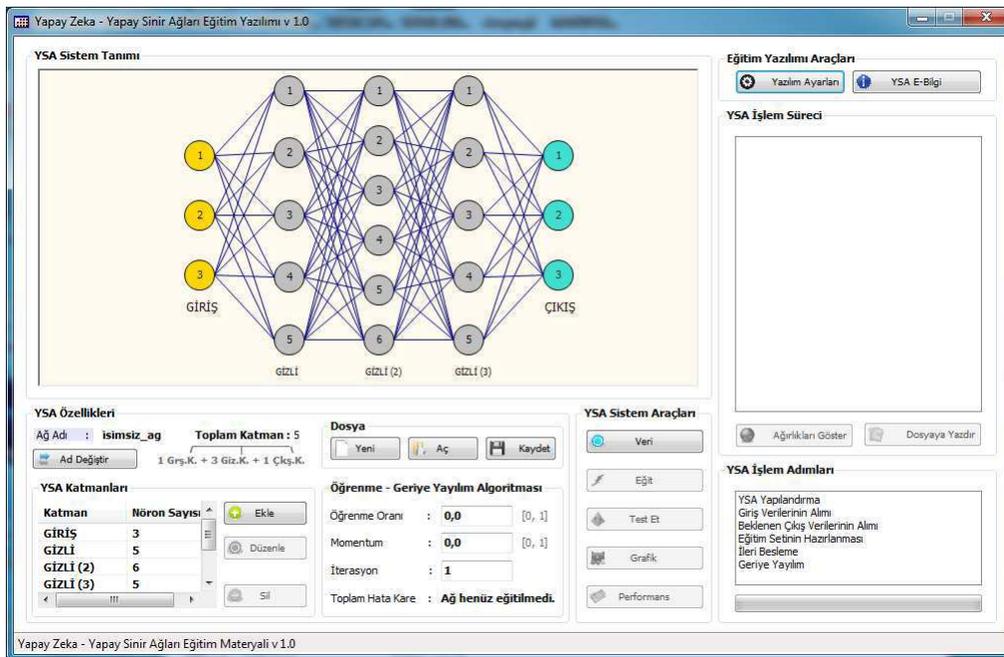


Fig. 12: The Artificial Neural Networks tool used for forming a hybrid system with the FL-LAB v2.

Artificial Intelligence technique. It is also important that programmatically, the output values can be easily read from the file and thus, any Artificial Intelligence technique based application or software system supporting data import can be easily adapted to form a typical hybrid system.

As similar to the other connection type, the outgoing connection operation of the FL-LAB v2 is also done via some simple controls. At this point, the Outgoing Connector Interface (which is opened by using the Outgoing Connection button on the Main Software System Interface), can be used fast and easily, thanks to the provided controls. Fig. 13 represents a screenshot from the related interface.

In addition to the explained using features and functions of the FL-LAB v2, some other additional ones

should also be explained briefly to finalize a complete introduction of the developed software system.

3.2.4 Additional features and functions

Apart from the explained using features and functions, the FL-LAB v2 also comes with some additional features and functions that are related to the reporting, software system settings or supportive interfaces. For example, it is possible to obtain a list of detailed information about the active Fuzzy Logic system/model. This can be done by using the Get System/Model Info button provided on the Main Software System Interface. After clicking on the related button, the Active System/Model Info Interface is opened. It is possible to save this list as a file via provided controls.

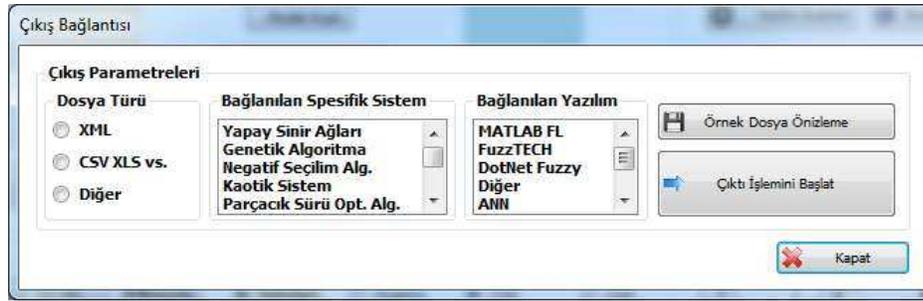


Fig. 13: The Outgoing Connector Interface.

