IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 15, 2020, accepted November 1, 2020, date of publication November 16, 2020, date of current version November 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037901

Hierarchical Fuzzy Logic for Multi-Input Multi-Output Systems

SHASHANK KAMTHAN, (Member, IEEE), AND HARPREET SINGH, (Life Senior Member, IEEE)
Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA
Corresponding author: Shashank Kamthan (kamthan.shashank@gmail.com)

ABSTRACT Fuzzy logic has created a high impact on research and development in almost all engineering applications. Recently, there has been an increasing interest in various offshoots of fuzzy logic approach and hierarchical fuzzy logic is one such area of research development and applications. With the increase in volume of data, hierarchical fuzzy logic has emerged as a highly suitable candidate for research. The objective of this paper is to develop methodology for multi-input multi-output hierarchical fuzzy systems. In particular, a system to be designed is broken into a number of sub-subsystems, where each subsystem is designed separately and then connected in hierarchical structure. The strategy used in this paper is to avoid the repetition of common terms across different subsystems of multi-input multi-output systems. This strategy has not been presented hitherto by any other author. This paper first discusses in detail the implementation of a multi-input single-output hierarchical system. It then extends the approach to multi-input multi-output hierarchical systems.

INDEX TERMS Fuzzy hierarchical system, fuzzy logic, fuzzy systems, hierarchical fuzzy systems, hierarchical systems, intelligent system, hierarchical MIMO systems.

I. INTRODUCTION

In the year 1965, Lotfi Zadeh first introduced the term ''Fuzzy Logic'' in his research paper on fuzzy sets [1]. Fuzzy logic is an effective means to resolve conflicts and provide realistic assessments because of its ability to deal with information that is uncertain, imprecise, or vague etc. Fuzzy logic involves use of linguistic variables, which serve to construct better mathematical and realistic models. Since the advent of classic papers by Lotfi Zadeh [1], fuzzy logic has been used for a large number of applications in different disciplines and used in real life applications [6]. A brief review has been given in this paper with a view to familiarize the concepts and terminology of fuzzy methodology, so that the reader acquires a foundation of fuzzy logic and hierarchical fuzzy systems.

Fuzzy logic originated from the generalization of classical logic [1] that incorporates the smooth transition between true and false. It is the structure and formation of multiple-values logic in which there are more than one truth values in range i.e., instead of representing output as 0 or 1, true or false etc.; it results as a degree of truth including both 0 and 1. Fuzzy logic accompanied the concept of partial truth, where values fluctuate between complete true or false. The fuzzy logic [1] comprises fuzzy set theory where set A in universe X is characterized by membership functions that represent real numbers for every element in X.

There is a huge limitation of standard logic that only allows results in the form of either 0 or 1; however, one may find varied conclusions when asking questions such as skin color etc. In such cases, degree of truth appears as the outcome of logical reasoning. At first instance, both probabilistic approach and degree of truth might be seen similar but on the contrary, degree of truth uses a mathematical model of uncertainty and probabilistic approach uses mathematical model of ignorance.

Let us take a real-life example of height i.e., assume height is classified into two categories: short and tall, where short height is defined in range [50, 70] inches; and tall height is defined in range [65, 85] inches. A normal person could be included in both categories, as fuzzy logic allows the crisp values to be a subset of two logical sets, which gives fuzzy logic an edge to be more intuitive and closer to the way people think. Linguistically, a person is not always short or tall exclusively. It can be represented as ''partially tall'' or ''little short'' etc.

In the modern era with complex systems and environment, fuzzy logic has been successfully used by many researchers

The associate editor coordinating the review of this manuscript and approving it for publication was Min Wang.

such as engineering, technology, scientists, mathematicians, analysts etc., and used in countless applications [2], [3] in areas not limited to automobiles, defense, security, medicine, biomedical, internet, power industry, consumer electronics, weather forecasting, law, business etc. A few examples of real-life applications are defined below.

One of the applications of fuzzy logic has been manifested in Japan by Sendai Subway Control System. Using fuzzy logic, Hitachi developed Nanboku line to control the train transportation system that results in one of the smoothest running systems in the world with enhanced efficiency and improved performance.

In commercial appliances, fuzzy logic has already appeared for applications such as ventilation, air conditioning, HVAC systems etc. to control the thermostats for heating, cooling, energy efficient systems. The outcome of fuzzy logic maintains steady temperature, compared to traditional thermostats in the market.

In the field of 3D animation, fuzzy logic application includes artificial intelligence-based animation for generating crowds. This application has been extensively used while making world's best movies such as The Lord of Rings, The Lion, Avatar, The Wardrobe Films etc.

Other applications [6] include analysis of power systems and identify any harmonic disturbances, where a fuzzy logic system analyzes the fundamental parameters such as voltage and current, as well as temperature to determine root cause of any failure.

A brief review of fuzzy and neuro-fuzzy systems is given below first for completeness.

A. FUZZY SYSTEMS

Fuzzy systems mainly work on the interpretability [1], [10] that is abstract from linguistic approach and understandable to humans. However, fuzzy systems extracted from experimental data may not necessarily comprehend human language and to increase the performance, different learning methods in fuzzy logic lead to loss of human interpretation. One of the major motivations to implement a fuzzy model is its transparency to understand human interpretation.

The fuzzy system represents a nonlinear system which comprises three steps: Fuzzification, Inference systems and Defuzzification.

Fuzzification converts inputs into suitable linguistic values that can be viewed as labels for fuzzy sets. Similarly, inference systems have three parts: Rule base contains selection of input-output relations; Data set comprises membership functions for rules; Normalization of input and output combination that reduces the redundancy of rules. The defuzzification combines the output from the inference system and converts it into desired forms.

B. NEURO FUZZY SYSTEMS

Neuro-fuzzy [5], [10] is a blend of neural network and fuzzy logic. The hybridization of these two technologies i.e., by adding linguistic reasoning style with fuzzy systems and cognitive science connection with neural networks, result in an intelligent system. In literature, this hybridization of fuzzy and neural networks commonly referred to as FNN (fuzzy neural network) or NFS (neural-fuzzy system).

A neuro-fuzzy system embraces a model in sets with fuzzy systems, where a model consists of a linguistic set with rules defined with IF-THEN statements. The main advantage of neuro-fuzzy systems is its ability to act as a feed-forward network, commonly known as universal approximation and interpret IF-THEN rules.

Both Mamdani and Sugeno models [10] have given a boost to the work suggested by Lotfi Zadeh [1]. The modeling of neuro-fuzzy systems is mainly focused on interpretability and accuracy. The Mamdani model has been used to define linguistic fuzzy modeling for interpretability and the Takagi-Sugeno-Kang model [10] has been used for precision modeling for accuracy. The steps for realization of neuro-fuzzy models are given as:

1. Derive rules for training network
2. Tune neural network parameters using fuzzy logic and add fuzzy logic criteria to define network
3. Realize a fuzzy sub-category. Identify membership functions through various algorithms such as
   a. Grid partitioning
   b. Back propagation
   c. Subtractive clustering
   d. Fuzzy C- means clustering
4. Represent fuzzy blocks for multi-layer networks:
   a. Fuzzification
   b. Fuzzy inference with rules or connections
   c. Defuzzification

Because of the limitation of precision, the neuro-fuzzy method has been more successful than fuzzy logic, as it incorporates interpretability from fuzzy logic and accuracy or precision from neural networks. Another difference is in handling of the rule base, where fuzzy logic limits the number of total rules, whereas, neuro-fuzzy has the ability to handle larger set of rules compared to conventional fuzzy but neuro-fuzzy has limitations to a number of input parameters.

C. PROBLEM STATEMENT

Despite several improvements discussed in literature, still there are various open points and limitations to conventional fuzzy and neuro-fuzzy design. It is a well-accepted fact that the conventional fuzzy and neuro-fuzzy systems both have limitations to dimension i.e., limitations to the large number of input variables and rules. The complexity of hierarchical systems has been best represented by a total number of rules [5]. Increase in input variables lead to increase in rules, thus increasing the overall complexity of the system. This limitation restricts the usage of fuzzy systems to solve complex problems and real-life applications with large dimensions. It thus increases the number of rules because these are directly proportional to the increase in number of input parameters and complexity of the system. In the last few decades, the hierarchical fuzzy system has appeared to be a viable solution to overcome the limitation of conventional fuzzy and neuro-fuzzy systems.

The aim of this paper is to address the problems and limitations possessed due to the conventional system by designing the fuzzy system in a form of hierarchy. This paper presents an approach to develop the hierarchical fuzzy based model with the focus to deal with large rule dimensions and especially in multi-output environments, without compromising the performance and effectiveness of the overall system. This paper presents an approach for rules reduction by eliminating common terms so as to reduce the complexity of the system. With ability to maintain composition both at functional level and linguistic level, hierarchical fuzzy based model is well suited and has the potential to handle data efficiently and provides a high transparency due to its rule base. The recent development has been around hierarchical fuzzy systems.

To the best of the authors' knowledge, nobody has exploited hierarchical fuzzy logic for multi-input multi-output systems. In this development, the authors for the first time have tried to avoid and remove duplication of common terms across various sub-systems.

There is no such algorithm available in literature to support the design of both type-1 and type-2 fuzzy inference systems [11] using hierarchical structure. In this paper, the algorithm presented for hierarchical approach has been studied in context of both type-1 and type-2 fuzzy inference systems. The type-1 and type-2 fuzzy inference system differ by membership function definitions and representation of rules. The type-1 fuzzy system has crisp membership definitions, however, the type-2 fuzzy inference system has fuzzy membership definitions in nature. The advantage of using type-2 fuzzy over type-1 fuzzy is its ability to manage uncertain and irregular environments. Type-2 fuzzy [14] has high computation due to a large number of parameters required to design and also it requires the type-reduction mechanism in its defuzzification level. The membership function of type-2 fuzzy inference system is known as interval type-2 fuzzy system and is bounded by two layers: lower and upper membership function. Type reducer defuzzification produces the output by averaging the intervals at various stages. Figure 1 below represents the membership function of type-1 and type-2 fuzzy inference system [11].

