A review on soft computing techniques used in induction motor drive application

Gadwala Durgasukumar[1], Repana Ramanjan Prasad[1,2], Srinivasa Rao Gorantla[2]
[1]Department of Electrical and Electronics Engineering, Vignan Institute of Technology and Science, Hyderabad, India
[2]Department of Electrical and Electronics Engineering, VFSTR University, Guntur, India

Article Info

Article history:

Received Mar 5, 2023
Revised Oct 30, 2023
Accepted Nov 7, 2023

Keywords:

ANFIS
Artificial neural networks
Induction motor
Type-2 fuzzy
Type-2 neuro-fuzzy

ABSTRACT

This study reviews hybrid models built using fuzzy systems and neural networks. Expertise for induction motor drives, using the learning capacity of artificial neural networks, is an explicit representation of a fuzzy inference system. The effectiveness of neuro-fuzzy approaches for training and inference in induction motor drives has drawn the attention of researchers. This article gives an overview of several artificial neural network approaches, fuzzy, type-1 fuzz logic, type-2 fuzzy logic, neuro-fuzzy systems, type-1 neuro-fuzzy and type-2 neuro fuzzy systems in accordance with the classification of research articles. The major goal is to give a succinct summary of current neuro-fuzzy research so that readers can choose appropriate strategies based on their own research interests, such as various types of neuro fuzzy systems, to enhance system performance in general.

Corresponding Author:

Repana Ramanjan Prasad
Department of Electrical and Electronics Engineering, Vignan Institute of Technology and Science
Hyderabad 508284, India
Email: prasad243@gmail.com

1. INTRODUCTION

Most industrial and home applications use induction motors as their workhorse. The reliability, ease of installation, control, and adaptability of these motors make them popular for a variety of applications. In variable speed drive applications, the controller is crucial to ensure that the motor follows the reference trajectory without deviating significantly. Additionally, a controller that can respond quickly and handle uncertainties. It has been traditional in the industry to use proportional-integral (PI) and adaptive controllers with fixed gains. While these controllers can handle the uncertainty that is inherent to a nonlinear induction motor (IM), there are some disadvantages as well. Researchers have recently focused on applying soft computing techniques to control IMs regarding high-end variable-speed drives. The approximation of nonlinear dynamic systems by artificial neural networks (ANNs) has been proven to be universal. Due to a reduction in the controller's complexity, overshoot elimination, and a reduction in training time, the induction motor drive performance has been improved with ANN [1]. ANNs have shown beneficial for forecasting, modelling, and regulating complex, uncertain systems for which conventional techniques have been inadequate because of their learning adaption and nonlinear mapping capabilities [1]–[10]. Artificial neural network (ANN) and rapid artificial neural network (RANN) are used to control the speed of induction motor drives for reducing computational time with different algorithms has been presented [11]-[39]. Type-1 and type-2 fuzzy logic controller-based speed control schemes of induction motor drive has been presented [40]-[54]. However, when a higher degree of uncertainty is preset in the system, the type-1 fuzzy logic controller is unable to function effectively. Researchers have published several works over the past few decades on the application of neuro-fuzzy controllers for adjustable speed drives [55]-[71].

Zadeh [72] introduced fuzzy logic sets in 1978. A description with increased fuzziness is more likely to handle inexact information in a logically correct way, according to Hisdal in [73]. The development of fuzzy logic led to the creation of (FLSs), which have been applied to several systems modelling and control applications. First fuzzy logic controllers (FLCs) were presented by Mamdani and Baaklini [74]. In 1983, Fuji Electric adopted T1 FLSs for a water treatment plant, and in 1987, Hitachi implemented them in a subway system. It is important to consider the shape of a membership function (MF) in the design of T1 FLSs. T2 fuzzy sets (F2 FSs) were introduced by Zadeh [72] to incorporate uncertainty into fuzzy systems. Similarly, T2FSs have fuzzy membership grades. In most T2 FLS applications, this extra uncertainty adds an extra degree of freedom (DOF). To simplify T2 FLSs, interval type-2 fuzzy logic systems (IT2 FLS) were proposed due to their mathematical complexity. In this special IT2 FLS, the mathematics of T1 FLS are applied to a simpler version of T2 FLS. It differs greatly from a T1 FLS by replacing the DE fuzzifier block with an output processing block. A complete theory of type-2 fuzzy logic systems has been developed by previous researches [75]–[79]. Hybrid learning algorithm methodologies, three interval type-2 fuzzy neural network (IT2FNN) designs were presented. IF-THEN rules characterize these systems, but their antecedents or consequents are type-2. Uncertain information can be represented and handled effectively using a type-2 fuzzy set. Rule-based fuzzy logic systems (FLS) can be modelled and minimized by type-2 fuzzy sets [80]-[98]. This paper examines the methodologies and limitations of the ANN, RANN, fuzzy, type-1 FLC, type-1 IFLC, ANFIS, T1NFC and T2NFC controller-based induction motor drive performance of various researchers.