Like most of all software systems, the FL-LAB v2 also provides a settings interface, where the computer user can examine and adjust some visual or functional using features or functions of the software system. The related interface is called as the Software System Options Interface and can be opened by using the Software System Options button on the Main Software System Interface.

Another additional feature/function of the FL-LAB v2 is some kind of reference source. On the other words, it is a HTML-based Web document that can be displayed to get some information about basics of the Fuzzy Logic technique and the Fuzzy Logic Control approach. In this sense, it is possible to use this document as a supportive reference source to the operations performed on the FL-LAB v2. The Web document, which is also called as the Fuzzy Logic Support Web Interface can be displayed by using the Fuzzy Logic Support button provided on the Main Software System Interface.

After explaining the related using features and functions of the FL-LAB v2, its effectiveness and adequacy should be tested in order to have idea about its contributions to the literature and enable readers to evaluate value of this scientific research work better. The following section is dedicated the evaluation process.

4 Evaluation

Within the evaluation process, effectiveness of the FL-LAB v2 and also its calculation abilities has been tried to be evaluated. To this end, Fuzzy Logic systems/models regarding to two different problems have been designed separately and the related solution-oriented works have been performed on these systems/models.

Firstly, a basic Fuzzy Logic system/model has been designed and the results obtained via FL-LAB v2 have been compared with the ones provided by some applications or software systems within the literature. This application and the related process are explained briefly as follows:

4.1 An Application with FL-LAB v2

The application performed on the FL-LAB v2 is some kind of 2 inputs – 1 output Fuzzy Logic system/model. On this model, a problem on analyzing the service quality and food attribute for obtaining a total service success level output has been tried to be solved. A brief system/model schema of the problem is shown in Fig. 14.



Fig. 14: A brief system/model schema of the Fuzzy Logic evaluation problem.

Some specific features for the designed Fuzzy Logic system/model are also explained as below:

- The Mamdani model has been used for designing the Fuzzy Logic problem structure.
- Two input variables (the service quality and food attribute) have been formed within a range of [1...10] whereas the output variable (total service success level) has been formed within a range of [1...30].
- For the service quality input variable, a total of three triangle membership functions (named as weak, good, and excellent respectively) have been used. Parameters of these membership functions has been set as [-5, 0, 5], [0, 5, 10], and [5, 10, 15] respectively.
- For the food attribute input variable, a total of three trapezoid membership functions (named as bad, normal, and delicious respectively) have been used. Parameters of these membership functions have been

set as [0, 0, 1, 3], [1, 4, 6, 9], and [7, 9, 10, 10] respectively.

–For the total service success level output variable, a total of three triangle membership functions (named as cheap, normal, and expensive respectively) have been used. Parameters of these membership functions have been set as [0, 5, 10], [10, 15, 20], and [20, 25, 30] respectively.

In addition to the mentioned features, the linguistic rules written for the problem structure are represented in Table 2.

Table 2: The linguistic rules written for the Fuzzy Logic application problem.

No	The Linguistic Rule
1	IF (the service quality IS EXCELLENT) AND (food attribute IS BAD) THEN (total service success level IS NORMAL)
2	IF (the service quality IS EXCELLENT) AND (food attribute IS DELICIOUS) THEN (total service success level IS EXPENSIVE)
3	IF (the service quality IS EXCELLENT) AND (food attribute IS NORMAL) THEN (total service success level IS EXPENSIVE)
4	IF (the service quality IS GOOD) AND (food attribute IS BAD) THEN (total service success level IS CHEAP)
5	IF (the service quality IS GOOD) AND (food attribute IS DELICIOUS) THEN (total service success level IS NORMAL)
6	IF (the service quality IS GOOD) AND (food attribute IS NORMAL) THEN (total service success level IS EXPENSIVE)
7	IF (the service quality IS WEAK) AND (food attribute IS BAD) THEN (total service success level IS CHEAP)
8	IF (the service quality IS WEAK) AND (food attribute IS DELICIOUS) THEN (total service success level IS CHEAP)
9	IF (the service quality IS WEAK) AND (food attribute IS NORMAL) THEN (total service success level IS NORMAL)