In this paper, type-1 fuzzy design has been considered but the same algorithm can be used for the type-2 fuzzy inference system.

Consider the diagram given in Figure 2 for proposed methodology in this paper. There are N-inputs and M-outputs. Users provide different inputs as a training data. The training data is grouped into two parts: common data and uncommon data. Sort both the common and uncommon data into the number of small size data samples. For every sample, generate a fuzzy inference system using fuzzy c-mean clustering. The clustering method reduces the number of rules required for the fuzzy system. Arrange these fuzzy subsystems in a desired hierarchical structure. Aggregate the outputs of the

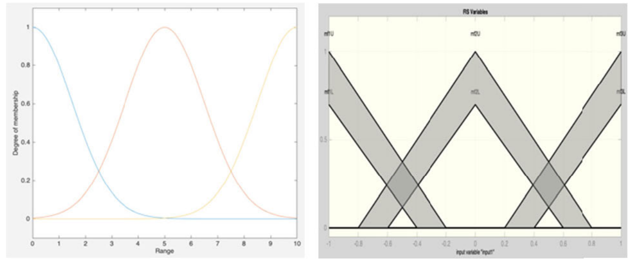


FIGURE 1. Membership function representation for type-1 and type-2 fuzzy inference system.

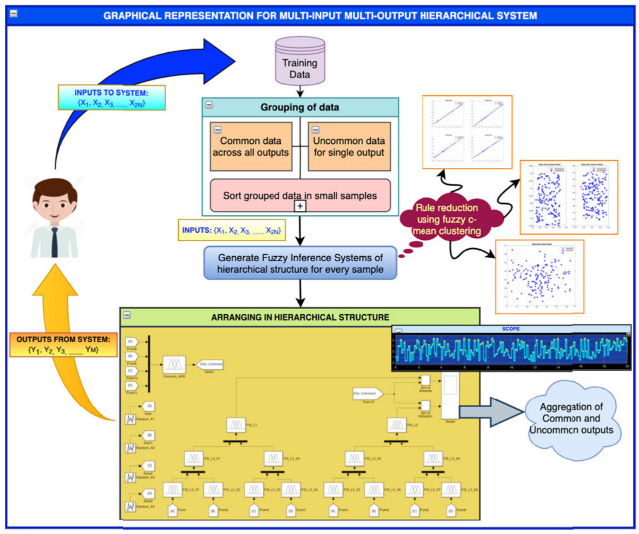


FIGURE 2. Graphical representation of multi input multi output hierarchical system.

hierarchical system obtained using both common and uncommon datasets. The multi input multi output hierarchical model has been designed using Simulink. The simulation has been done in the MATLAB and Simulink environment and output has been shown in scope.

Section II describes various hierarchical tree structures along with mathematical representation. The section also discusses various real-life applications of hierarchical fuzzy systems. Section III discusses the advantages of hierarchical fuzzy logic over conventional systems. Section IV presents the algorithm for the multi-input single-output hierarchical fuzzy system. The detailed steps have been discussed and steps to reduce rule-base using fuzzy c-mean clustering have been shown. Section V shows the comparison of a multi-input single-output system with conventional and hierarchical approaches. Section VI shows the systematic and generalized algorithm for a multi-input multi-output hierarchical system. Section VII shows the results of a multi-input multi-output hierarchical fuzzy system and its comparison with a conventional fuzzy system.

II. HIERARCHICAL FUZZY SYSTEMS

Hierarchical systems, as defined in wiki, are an arrangement of items (objects, names, values, categories, etc.) in which items are represented as being "above", "below", or "at the same level as" one another.

There has always been a challenge in using fuzzy systems to solve complex or high dimension problems, mainly due to its working limitation within a certain dimension. To overcome this problem, the hierarchical fuzzy system has appeared to be a viable and most effective option.

Fuzzy logic is a white-box system that enables transparency between interpretation and analysis. However, conventional fuzzy systems are the universal approximators with random function approximation to any accuracy. Due to transparency, fuzzy systems have been utilized in complicated systems and thus lead to its limitation of dimension, which becomes a bottleneck in most cases.

The limitation arises due to the increase in number of input parameters. The increase in input parameters leads to exponential increase in number of rules, number of parameters in mathematical relations and number of dataset or knowledge base required for searching systems.

The outcome of the above consequences results in over-fitting of systems that loses generalization of systems and its transparency. To deal with this problem, in the early 90's, Raju, Zhou and Kisner [5] ignited the idea of hierarchical fuzzy systems [5]. A number of sub-categorized structures, known as fuzzy logic units, with low dimensions are connected in the form of hierarchy. In some areas, various researchers consider hierarchical systems on the top of fuzzy logic to refine final output. Common applications to hierarchical fuzzy systems are classification, clustering, planning, and tracking systems etc. The system presented as MISO (Multi-Input and Single-Output) system, whereas MIMO (Multi-Input and Multi-Output) systems, without losing generalization, can be represented in several MISO subsystems. The lowest level of hierarchical structure gets access to real inputs whereas all the consecutive levels are connected to both output from previous level and real inputs.

A. HIERARCHICAL TREE STRUCTURE

The hierarchical fuzzy tree structure [5] can be described in multiple levels, and multiple systems in each level. The output of these fuzzy systems is used as inputs to the next consecutive level along with inputs as shown in Figure 3(a).

Chung and Duan [5] proposed the hierarchical fuzzy system as incremental structures in the year 2000, shown in Figure 3(b). The model showcases multiple-stage structure where every level represents one stage. In this architecture, there is one fuzzy system defined at every stage. The next level consumes the output from the previous level with original input parameters. To maintain accuracy, it is highly important to consider similar inputs for the first stage and less important shall be defined later.

Wang [9] proposed another architecture commonly defined as aggregated hierarchical structures, refer to Figure 3(c). In this architecture, every stage has multiple fuzzy systems, and all the original inputs are handled at the lowest stage. In the next stage onwards, outputs from previous systems are managed by next stage fuzzy systems.

To refine the accuracy of the model, another method defined by Karr and Magladena [5] is to consider and distribute original inputs at all levels as shown in Figure 3(d).

None of the above authors gave any methodical approach to design multi-input multi-output hierarchical systems. The comparison between different hierarchical tree structures is described above. In literature review, no one has developed any methodology for a hierarchical fuzzy system rules reduction especially for multi-input multi-output systems and also no multi-input multi-output hierarchical model design procedure is available in literature. In this paper, the authors for the first time present a systematic and generalized approach for the development of multi-input multi-output hierarchical systems with reduced common sub-systems by avoiding and removing duplication of common terms across various sub-systems of hierarchical systems. Any of the given hierarchical structures can be designed and implemented by the given algorithm.

B. REPRESENTATION OF HIERARCHICAL SYSTEMS

To represent hierarchical systems in mathematical form [13], consider an incremental hierarchical tree structure, where the first fuzzy logic unit at the lowest level consumes two real inputs. Similarly, the second fuzzy logic unit consumes the output from the previous fuzzy logic unit and real input and so on. This process advances till all the real inputs have been used. Assuming there are n-input parameters $\{x_1, x_2, x_3, \ldots x_n\}$ and $\{\widehat{x_1}, \widehat{x_2} \ldots \widehat{x_n}\}$ are the fuzzy variables extracted from input variables. 'm' presents total fuzzy rules, membership functions for inputs and outputs are presented by $\mu_{U_i^j}(x_i)$ *and* $\mu_{O^j}(y_i)$. With singleton fuzzifier and centroid defuzzifier, the hierarchical system can be represented below by equation 1(a) where mathematical forms of fuzzy logic units are shown by equation 1(b).

$$\begin{aligned} &IF\ (\widehat{x_1} = U_1^j)\ AND\ (\widehat{x_2} = U_2^j), \quad THEN\ (y_1 = O_1^j) \\ &IF\ (\widehat{x_{i+1}} = U_{i+1}^j)\ AND\ (\widehat{y_{i-1}} = O_{i-1}^j), \quad THEN\ (y_i = O_i^j) \end{aligned} \tag{1a}$$

where, 'j' $= 1, 2, 3, \ldots$, m; $\{U_1^j$ *and* $U_2^j; O_1^j$ represents the fuzzy sets for inputs and outputs respectively. 'i' ranges from $(2, 3, 4, \ldots n - 1)$. $\widehat{x_1}$ *and* $\widehat{x_2}$ present inputs to the first fuzzy logic unit. $\widehat{x_{i+1}}$ *and* $\widehat{y_{i-1}}$ present the real-input variable and output from the previous fuzzy logic unit respectively.

$$\begin{aligned} y_1 &= \frac{\sum_{j=1}^{m_1} Q_{1,j} \mu_{U_1^j}(x_i) \mu_{U_2^j}(x_i)}{\sum_{j=1}^{m_1} \mu_{U_1^j}(x_i) \mu_{U_2^j}(x_i)} \\ y_i &= \frac{\sum_{j=1}^{m_i} Q_{i,j} \mu_{O_{i-1}^j}(y_{i-1}) \mu_{U_{i+1}^j}(x_{i+1})}{\sum_{j=1}^{m_1} \mu_{O_{i-1}^j}(y_{i-1}) \mu_{U_{i+1}^j}(x_{i+1})} \end{aligned} \tag{1b}$$

C. APPLICATIONS OF HIERARCHICAL SYSTEMS

With the current transformation towards extension of wireless sensor networks (WSNs), network transformation towards Internet of things (IoT) etc., most of the real-life problems