2. BRIEF ABOUT ARTIFICIAL NEURAL NETWORKS

ANNs have drawn scientists' interest due to their numerous advantages over traditional algorithmic techniques. Their benefits include being able to train, being easy to build, being able to estimate nonlinear functions, being able to endure network distortion, and being able to work without precise mathematical models. ANNs are helpful for finding and managing nonlinear systems because they can estimate a range of nonlinear functions to any required level of accurateness. In an AC motor application, speed or rotor resistance can be evaluated using stator voltages and currents as network inputs [8]–[10]. The indirect vector control strategy of the IM drive is employed using the ANN as the speed controller. Figure 1 illustrates how this system operates.

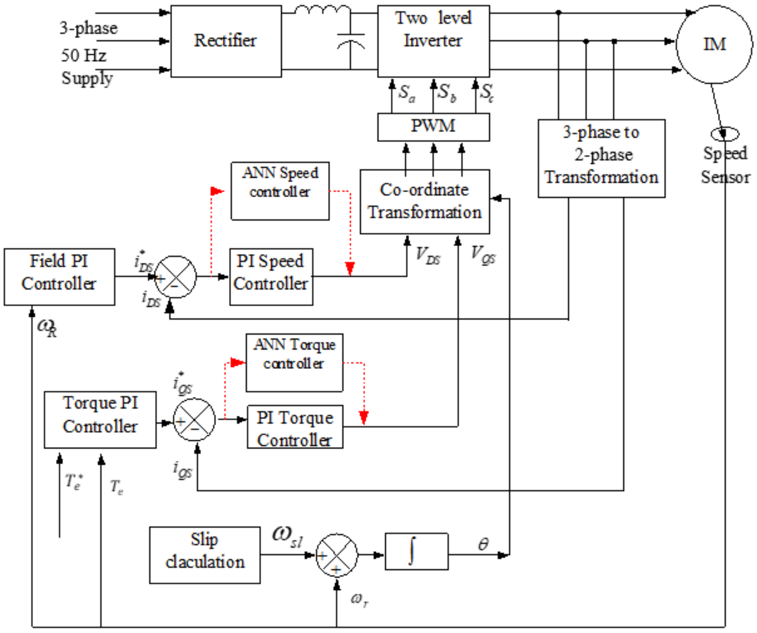

Figure 1. An ANN-based indirect vector control system

A reference command torque is calculated using phase current readings, as well as the desired command speed. An ANN-based speed controller is then fed the error speed and command speed. The ANN controller produces the suitable command torque for the i^*_{qs} component. Through a voltage source inverter, the correct i^*_{ds} component is determined from the flux command.

2.1. Artificial neural network architectures

The ANN structures employed in this study are shown in Figure 2, where I stand for an error, j for its integral, and o for a control tool signal. The neuron's inputs are represented by the symbols *I* and *j*, while its output is represented by the letter o. A nonlinear sigmoid function processes the combined inputs in hidden layer.