After all of the mentioned design and development steps, the related interfaces of the FL-LAB v2 are used to feed the system/model and get the output responses (values). At this point, the same problem system/model has also been designed with some other Fuzzy Logic applications or software systems within the literature to compare the responses for checking calculation abilities and the accurateness of the FL-LAB v2. In this sense, MATLAB – Fuzzy Logic Toolbox, Fuzzy TECH, jFuzzy Logic, and Dot Fuzzy are used for the comparison procedure.

Table 3 shows some responses obtained via FL-LAB v2, MATLAB – Fuzzy Logic Toolbox, jFuzzy Logic, and Dot Fuzzy; according to the related input values.

As shown in Table 3, the FL-LAB v2 provides accurate and precision responses for the given input values. Generally, it seems that the MATLAB – Fuzzy Logic Toolbox tries to round the obtained values whereas other remaining applications or software systems gives approximate responses.

It can be expressed that the FL-LAB v2 is an efficient and effective enough software system platform for working on the Fuzzy Logic technique and Fuzzy Logic Control approach. On the other words; in addition to the easy-to-use, interactive using features and functions, the FL-LAB v2 also employs a adequacy programmatic infrastructure that allows performing fast, effective, accurate, and efficient scientific research works.

It is also a good idea to provide a brief comparison indicating some improved aspects of the FL-LAB v2, after the first version of the related software system. In this way, readers will be able to understand more about the performed works so far. Table 4 represents a list of improvements associated with the mentioned comparison aspects.

5 Conclusions and Future Work

The FL-LAB v2 is an easy-to-use, effective, and efficient software system, which allows designing and developing Fuzzy Logic based system/models and working on them via simple but strong enough using features and functions provided on a comprehensive and generic programming infrastructure. With the support of provided using features and functions, the FL-LAB v2 enables computer users to work on a fast software system environment. The FL-LAB v2 comes with visual and interactive controls and in this way, it improve the performance during a typical working session related to the Fuzzy Logic technique or the Fuzzy Logic Control (Fuzzy Control) approach.

As different from the current applications or software systems within the literature, the FL-LAB v2 allows computer users to track performed processes and understand the mathematical and logical working flow of the related Artificial Intelligence technique. On the other words, the FL-LAB v2 removes unnecessary technical, complex details or using features/functions to improve using experiences. Furthermore, the FL-LAB v2 also includes some predefined features/functions enabling computer users to design and develop simple or advanced Fuzzy Logic systems/models without having too much foreknowledge about the related subject area. Via these features/functions, the efficiency of the designed and developed software system is improved. Additionally, another critical using feature/function of the FL-LAB v2 is the incoming and outgoing connection infrastructure allowing forming hybrid systems (for instance; ANFIS, Genetic Fuzzy systems...etc.) supported with the Fuzzy

Table 3: Comparison of responses obtained according to the related input values.

Input 1: the service quality	Input 2: food attribute	Output: total service success – for;				
		FL-LAB v2	MATLAB – FL Toolbox	Fuzzy TECH	jFuzzy Logic	Dot Fuzzy
5,0000	8,0000	19,2551	19,3000	19,2661	19,2800	19,2550
2,0000	8,0000	13,9978	14,0000	13,9990	14,0020	14,1010
4,6000	3,2000	23,5806	23,6000	23,5900	23,6005	23,5888
9,6540	1,4560	16,0781	16,1000	16,0500	16,0027	16,0754
6,6700	3,5670	25,0010	25,0000	25,0008	25,0051	25,0009

Table 4: FL-LAB v2 improvements associated with the mentioned comparison aspects.