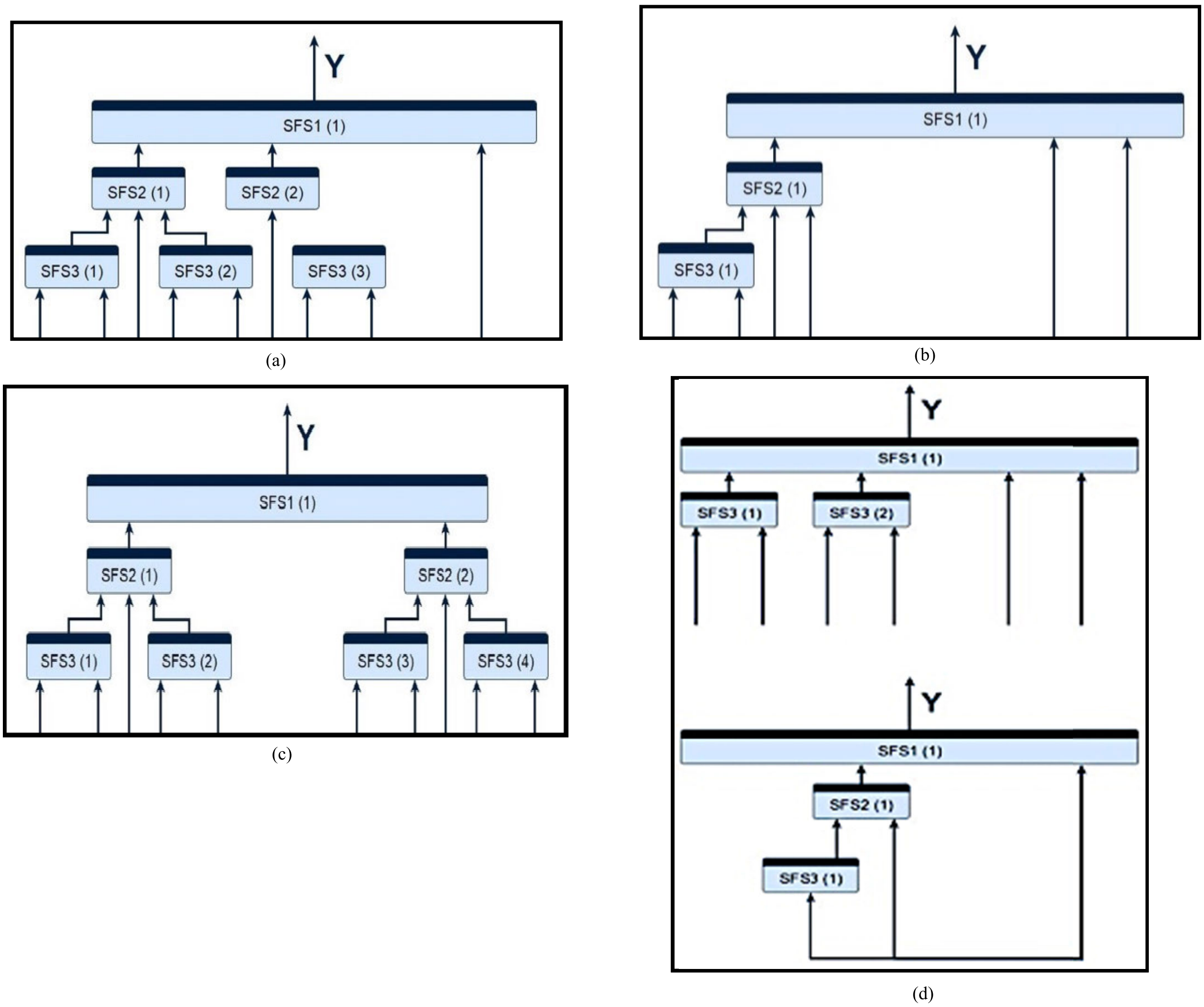


FIGURE 3. Structure of hierarchical systems.

are very much evolved and have uncertain behavior. The rule dimensions have been increased drastically and so thus the complexity of the system. The conventional systems are incapable of handling large dimension datasets. Thus, the increase in complexity for real-life problems becomes unmanageable and goes beyond the capabilities of conventional algorithms. However, hierarchical systems are an appropriate answer to those problems due to its ability to reduce the complexity, handle large dimensions and large uncertainty.

Other applications of hierarchical fuzzy in real-life is to implement data mining that requires multi-objective programming for Type 1 and Type 2 fuzzy systems [9]. Similarly, for the search problem and optimization task of structure and parameter spaces in model, the hierarchical fuzzy system has been used for multi-agent architecture [11] to efficiently parallelize and distribute the optimization tasks between structure and set of parameters.

Another application is the use of hierarchical fuzzy in designing adaptive Kalman filtering [8]. The objective of hierarchical fuzzy is to provide more effective representation irrespective of high dimensions. It consists of membership functions based on the concept of vagueness and uncertainty. However, complex abstractions can be achieved by combining functions with other mathematical systems. The system design using the Kalman filter is unable to handle uncertain behavior whereas hierarchical fuzzy can handle uncertain behavior. With similar context, the fusion of Kalman filter concept and hierarchical fuzzy system for the estimation process can be achieved. This inductive inference of the hierarchical fuzzy can be beneficial while supplementing from probabilistic theory and it is more reliable to relate to nonlinear relations.

III. ADVANTAGE OF HIERARCHICAL FUZZY LOGIC OVER CONVENTIONAL FUZZY LOGIC

In conventional fuzzy logic, membership functions are mainly used to create input segments and form possible interconnections between input and output in the form of rules. This leads to exponential growth in a number of rules as a number of inputs increases.

Consider a fuzzy system with 'n' inputs and each input has 'm' membership functions. Then the total possible rules can be described in the form of: $\mathbf{'m^n'}$. Similarly, for hierarchical fuzzy systems with n-inputs and each input has 'm' membership functions. It is assumed that two inputs are used for every fuzzy logic unit in each layer of hierarchy. Thus $\mathbf{'(n-1)'}$ fuzzy logic units are required to cover 'n' inputs. Then the total number of rules for the overall hierarchical fuzzy system can be described in the form of: $\mathbf{'(n-1).m^2'}$.

For a large number of inputs, it is very difficult to build conventional fuzzy systems that can handle such combinations of rules. This is why the hierarchical fuzzy system has been preferred over the fuzzy system that offers solutions with reduced rules. Consider Mackey-Glass (MG) time delay [12] differential equation (2):

$$\dot{x}(t) = \frac{0.2x(t-T)}{1+x^{10}(t-T)} - 0.1x(t) \tag{2}$$

For the above time-series, considering the known values of time series up to "t" point in time, to predict future values at some point (t + P). A standard method is to map D sample points with every 'u' unit in time (x(t-(D-1) u), ..., x(t-u), x(t)) to a predicted future value x (t + P). Using the conventional settings for predicting time series, assume D = 4 and u = P = 6, the example for an input training data can be considered in vector form as represented in equation (3):

$$\mathbf{W(t)} = [\mathbf{x(t-19)}, \mathbf{x(t-12)}, \mathbf{x(t-6)}, \mathbf{x(t)}] \tag{3}$$

Gather 500 samples as training data extracted from above equations. Build a conventional fuzzy system using the dataset. Design a fuzzy inference system using fuzzy clustering and with 16 cluster points. It is assuming that all the inputs are categorized into two gaussian membership functions each. For conventional fuzzy logic, the maximum possible fuzzy rules are $2^4 = 16$ rules.

Figure 4 shows the conventional fuzzy systems for Mackey-Glass time-delay differential model. Using the same training data set, design a two-level hierarchical fuzzy system. Assuming level-1 with two fuzzy logic units and each unit contains two inputs. The two outputs from level 1 i.e., output from each fuzzy logic unit from level 1, becomes input to the level-2 fuzzy logic unit. The output of level 2 becomes the final output.

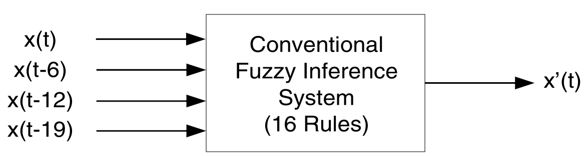


FIGURE 4. Conventional fuzzy system for Mackey-Glass time delay differentials model.

As discussed above, each input has categorized into two membership functions. With two inputs, each fuzzy logic unit at level 1 has the maximum of $2^2 = 4$ rules possible. Consider four rules for the level-2 fuzzy logic unit. Total rules in hierarchical fuzzy structure is reduced to $4 + 4 + 4 = 12$ rules.

Figure 5 shows the hierarchical fuzzy systems for Mackey-Glass time-delay differential model.

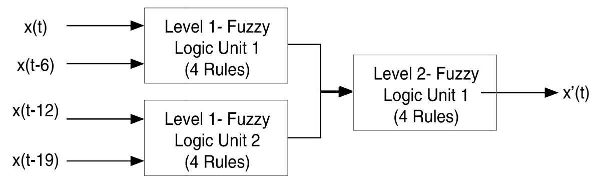


FIGURE 5. Aggregated hierarchical fuzzy structure for MG time delay differentials model.

Figure 6 shows the comparison between three approaches [9]: final x(t) extracted from equation (1), conventional and hierarchical fuzzy system.

Where, 'Expected' = x(t) extracted from equation (1); 'Conventional' = Fuzzy inference system output x'(t) from Conventional System; 'Hierarchical' = fuzzy inference system x'(t) from a two-layer Hierarchical fuzzy inference system. It is observed that the correlation between 'Expected' and 'Conventional' fuzzy systems is 0.93 and correlation between 'Expected' and 'Hierarchical' fuzzy systems is 0.91.