Comparison aspect	Improvement(s)
Runtime	Runtime of the FL-LAB v2 is better (approximately 20%) than the first version. This improvement is associated with revisions done on especially programmatic infrastructure.
Usability	Usability of the FL-LAB v2 is better than the first version. After adding some more visual and interactive interfaces, usability of the software system has been improved. Especially, the feature on connection with other application has provided many advantages in terms of usability.
Technical infrastructure	Technical infrastructure of the FL-LAB v2 is better than the first version. While revising the program codes of Fuzzy Logic, the authors have added some more features/functions regarding to working on more advanced system-models in a wide spectrum.
Programmatic infrastructure	FL-LAB v2 comes with a revised and up-to-date programmatic infrastructure including different programming methods to improve general performance.
Educational aims	Educational aspects of the first version are still included in the FL-LAB v2. The authors have not added more features/functions providing educational works. But improvements in technical infrastructure and usability will enable FL-LAB v2 to be used more during educational activities related to more advanced Fuzzy Logic-based works/studies.

Logic technique. It is easy for computer users/researchers to ensure connection among the FL-LAB v2 and other Artificial Intelligence techniques/algorithms via only simple file structures. As it can be understood, the FL-LAB v2 is an effective and efficient software system environment, which allows realizing research works in

the context of wide spectrum including simple and more advanced systems/models.

It is also remarkable that the FL-LAB v2 is also an advanced software system enabling computer users/researchers to form typical Fuzzy Logic Control (Fuzzy Control) processes. In this sense, the FL-LAB v2 provides fast and effective interfaces and also controls for intelligent control operations. In the context of the related using feature/function, the FL-LAB v2 allows defining control parameters with state-space analysis methods. The related controls for defining parameters are automatically created by the system according to the structure of active Fuzzy Logic systems/models. All of the introduced using features and functions point a general, effective software system environment aiming improving the current literature. In this sense the FL-LAB v2 is also evaluated as an alternative for the related applications or software systems that can be encountered within the literature.

As it can be understood, positive feedbacks for new improvements on the software system encourage the authors to work for newer developments. In this context, it is possible to express some future ideas on the FL-LAB v2. Regarding the future works for the FL-LAB v2, there are currently some research works to add more different types of membership functions (for example, some derivatives of the Gauss Bell membership function or enabling computer users/researchers to form custom membership functions... etc.) to future developments of the software system. Additionally, more adjustment controls for provided Fuzzy Logic systems/models will also be included within newer versions. Also, there will be also some works on designing and developing newer versions of the FL-LAB on the WPF programming platform. Thus, it will be possible to use custom controls improving interactivity, ensuring more easy-to-use features (easy using experiences), and also forming visually advanced interfaces.

References

- [1] U. Kose and O. Deperlioglu, An educational, virtual laboratory system for fuzzy logic. In Proceedings of the International Symposium on Computing in Science and Engineering, 3-5 June. Gediz University (2010).