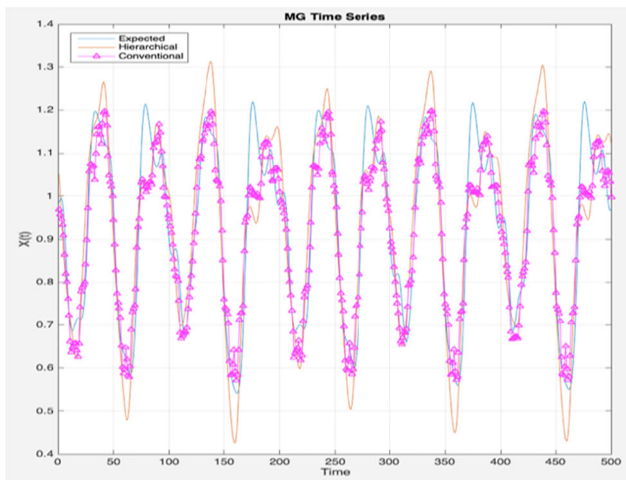


FIGURE 6. Comparison of various approaches.

Table 1 below gives descriptive comparison for conventional fuzzy logic, neuro-fuzzy logic and hierarchical fuzzy logic approach. The comparison has been made on the basis of number of rules, system performance, processing time, development of overall system etc.

IV. DESIGN OF MULTI INPUT SINGLE OUTPUT SYSTEM USING HIERARCHICAL FUZZY LOGIC

For Multi-Input-Single-Output, one of the methodologies to design hierarchical models is discussed in paper "Hierarchical Fuzzy Systems" by Radek Sindelar [4]. This paper addresses rule-base explosion in conventional systems for

TABLE 1. **Comparison between fuzzy, neuro-fuzzy and hierarchical fuzzy logic.**

Fuzzy Logic Approach	Neuro-Fuzzy Approach	Hierarchical Fuzzy Approach
Time consuming	Learning process may be time consuming	Less time consuming as every fuzzy logic has reduced number of rules to process
Requires optimization and fine tuning before operating fully functional	Solutions cannot be interpreted in linguistic form very easily.	Easy to interpret and optimize due to a smaller number of rules and dimensions of each fuzzy logic unit
Could lead to problem of finding suitable membership functions for fuzzy systems	Black box architecture. Difficult to identify whether solution is plausible (cannot interpret in terms of rules)	White-box architecture. Due to multi-layer structure, membership function can be controlled
Degradation of performance for large number of rule base	Degradation of performance for large number of rule base extracted from training data	Better performance compared to conventional fuzzy with large number of rule base
More processing is required with a large [P x Q] system. Where P = number of inputs and Q = number of outputs	More processing is required with a large [P x Q] system. Where P = number of inputs and Q = number of outputs	Distributed systems with reduced [P' x Q'] systems. Where P' = number of reduced inputs where [P' < P] and Q' = number of reduced outputs where [Q' < Q] Provides enhanced processing.
Total number of rules is less than equal to m^n	Total number of rules is less than equal to m^n	Total number of rules = (n-1).m^2; Where Total rules required by hierarchical system < total rules required for conventional system
Difficult to develop fuzzy model with big input and rule dimension systems i.e., high number of input-output parameters	Neural Network is dependent on the availability of data. Mathematical model or pre-existing knowledge of a problem is not necessary.	Ability to develop hierarchical model for big data without any limitation to input and rule dimensions

large dataset and proposed solution by converting to hierarchical structure. The author developed the hierarchical system layer-by-layer to avoid making a conventional model without leveraging previous information about known structures. The author introduces structure, prototype algorithm and weighted analysis. It is preferred to connect inputs with similar behavior to one subsystem and inputs with different behavior to another.

Consider the fuzzy system with three inputs (x_1, x_2, x_3) and 1 output (y) has been divided into two subsystems: S1 and S2 with different rule bases corresponding to their respective input-output combinations. Assume three inputs two-layer hierarchy structure, where x_1 and x_2 are inputs to subsystem S_1 and output of subsystem S1 along with input x_3 drives subsystem S2 to obtain final output.

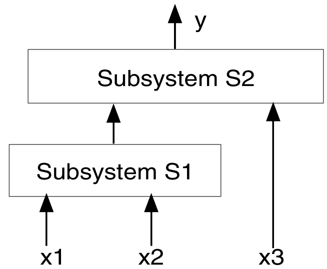


FIGURE 7. **Flowchart of 3 × 1 hierarchical system.**

Figure 7 shows three inputs two-layer hierarchy structure, where X_1 and X_2 are inputs to subsystem S1 and output of subsystem S1 along with input X_3 drives subsystem S2 to obtain final output. The proposed algorithm started with the lower layer. Subsystem S1, comprises Inputs x_1 and x_2. However, in the next layer, a combination of input and output from the lower layer is passed through subsystem S2.

The rule base of these subsystems is described as:

For subsystem S1:

$$R_1 : IF\ x_1\ is\ A_i\ AND\ x_2\ is\ B_i,\ THEN\ z\ is\ U_i$$

For Subsystem S2:

$$R_2 : IF\ x_3\ is\ C_i\ AND\ z\ is\ U_i,\ THEN\ y\ is\ P_i$$

The fundamental limitation of the conventional fuzzy is the increase of the rule base proportionally with the number of input variables. With a large set of data, there is a high probability to lose transparency and accuracy etc. There are several approaches available in the literature on designing hierarchical fuzzy systems [2] i.e., decomposition of large conventional fuzzy into small systems with simple sub-structures and interconnect them all with certain methodology.

For example, consider a fuzzy system with two layers with single output per layer. The output of one subsystem becomes the input to another subsystem. This paper presented a procedure to design hierarchical fuzzy systems and in any of Mamdani and Sugeno type methods. The algorithm starts with the lower level, where inputs are real. The implementation uses a two-side Gaussian membership function that is smooth, continuous, differentiable. Triangular membership functions can be used as it is presented by straight line and are simple to represent. The reason to choose gaussian membership function is because most of the real-life applications behave in a gaussian manner.

Like conventional fuzzy, each hierarchical system has three blocks [3] for each subsystem: Fuzzifier, Rule editor and Defuzzifier. The "centroid" weighted approach has been considered as a defuzzification method. The approach considers cluster approach to define subsystems, extracted from random data. The steps to design hierarchical subsystems are defined below. As given above, hierarchical systems have been illustrated by various authors. It has been implemented for a two-inputs one-output system by Radek Sindelar [4]. In this paper, we implement the procedure for a new example. All the respective tables and results are generated using the MATLAB simulation environment.

Considering a nonlinear function 'y' with training set of random generated 1500 samples in specified range, where 'y' is represented by equation (4) as:

$$\mathbf{y} = \mathbf{x_1(1 - x_2) + x_2x_3} \tag{4}$$

Consider input membership functions as: small and large, whereas output membership functions as: small, medium, and large. To convert in hierarchical system, expression above is further decomposed and is shown by equation (5):

$$\mathbf{y} = \mathbf{z + x_2x_3} \tag{5a}$$

$$\mathbf{z} = \mathbf{x_1(1 - x_2)} \tag{5b}$$

The equation 5(b) represents subsystems S1 and equation 5(a) represents subsystem S2. Where subsystem S1 has two inputs: x_1 and x_2; and one-output 'z'; however, subsystem S2 has two inputs: x_3 and z; and one-output y. It is observed that output 'z' from subsystem S1 becomes input to the subsystem S2. Following all the above steps following mathematical sequence, Table 2 shows the rule base for subsystem S1 and S2.

TABLE 2. Rule base.

x_1 (Small, Large)	x_2 (Small, Large)	z (Small, Large)
Small	Small	Small
Small	Large	Small
Large	Small	Large
Large	Large	Small

(a) Table for Subsystem S1

x_3 (Small, Large)	z (Small, Large)	Final Output 'y' (Small, Medium, Large)
Small	Small	Small
Small	Large	Medium
Large	Small	Medium
Large	Large	Large

(b) Table for Subsystem S2

Rules shown in Table 2 are generated in following steps:

Step 1 (Random Data Generation): Generate 'P' numbers in the interval (a, b) with the formula given below in equation (6). Where, 'r' defines random data and 'rand(P,1)' is the MATLAB function to generate 'P' random numbers

$$\mathbf{r = a + (b - a).^{*}rand(P, 1)} \tag{6}$$

Step 2 (Scaling of Parameter in Range): Fuzzy works well in the specified range of [0, 1], to scale, divide each individual element of input-output from its own maximum value.

Step 3 (Center Points Using Fuzzy Clustering): Use "find cluster" to identify the center points within the required number of clusters [7].

Center Points = find cluster ([In, Out], No. of Cluster)

Step 4 (Generate FIS Model Using ANFIS):

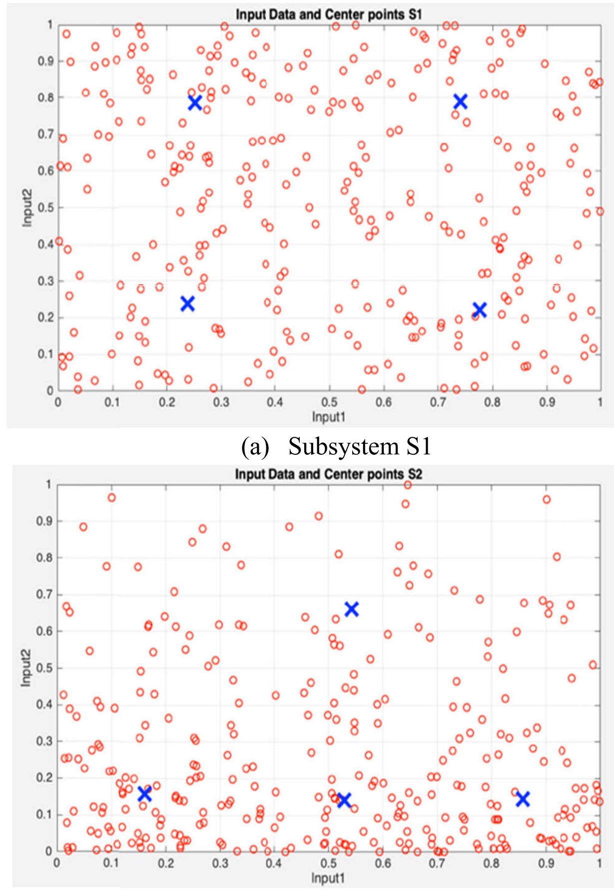
: MATLAB Syntax for Type-1 FIS:

Type1FIS = genfis3([In, Out], 'Type', 'Cluster')

: MATLAB Syntax for Type-2 FIS:

Type2FIS = convertToType2(Type1FIS)

To design the sub-system, any of the four defuzzification methods (grid partitioning, back propagation, subtractive clustering and fuzzy C-mean clustering) can be picked in this research. Fuzzy C-Mean clustering [7] is considered due to its ability to generate subsystems with both Mamdani- and Sugeno- type fuzzy inference systems. Figure 8 represents the fuzzy clustering of inputs x_1 and x_2 within the region for



(a) Subsystem S1

(b) Subsystem S2

FIGURE 8. Fuzzy clustering of subsystems S1 and S2.

subsystem S1 and of output z of subsystem S1 and input x_3 for subsystem S2. In figure 8 below, for S1, x_1 is presented as Input 1 and x_2 as Input 2 respectively. For S2, x_3 is presented as Input 1 and z as Input 2 respectively. With required center points for a dataset, these regions are contrived to four or eight points.

Where, top plot describes the input space of S1 with four clusters. These center points along with expected output are further used to extract rules defined in Table 3 below:

TABLE 3. Subsystem S2 – cluster.

x_1 (Small, Large)	x_2 (Small, Large)	z (Small, Large)
Small	Small	Cluster I
Small	Large	Cluster II
Large	Small	Cluster III
Large	Large	Cluster IV

(a) Table for Subsystem S1 – Cluster

x_3 (Small, Large)	z (Small, Large)	Output 'y' (Small, Medium, Large)
Cluster I	Small	Small
Cluster I	Large	Medium
Cluster II	Small	Medium
Cluster II	Large	Large
Cluster III	Small	Small
Cluster III	Large	Medium
Cluster IV	Small	Small
Cluster IV	Large	Medium

(b) Table for Subsystem S2 – Cluster

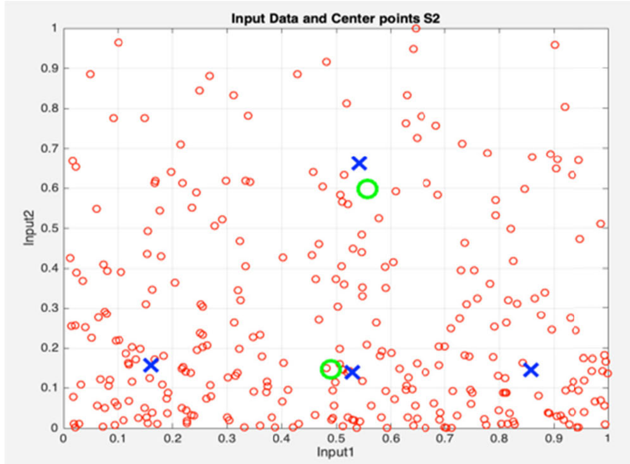


FIGURE 9. Fuzzy clustering for subsystem S2 with four center points contrived to two center points.

TABLE 4. Subsystem S2 – reduced cluster.

x_1 (Small, Large)	x_2 (Small, Large)	z (Small, Large)
Small	Small	Cluster I
Small	Large	Cluster II
Large	Small	Cluster III
Large	Large	Cluster IV

(a) Table for Subsystem S1– Cluster

x3 (Small, Large)	z (Small, Large)	Final Output 'y' (Small, Medium, Large)
Cluster V	Small	Small
Cluster V	Large	Medium
Cluster II	Small	Medium
Cluster II	Large	Large

(b) Table for Subsystem S2 – Cluster (reduced)

RULES SIMPLIFICATION

Data for every subsystem is clustered and then the relationship is established as rules. Consider multiple inputs that have similar clusters and same output, then these clusters can be clubbed together to reduce the number of rules. Figure 9 shows four clusters of subsystems S2 and its reduced number of clusters to two. As three out of four clusters have the same output behavior for input x_2, so these clusters can be clubbed to one cluster. Thus, a number of clusters has been reduced from four to two. The number of clusters represents the number of rules.

The similar behavior clusters can be clubbed into another cluster V and thus further simplified to two clusters instead of four. Table 4 shows the re-structure of rule base from eight rules to four rules for subsystem S2. However, the rule base for subsystem S1 remains the same.

V. VERIFICATION OF MULTI INPUT SINGLE OUTPUT SYSTEM (MISO)

The idea of merging clusters in the rule base is to simplify the structure of the system but it is expected to maintain and retain the total rule base behavior. Two types of aggregation methods have been discussed: maximum and weighted average. Table 5 shows the properties, such as, standard deviation and variance, of conventional and hierarchical fuzzy models for different aggregation methods using MATLAB commands [10].

VI. DESIGN OF MULTI INPUT MULTI OUTPUT SYSTEM USING HIERARCHICAL FUZZY LOGIC

The main contribution of the paper is the development of generalized algorithms for multi-input multi-output hierarchical systems. This concept has not been presented by any author so far in the literature.

In this paper, the multi-input single-output approach has been extended to multi-input multi-output for the first time. This approach is different from Radek in the sense that in Radek's approach, subsystems have been generated based on decomposition of mathematical expressions, whereas in our approach subsystem has been generated by segmenting input-output data. Each segment leads to a fuzzy unit in the structure.

TABLE 5. Properties of various fuzzy approaches.

	Conventional Fuzzy Logic vs Hierarchical Fuzzy Logic	
Aggregation	Maximum	Weighted Average
Correlation	0.96	0.98

(a) Correlation between weighted approach

	Conventional Fuzzy		Hierarchical Fuzzy	
Aggregation	Max	Weighted Average	Max	Weighted Average
Standard Deviation	0.2265	0.1480	0.2054	0.2100
Variance	0.0513	0.0219	0.0422	0.0441

(b) Standard deviation and variance

Consider n outputs, where all outputs are functions of four inputs: x_1, x_2, x_3, x_4. Equation (7) represents outputs as:

$$\begin{aligned} \mathbf{y_1} &= \mathbf{f}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}; \ldots \\ \mathbf{y_2} &= \mathbf{f}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}); \ldots \\ &\quad \ldots. \\ \mathbf{y_n} &= \mathbf{f}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, \mathbf{x_4}); \end{aligned} \tag{7}$$

To design hierarchical structure, the main strategy is to gather all input-output combinations that are common among all the outputs. This will reduce the number of fuzzy logic units in the hierarchical structure. Using these input-output combinations, a fuzzy logic system has been designed that will be common for all outputs. The generic procedure is as described below:

- Generate random data for all the outputs using mathematical equations
- Gather all the input-output data combinations that are common among all the outputs
  - Design common fuzzy logic system
  - Evaluate output
- Gather all the input-output combinations that are uncommon across all the outputs separately
  - Define hierarchical level and number of fuzzy logic unit per level
  - For individual output
    - Design fuzzy logic units
    - Evaluate final output
- Final output for individual function will be average between evaluated output from common component and output from hierarchical structure

Fuzzy logic supports multi-input-single-output (MISO) systems. In this paper, the methodology to design multi-input-multi-output (MIMO) is presented using hierarchical fuzzy logic. Random data of 1500 samples have been taken as training data and then evaluate output from mathematical expressions for Y_1 and Y_2. For multi-output systems, to reduce the complexity, filter the input-output data common across all systems and generate an independent fuzzy inference system. For uncommon data, depending on the number of fuzzy logic units required at a lower level of hierarchy, pool input columns in various samples with the same desired output. Find the required number of cluster points and consider these points as a rule base. Generate a fuzzy inference system for every fuzzy unit using these samples. The output from a lower layer becomes input to the next layer. Repeat the same process to collect input, pool in various samples, generate a fuzzy inference system and evaluate it. For any system, the measured output is average of output from final fuzzy output and common fuzzy. The pseudo code is given as:

Algorithm Pseudo Code to Design Hierarchical Systems

Data Generation: *Generate random data of 1500 samples*

Segmentation: *Using Mathematical analysis, create and pool common and uncommon inputs and pair them with output*

Find Cluster: *Find center points (relation between Input and Output) for each group by following syntax*

Center Points = findcluster ([Input, Output], Total Cluster)

Execution:

- *Define membership functions for Input and Outputs*
- *Design fuzzy inference system for common dataset and evaluate the output*
- *For all the outputs follow next steps separately*

  For: Hierarchy Level = '1' to (Final Level)
  - ***If (Lower level of hierarchy)***
    - *Create different sub-groups or segments from uncommon dataset*
  - ***Else***
    - *Output from previous level becomes input to the next level with real output*
    - *Create different sub-groups or segments from input (previous level output) dataset, keeping same output*
  - *Generate Fuzzy Inference system as follows:*

    For (Number of Sub-groups)
    - *Find center points (IO Mapping)*
    - *Generate Type-1 FIS:* *Type1FIS = genfis3([In, Out], 'FIS', 'Cluster')*
    - *(Optional) Generate Type-2 FIS: Type2FIS = convertToType2(Type1FIS)*
    - *Evaluate Output using "evalfis"*

Result: *Output is average of final hierarchical output for individual system and final output from common fuzzy unit*

The algorithm presented below can be used for N outputs. Following steps are given below to realize both type-1 and type-2 multi-input-multi-output systems for 'N' outputs:

STEP [1]. Consider randomly generated training data separately

a. 1500 random samples have been generated for both y_1 and y_2 separately as shown below in (8):

$$\sum_{i=1}^{2} \boldsymbol{r}(\boldsymbol{y}_i) = \boldsymbol{a} + (\boldsymbol{b} - \boldsymbol{a}).^{*}\boldsymbol{rand}(\boldsymbol{Q}, \mathbf{1}) \tag{8}$$

where, Q denotes the generation of random numbers in the range (a, b) and "i" defines output.