- [2] L.A Zadeh, Fuzzy sets. *Information and Control* **8**:338-353 (1965).
- [3] P. Mahonen and T. Frantti, Fuzzy classifier for star-galaxy separation. *The Astrophysical Journal* **541**:261-263 (2000).
- [4] N.K Kasabov, *Foundations of neural networks, fuzzy systems, and knowledge*. MIT Press, Cambridge (1998).
- [5] N. Yalcin and U. Kose, A web based education system for teaching and learning fuzzy logic. In *Proceedings of the International Computer and Instructional Technologies Symposium*. 7-9 October. Karadeniz Technical University (2009).
- [6] U. Kose, Fundamentals of fuzzy logic with an easy-to-use, interactive fuzzy control application. *International Journal of Modern Engineering Research* **2**: 1198-1203 (2012).
- [7] C.C Lee, Fuzzy logic in control systems: Fuzzy logic controller-Part I/Part II. *IEEE Transactions on Systems, Man, and Cybernetics* **20**:404-435 (1990).
- [8] B. Kosko, *Fuzzy engineering*. Prentice-Hall, New Jersey (1997).
- [9] A. Majumdar and A. Ghosh, Yarn strength modeling using fuzzy expert system. *Journal of Engineered Fibers and Fabrics* **3**: 61-68 (2008).
- [10] V.V Nabiyevev, Yapay zeka (problemler-yontemler-algoritmalar) / Artificial Intelligence (problems-methods-algorithms). Seckin Press, Ankara (2010).
- [11] S. Kutuva, N.P Reddy, Y. Xiao, X. Gao, S.I Hariharan, S. Kulkarni, A novel and fast virtual surgical system using fuzzy logic. In *Proceedings of the IADIS Virtual Multi Conference on Computer Science and Information Systems*. 15-19 May, IADIS (2006).
- [12] A. Piegat, *Fuzzy modeling and control*. Physica-Verlag, Heidelberg (2001).
- [13] R.R Yager and D. Filev, *Essentials of fuzzy modeling and control*. Wiley, New York (1994).
- [14] C. Von Altrock, *Fuzzy logic*. R. Oldenburg Verlag, Munich (1995).
- [15] O. Deperlioglu, Bir anahtarlama kipli konvertorun sinirsel – bulanik denetimi / Neuro – fuzzy control of a switch mode converter. *Doctoral Dissertation Thesis*, Ankara Gazi University (2001).
- [16] C. Elmas, *Bulanik mantik denetleyiciler / Fuzzy logic controllers*. Seckin Press, Ankara (2003).
- [17] P. Dadone, *Design optimization of fuzzy logic systems*. *Doctoral Dissertation Thesis*, Virginia Polytechnic Institute and State University (2001).
- [18] R. Zhao and R. Govind, Defuzzification of fuzzy intervals. *Fuzzy Sets and Systems* **43**:45-55 (1991).
- [19] K.M Passino and S. Yurkovich, *Fuzzy control*. Addison-Wesley, Massachusetts (1997).
- [20] J. Jantzen, *Design of fuzzy controllers*. Technical Report (No: 98-E 864). Technical University of Denmark (1998) <http://faculty.petra.ac.id/resmana/private/fuzzy/design.pdf> Accessed 25 June 2012.
- [21] J. Yan, M. Ryan, J. Power, *Using fuzzy logic: Towards intelligent systems*. Prentice-Hall, Cambridge (1994).
- [22] R. Fuller, On fuzzy reasoning schemes. In: *The state of the art of information systems in 2007*. TUCS Publication, Turku, pp 85-112 (1999).
- [23] M.J Wierman, *Fuzzy sets, fuzzy logic, and control*. M.J Wierman, Nebraska (2008).
- [24] P. Shahmaleki, M. Mahzoon, A. Kazemi, M. Basiri, Vision-based hierarchical fuzzy controller and real time results for a wheeled autonomous robot. In *Tech (2010)* <http://www.intechopen.com/books/motion-control/vision-based-hierarchical-fuzzy-controller-and-real-time-results-for-a-wheeled-autonomous-robot> Accessed 26 June 2012.
- [25] O. Deperlioglu and U. Kose, An educational tool for artificial neural networks. *Computers and Electrical Engineering* **37**: 392-402 (2011).



Utku Kose received the B.S. degree in 2008 from computer education of Gazi University, Turkey as a faculty valedictorian. He received M.S. degree in 2010 from Afyon Kocatepe University, Turkey and now, he continues D.S. / Ph. D. at Selcuk University, Turkey in

the field of computer engineering. Between 2009 and 2011, he has worked as a Research Assistant in Afyon Kocatepe University. Following, he has also worked as a Vocational School – Vice Director and a Lecturer in Afyon Kocatepe University between 2011 and 2012. Currently, he is a Lecturer in Usak University, Turkey. His research interest includes artificial intelligence, the chaos theory, distance education, computer education, online / virtual laboratories, and cryptography.



Omer Deperlioglu received the B.S. degree in 1988 from the electrical education of Gazi University, in Turkey. He received M.S. degree in 1996 from Afyon Kocatepe University, Turkey and he completed Ph. D. degree in 2001 at Gazi University in field of

controlling switch-mode dc – dc converters with neuro – fuzzy system. He is currently an Associate Professor in Afyon Kocatepe University, Turkey. His research interest includes computer-based control systems, fuzzy logic control, neuro – fuzzy control, and distance learning.