STEP [2]. Define following information:

a. Number of hierarchical levels
b. Fuzzy logic units in every level
c. Number of clusters corresponds to required number of rules for every fuzzy logic unit

STEP [3]. Grouping

a. Group common Input-Output samples common across data samples for both the outputs
b. Segment uncommon data separately for each output

STEP [4]. For the first or lower level of hierarchical structure i.e., aggregated tree structure, all the real input adheres to the first or the lower level of hierarchical structure. This layer consists of a fuzzy logic unit generated for all input samples extracted in Step 3. The steps to generate fuzzy logic unit are as follows:

a. Take input sample and map with desired output
b. Using MATLAB commands [10], the syntax to generate Fuzzy inference system is defined below:
   i. Type1FIS = genfis3([Input], [Output],'FIS Type', 'Number of Cluster'); Where –
      1. [Input] = Input sample matrix
      2. [Output] = Expected output matrix
      3. 'FIS Type'
      4. "Number of Clusters" = Desired number of rules for fuzzy logic unit
   ii. If Type-2 Fuzzy is designed, use syntax: Type2FIS = convertToType2(Type1FIS); where 'Type1FIS' is the type-1 fuzzy inference system and 'Type2FIS' is type-2 fuzzy inference system converted from type-1 inference system
   iii. Evaluate output of fuzzy unit with 'evalfis' command and Store the values in a buffer

STEP [5]. For the next level(s) of hierarchical structure,

a. Take all the output from previous hierarchy levels and consider them as input to current level
b. Create a group of segments of matrix and map each segment with map with desired output
c. Using MATLAB commands, the syntax to generate Fuzzy inference system is defined below:
   i. Type1FIS = genfis3([Input], [Output], 'FIS Type', 'Number of Cluster'); Where –
      1. [Input] = Input sample matrix
      2. [Output] = Expected output matrix
      3. 'FIS Type'
      4. "Number of Clusters" = Desired number of rules for fuzzy logic unit
   ii. If Type-2 Fuzzy is designed, use syntax: Type2FIS = convertToType2(Type1FIS); where 'Type1FIS' is the type-1 fuzzy inference system and 'Type2FIS' is type-2 fuzzy inference system converted from type-1 inference system
   iii. Evaluate output of fuzzy unit with 'evalfis' command and Store the values in a buffer

STEP [6]. For the last hierarchy level

a. Take all the output from previous hierarchy levels and consider them as input to current level
b. Create a group of segments of matrix and map each segment with map with desired output
c. Using MATLAB commands, the syntax to generate Fuzzy inference system is defined below:
   i. Type1FIS = genfis3([Input], [Output], 'FIS Type', 'Number of Cluster'); Where –
      1. [Input] = Input sample matrix
      2. [Output] = Expected output matrix
      3. 'FIS Type'
      4. "Number of Clusters" = Desired number of rules for fuzzy logic unit
   ii. If Type-2 Fuzzy is designed, use syntax: Type2FIS = convertToType2(Type1FIS); where 'Type1FIS' is the type-1 fuzzy inference system and 'Type2FIS' is type-2 fuzzy inference system converted from type-1 inference system
   iii. Evaluate output of fuzzy unit with 'evalfis' command and Store the values in a buffer

STEP [7]. Repeat STEP [4] – STEP [6] for common data samples

STEP [8]. Final output is the average of evaluated output at last level of hierarchy at STEP [6] and output at STEP [7].

In the above said methodology, three hierarchy levels are used, and the number of fuzzy logic units considered for each level is 2^R, where R is the number of levels and R = 1 is the top level. Higher value of R represents the lower level of hierarchy.

The algorithm is illustrated with the help of the following example below.

Two non-linear mathematical functions y_1 and y_2 have been taken with a training set of R random generated sets, where R is the number of samples. Equation (9) represents y_1 and y_2 as:

$$\begin{aligned} y_1 &= x_1x_2 + x_3x_4 \\ y_2 &= x_1x_2 + (x_3 + x_4) \end{aligned} \tag{9}$$

To convert in hierarchical system, the equation (9) above can be further decomposed into parts as shown in (10):

$$\begin{aligned} y_1 &= c + z_1 \\ y_2 &= c + z_2 \\ c &= x_1x_2 \\ z_1 &= x_3x_4 \\ z_2 &= x_3 + x_4 \end{aligned} \tag{10}$$

where, 'x_1', 'x_2', 'x_3', 'x_4' represent four inputs; 'c' represents the common component or term between two outputs y_1 and y_2. 'z_1' and 'z_2' are the subcomponents for output y_1 and y_2 respectively. The relationship between z_i and y_i are shown in equation (10).

Figure 10 shows the block diagram of a multi-input multi-output system with four inputs (x_1, x_2, x_3, x_4) and two outputs (y_1 and y_2). In the diagram, 'c' represents the fuzzy logic unit, which is common across outputs y_1 and y_2. Whereas z_1 represents the subsystem of output y_1 and z_2 represents the subsystem of output y_2. The combination of 'c' and 'z_1' gives final output 'y_1' and the combination of 'c' and 'z_2' gives final output 'y_2'.

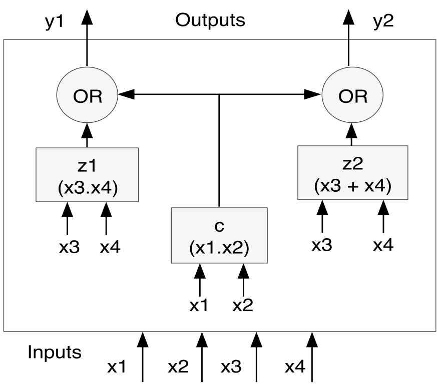


FIGURE 10. Block diagram of 4 × 2 system.

Following all the above algorithm steps and mathematical sequence, Table 6 shows the rule base between inputs and outputs of a single fuzzy logic unit of hierarchical fuzzy system, as mentioned in figure above. In this example, inputs and outputs are categorized in three membership functions each as: small ('S'), medium ('M') and large ('L').

TABLE 6. Rule base.

Input 1	Input 2	Y1	Y2
S	S	S	S
S	S	M	S
S	S	S	S
S	S	M	M
S	L	S	S
S	L	S	M
S	L	S	M
S	L	L	L
L	S	S	M

Where, 'Input 1' and 'Input 2' are the inputs to each fuzzy logic unit of every layer of a hierarchical fuzzy system. 'Y_1' and 'Y_2' are the two outputs of a hierarchical fuzzy system.

A. REDUCTION OF RULES

For the above said system with four inputs and two outputs, where each input and output have three membership functions. To design the system using conventional fuzzy requires a total number of ($2{*}3^4 = 162$) rules for a two-outputs system. However, to design the above system using hierarchical fuzzy, requires the total of ($2{*}7{*}3^2 + 9 = 135$) rules for a two-outputs system. Where, the common fuzzy logic unit between two outputs has nine rules.

B. SIMULINK MODEL

The Simulink model for the hierarchical system described in equation (10) is given below, to aid the reader in implementation of hierarchical fuzzy systems for control applications. Figure 11 shows the Simulink model with four inputs (x_1, x_2, x_3, x_4) and two outputs (y_1 and y_2) using two-layers hierarchical fuzzy logic.

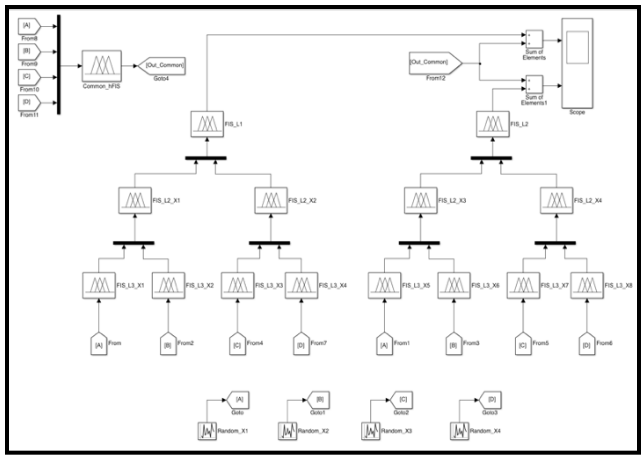


FIGURE 11. Simulink model for two-layer 4 × 2 system.

A system has been generated using the common data samples across two outputs. For data samples other than common data, hierarchical fuzzy approach has maintained separate fuzzy units. This approach will provide a reduced rule base, as the common fuzzy inference system has been used for both outputs that have common input output relation. The Simulink model has been designed to showcase the system overview of multi-input multi-output systems. 'Scope' in the diagram represents outputs: y_1 and y_2.

The steps to design 3-layer hierarchical fuzzy model using Simulink have been described below as:

1. Random data generator block has been used to generate 1500 random data samples as per equation (9) for 4 inputs (x_1, x_2, x_3, x_4) and 2 outputs (y_1, y_2)
2. Consider N-layer (where, N=3) aggregated hierarchical tree structure with 2^{N-1}, 2^{N-2}, 2^{N-3} and so on with fuzzy logic units at the first, second and third layer of hierarchy. Assuming all the fuzzy logic units at the first layer have $[1 \times 1]$ input-output system and $[2 \times 1]$ input-output system at other layers.
3. Rules Generation for fuzzy logic unit: Using fuzzy c-mean clustering, find desired number of clusters. Each cluster represents the rule/relation between input and output.

4. Fuzzy inference system generation: using clusters and input-output data samples, use the ''genfis3'' command in MATLAB to auto-generate a fuzzy inference system.
5. Filter the data samples in two parts. Input-Output data samples. In other words, filter common and uncommon datasets between outputs y_1 and y_2
6. Design of a fuzzy inference system. Generate a fuzzy inference system using fuzzy c-mean clustering with desired clusters using samples filtered in step 5. Evaluate output using ''evalfis'' command
7. Final Outputs for (y_1 and y_2): Using centroid defuzzification method, final output is the average of output from following: Common and not-common fuzzy system

C. CLUSTERING

Figure 12 (a) shows the fuzzy clusters for lower level with four center points. System has four fuzzy logic units and with four inputs (x_1, x_2, x_3, x_4) respectively. For the lower layer, one real input consumes one fuzzy logic unit and every unit has one segregated output. Number of clusters will be contrived to four points for every fuzzy logic unit.

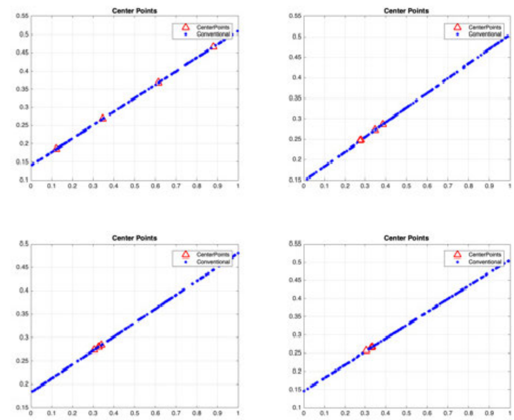
Figure 12 (b) shows the fuzzy clusters for level-2 with four center points. System has two fuzzy logic units and with four outputs from previous levels as inputs. For level 2, two outputs from the previous level are consumed by one fuzzy logic unit and every unit has one segregated output that goes to the next level. Figure 12 (c) shows the fuzzy clusters for level 3 with defined center points for every fuzzy logic unit. System has one fuzzy logic unit and with two outputs from previous levels as inputs.

Similarly, the data has been extracted by filtering common data from the datasets of y_1 and y_2 (assuming outputs are common for the input combinations). Using common data, a fuzzy logic unit with 4 inputs and 1 output has been generated as a common component. The objective of generating a common component is to reduce the number of fuzzy logic units and the rule bases, as this will reduce complexity. The final output is the average between the output of the final fuzzy logic unit and common component, as shown in the Simulink block diagram.

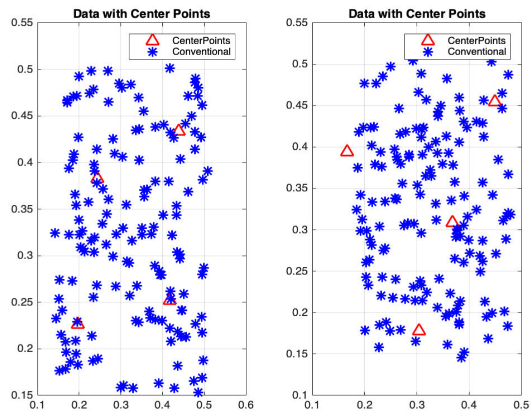
D. SYNTHESIS USING XILINX ISE ENVIRONMENT

With the increasing use of FPGA in almost all the areas. The hardware implementation of fuzzy systems has been a topic of recent interests [16]. The implementation of hierarchical fuzzy logic using FPGA would be a topic of future interest by various authors.

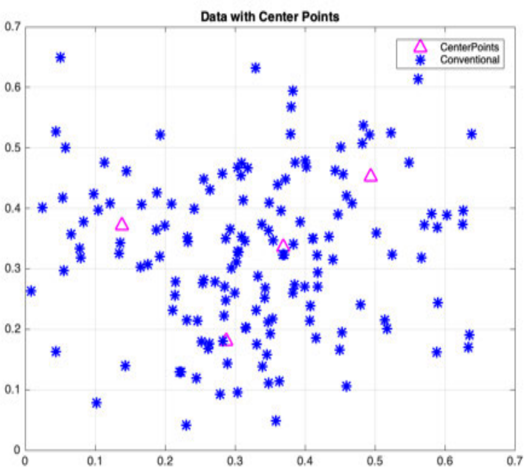
In this paper, the Simulink environment has been used to develop and validate models for the given approach. For simulation in real-time, an FPGA environment has been considered using HDL language [15]. To the best of the authors' knowledge, no one has presented the steps to synthesis using Verilog HDL. In this paper, Verilog HDL has been considered. One of the major benefits of Verilog for the developer is the ability to quickly create, simulate and verify a test model.



(a) Clustering of four fuzzy units for level 1



(b) Clustering of two fuzzy units for level 2



(c) Clustering for final level 3

FIGURE 12. Fuzzy clustering of various levels.

The steps below present the procedure used to auto-generate Verilog code using the Simulink HDL coder and simulate the same in Xilinx platform.

The first step is to generate Verilog code using the Simulink HDL coder. In the code generation settings, select Generate HDL option as code style and select "Verilog" option as target code generation. After the settings have been completed, compile the system and generate Verilog code. HDL Coder does create a link to Xilinx ISE project and runs the selected logic synthesis and place-and-route steps in generated code.

The second step is to simulate and synthesize in an FPGA environment. The simulation of Verilog code has been done via ISim simulator. ISE 14.1 is used to synthesize with Xilinx ARTIX 7 XC7A100T as a target device with 4ns IO delay. HDL design is synthesized where an abstract form of desired circuit behavior is turned into a design implementation in terms of logic gates. Netlists are generated by post synthesis. The program maps the logical design to Xilinx. It performs the DRC check on the design and it then maps directly to the target Xilinx FPGA Device. The next step is to assign cells to the specific locations within an FPGA device, called cell placement where all the cells and resources are connected with the step called Routing. After place and route is completed, a bit stream is created that configures FPGA when downloaded into it, using the program iMPACT.

VII. VERIFICATION OF RESULT FOR MULTI INPUT MULTI OUTPUT SYSTEM (MIMO)

Figure 13 shows the comparison [9] among three approaches. Green shows outputs 'y_1' and 'y_2' from the mathematical expression shown in equation (9), Red shows the fuzzy inference system output from conventional logic and Blue is the fuzzy inference system output from hierarchical fuzzy logic.

Raw comparison represents the final result for y_1 and y_2 using three approaches: mathematical expression as presented in equation (9), conventional fuzzy system and hierarchical fuzzy system. With raw comparison, it is observed that results from hierarchical fuzzy systems are much closer to results from mathematical expression. The expected results will represent the results from the mathematical expression presented in equation (9).

To compare the quality of the methodology presented in this paper, two criteria have been used. These criteria have been described below as:

1. The mean average percentage error defined as

$$MAPE = \frac{1}{L}\left(\sum \frac{|Y_{measured} - Y_{desired}|}{Y_{measured}}\right) * 100\%$$

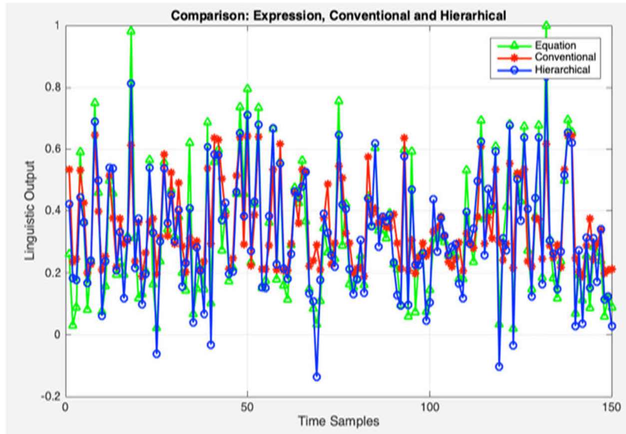
2. Zone Error – which counts the sample belongs to class

$$Class\ I = \frac{|Y_{measured} - Y_{desired}|}{Y_{measured}} \leq 5\%$$

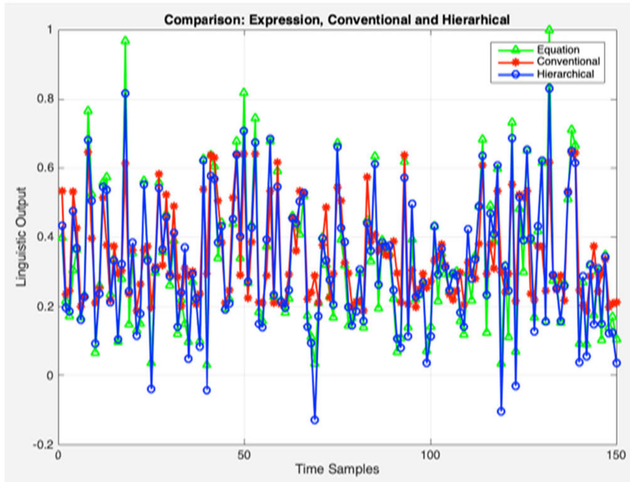
$$Class\ II = 5\% < \frac{|Y_{measured} - Y_{desired}|}{Y_{measured}} \leq 10\%$$

$$Class\ III = \frac{|Y_{measured} - Y_{desired}|}{Y_{measured}} > 10\%$$

where, 'L' = length of samples, '$Y_{measured}$' = output from model and '$Y_{desired}$ = output from mathematical expression.



(a) Raw comparison for output 'y_1'



(b) Raw comparison for output 'y_2'

FIGURE 13. Comparison between different approaches.

TABLE 7. Fuzzy models comparison.

Model	Total Rules	MAPE	Class I	Class II	Class III
Conventional Y_1	162	3.19	32	41	27
Conventional Y_2	162	5.99	33	46	21
Hierarchical Y_1	135	4.54	82	8	10
Hierarchical Y_2	135	4.91	66	22	12

Table 7 shows comparison between different approaches.

While comparing the qualities between conventional fuzzy systems and hierarchical systems, it is clearly observed that the number of rules required to design hierarchical systems is less than conventional systems that reduces the complexity of the overall system. The performance of a hierarchical system is better than conventional systems as the majority of error points of a hierarchical system lie in Class I category whereas most of the error points of a conventional system lie in between Class I and Class II. In other words, compared with a conventional system, the result of a hierarchical system has less deviation or error percentage from results of mathematical expression as in equation (9).

Table 8 shows the properties of all three approaches: mathematical expression described by equation (9), conventional fuzzy model and hierarchical fuzzy model. Table 8(a) presents the correlation among these approaches to see which one provides more accuracy. Similarly, Table 8(b) shows the properties of given approaches. Six parameters have been used to show properties: Standard deviation, variance, median, mean, maximum and minimum.

TABLE 8. Properties of various approaches.

S.No.	Expression and Conventional	Expression and Hierarchical	Hierarchical and Conventional
Correlation for Y_1	0.8215	0.9318	0.8255
Correlation for Y_2	0.8356	0.9648	0.8259

(a) Correlation between various approaches

Type	SD	VAR	MED	Mean	Max	Min
Expression	0.16	0.03	0.24	0.26	1	0.01
Conventional	0.07	0.01	0.27	0.27	0.4	0.16
Hierarchy	0.16	0.03	0.26	0.26	0.8	-0.13

(b) Properties of approaches

Where, 'Expression', 'Conventional' and 'Hierarchy' represents output 'y_1' and 'y_2' from mathematical equations, conventional and hierarchical fuzzy systems, respectively. 'SD', 'VAR', and 'MD' represents Standard deviation, Variance, and Median respectively. It is observed that hierarchical fuzzy systems have higher accuracy and are much closer to results from mathematical expression with the correlation $> 93\%$, whereas conventional fuzzy has correlation of 82% approx. Where the expected result represents results from mathematical expression presented in equation (9).

VIII. CONCLUSION

The large rule base, due to increase in inputs, becomes a bottleneck for using conventional fuzzy systems in complex real-life applications. To overcome these limitations, this paper presents a methodology to design a hierarchical fuzzy system. The multi-input single-output system has been implemented first and then the implementation has been extended to multi-input multi-output systems. The aggregated hierarchical tree structure has been considered for implementation. The algorithm can be used to design different hierarchical structures. This algorithm supports design of both Type-1 and Type-2 fuzzy logic systems. It is clearly demonstrated that the multi-input multi output hierarchical fuzzy system can be successfully designed by reducing the common sub-system terms without losing accuracy of the system. The current implementation has been done in the MATLAB and Simulink environment. The fuzzy clustering has been used to find desired clusters of input-output combinations. Several properties, such as standard deviation, variance, median, mean, max and min have been compared. It has been shown that the number of rules for a hierarchical fuzzy system has been reduced compared to a conventional fuzzy system. The proposed methodology will help researchers to use hierarchical fuzzy logic for multi-input multi-output systems and big data applications such as WSNs, IoTs etc.

AUTHORS' CONTRIBUTION

The major contribution in this article is the systematic and methodical development of the generalized algorithm for multi-input multi-output hierarchical systems. The algorithm reduces the duplicate common terms in the various sub-systems of multiple input-output systems. The approach used is similar to reducing common prime implicants in the simplification of multiple output Boolean functions. The generalized algorithm can be implemented using any of the hierarchical tree structures described previously by various researchers.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for the helpful suggestions to bring this paper in the present form. The authors are also thankful to Le Yi Wang and M. M. Gupta for helpful suggestions. The authors also acknowledge the financial support by Farshad Fotouhi and Mohammed Ismail Elnaggar.

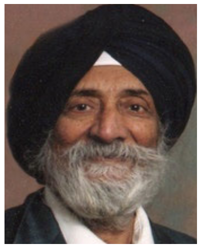
REFERENCES

[1] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.
[2] S. Kamthan, H. Singh, and T. Meitzler, "UAVs: On development of fuzzy model for categorization of countermeasures during threat assessment," in *Proc. 19th Unmanned Syst. Technol.*, May 2017, Art. no. 1019518.
[3] S. Kamthan, H. Singh, and T. Meitzler, "Survivability: A hierarchical fuzzy logic layered model for threat management of unmanned ground vehicles," *Auton. Syst., Sensors, Vehicles, Secur., Internet Everything*, vol. 10643, May 2018, Art. no. 106430W.
[4] R. Sindelar, "Hierarchical fuzzy systems," *IFAC Proc.*, vol. 38, no. 1, pp. 245–250, 2005.
[5] R. Scherer and L. Rutkowski, "A survey of hierarchical fuzzy systems," *Proc. 5th Conf. Neural Netw. Soft Comput.*, Zakopane, Poland, Jun. 2000 pp. 6–10.
[6] H. Singh, M. M. Gupta, T. Meitzler, Z.-G. Hou, K. K. Garg, A. M. Solo, and L. A. Zadeh, "Real-life applications of fuzzy logic," *Adv. Fuzzy Syst.*, vol. 2013, no. 2013, Art. no. 581879.
[7] Y. Lin and S. Chen, "A centroid auto-fused hierarchical fuzzy c-means clustering," *IEEE Trans. Fuzzy Syst.*, early access, Apr. 29, 2020, doi: 10.1109/TFUZZ.2020.2991306.
[8] R. M. Asl, R. Palm, H. Wu, and H. Handroos, "Fuzzy-based parameter optimization of adaptive unscented Kalman filter: Methodology and experimental validation," *IEEE Access*, vol. 8, pp. 54887–54904, 2020.
[9] F. Cantas, I. Kurt, and T. Mevlut, "Comparison of hierarchical and non-hierarchical fuzzy models with simulation and an application on hypertension data set," *Meandros Med. Dental J.*, vol. 19, no. 2, p. 138, 2018.
[10] H. Singh, S. Kamthan, A. M. Dixit, A. Mustapha, T. Meitzler, and A. Meitzler, "Fuzzy and neuro fuzzy techniques for crack detection in armor plates," in *Proc. MSV*, 2008, pp. 298–307.
[11] Y. Jarraya, S. Bouaziz, H. Hagras, and A. M. Alimi, "A multi-agent architecture for the design of hierarchical interval type-2 beta fuzzy system," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 6, pp. 1174–1188, Jun. 2019.

[12] L. Glass and M. Mackey, ''Mackey-glass equation,'' *Scholarpedia*, vol. 5, no. 3, p. 6908, 2010.
[13] L. Sun and W. Huo, ''Adaptive fuzzy control of spacecraft proximity operations using hierarchical fuzzy systems,'' *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 3, pp. 1629–1640, Jun. 2016.
[14] Y. Jarraya, S. Bouaziz, A. M. Alimi, and A. Abraham, ''Hierarchical fuzzy design by a multi-objective evolutionary hybrid approach,'' *Soft Comput.*, vol. 24, no. 5, pp. 3615–3630, Mar. 2020.
[15] D. N. Oliveira, A. P. de Souza Braga, and O. da Mota Almeida, ''Fuzzy logic controller implementation on a FPGA using VHDL,'' in *Proc. Annu. Meeting North Amer. Fuzzy Inf. Process. Soc.*, Toronto, ON, USA, Jul. 2010, pp. 1–6.
[16] T. Yamakawa, ''Hardware implementation of fuzzy logic controller-designability, stability and robustness,'' in *Control and Systems Engineering*, vol. 27. Cham, Switzerland: Springer, 2015, pp. 261–273.

SHASHANK KAMTHAN (Member, IEEE) received the B.E. degree from Rajiv Gandhi Proudyogiki Vishwavidyalaya, India, in 2007, and the M.Sc. degree in electrical engineering from Wayne State University, Detroit, MI, USA, in 2008, where he is currently pursuing the Ph.D. degree.

He is currently working as a full-time Senior Embedded Systems Consultant in Automotive Domain. On the professional front, he has over 11 years of professional experience. During his stint with automotive industry, he has worked in the field of model-based development, AUTOSAR, connected vehicle systems (V2X), embedded software development, and systems integration. He has more than 12 publications in international journals and conferences. His current areas of interest are target detection, artificial intelligence, connected vehicles, deep learning techniques, autonomous vehicles, mobility, controls etc.



HARPREET SINGH (Life Senior Member, IEEE) received the B.Sc.Eng. degree from Punjabi University, India, in 1963, and the M.S. and Ph.D. degrees from the University of Roorkee, India, in 1966 and 1971, respectively.

He taught at the University of Roorkee, from 1963 to 1981. Since 1981, he has been with Wayne State University, Detroit, MI, USA, where he has been a Professor with the Department of Electrical and Computer Engineering. He has worked in diversified areas of systems, networks, controls, fuzzy logic, DNA nano-digital circuits, bio-circuits, and VLSI design. He has more than 300 publications in international journals and conferences, and has received a number of awards. He has organized a number of national and international conferences and has supervised more than 25 Ph.D. theses in different areas of electrical and computer engineering. He has been awarded the Distinguished Alumni Award of prestigious IIT Roorkee, in 2012. He has also been awarded the Distinguished Alumni Award from the President of India at the Thapar Engineering College, Patiala, in 1983. As the Founder Chair of Life Affinity Group of South Eastern Michigan section, the section received the 2017 IEEE Life Member Affinity Group Achievement Award.