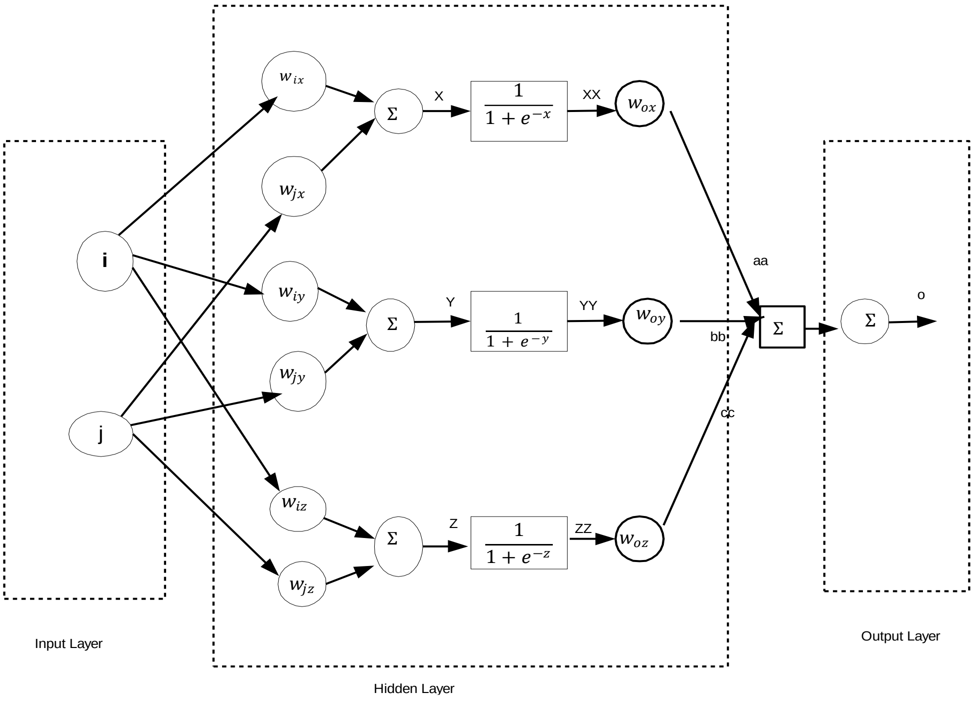

Figure 2. Artificial neural network architectures

Implementation of an ANN-based decoupling controller and flux estimator using MATLAB/ Simulink. As the error reductions, the convergence is very slow, hence the Levenberg-Marquart algorithm took 4500 iterations to reach an error of 0.001, and decoupling control took 105000 iterations to reach a value of 0.00105. At each epoch, a very extensive computation is performed, requiring a larger amount of computer memory [1]. An induction motor is controlled by fuzzy and neural algorithms [11]. The neural network is trained over 5000 epochs and tested over 500, which places a heavy computing demand on the system. In this article, only nominal speed conditions were taken into consideration for training the neural network. The fluctuation was therefore visible during low-speed operations. Small variations may be seen in the FVSC-based system from the simulation results. Due to the changed operating conditions, the neural network model prediction error in the NIMC first displays certain oscillations, which also have an impact on the torque response. The results in [12], it was suggested to utilize ANN to identify the induction motor parameters under a standstill scenario. The model error, which is only present at low frequencies, causes Resistance in series (Rs) and Inductance in series (Ls) to become increasingly inaccurate for frequencies higher than 1 rad/s. The rotor time constant reduces as frequency rises. A thorough parameter estimate method using neural networks was proposed [13]. Parameter estimation is challenging at slow speeds. In less than 0.1 seconds, the projected position can keep up with the real value. Although it is highly undesired for the estimate speed computation, the presence of ripples was found in the estimated position.

The speed management of an induction motor using a two-layered neural network and a neural network plant estimator with load torque observer has been described in [14]. It is challenging to accurately determine the system inertia J in this situation, which causes time constant variance. A hidden-layer recurrent artificial neural networks (RNN) adaptive-backstepping control system has been proposed in [15], and the neural network (NN) parameter-training algorithms are derived using the gradient-descent method. The global convergence of these parameters cannot be guaranteed by the gradient-descent method. The parameters convergence of the NN will be simple to achieve if learning rates are selected to be minimal but learning speed will be slow as a result [16]. However, learning will proceed quickly if and are set to be large. For induction motor drives, a novel unit to evaluate speed and rotor resistance has been introduced [17], [18]. This component only works with single output systems because it is based on the new adaptive linear neuron (ADALINE) structure. The correct numerical number for a learning rate, however, may only be determined by trial and error.

It was suggested in [19] to identify electromagnetic torque using supervised neural networks. The analogous circuit parameters, for example, are certain variables that are typically unknown under normal working conditions and must be known to understand electromagnetic torque. Te MLP networks with a single hidden layer have been used to recognize the model. It has been demonstrated to calculate induction motor parameters using neural networks [20]. They do not consider how the motor model would operate in the presence of harmonics and unbalanced voltages in this. The production cost that should be trained on a specific manufacturing technology facility might also be evaluated using a cascaded artificial neural network [21]. A unique method of rotor position detection without sensors has been described [22]. The projected rate of inductance change at low speeds was severely decreased by simulation results. A self-tuning speed controller built on RNN was proposed [23]. RNN structure, however, only has one input and one output. The number of inputs and outputs to the ANN are the only fixed parameters during design and training. Furthermore, it is commonly acknowledged that learning any arbitrary nonlinearity only requires a maximum of two hidden layers. However, equally important characteristics are the number of hidden neurons, learning rate, momentum gain, threshold value, and training patterns.

Recurrent neural network topologies for estimating the parameters of dynamical system state-space models are given in [24]. However, analogue multipliers and dividers, which are more expensive, are used when the RNN design is implemented. For efficiency optimization, neural network-based rotor flux estimators have been developed [25]. The 4 and 3 input neural network-based rotor flux estimator was not improved by the mutual inductance, according to the simulation results. A new flux estimator based on neural networks was proposed [26]. According to the experiment's findings, torque and flux have very high levels of ripple content, which has an impact on drive. Online calculation of Rotor resistance by considering fluctuations in stator resistance is done in [27] using a hybrid method of NN and fuzzy logic. In this methodology, the other NN that is introduced in parallel estimates the stator resistance individually. As a result, stator resistance variations have no effect on rotor resistance estimation. These two parallel NNs must be implemented, which necessitates complex calculations. The rotor resistance estimation methods based on NN have a transient inaccuracy that cannot be ignored, according to the dynamical research of NN. In the slip frequency type of vector control, where the rotor flux is constant, it is difficult to estimate the rotor resistance and speed at the same time. But in the transient condition of the speed, the rotor flux is not always constant. At approximately 70% of the rated rotor speed, the results are attained. As a result, at low and zero speeds, no results are reported. In [28], artificial neural networks are used to estimate the induction motor parameters. Rotor resistance and mutual inductance are the estimated values. A feed-forward neural network can approximate any continuous function, according to NN architecture.

A speed estimate approach for an induction motor based on a multilayer NN with a single hidden layer was developed [29]. The number of nodes has a significant impact on how long the NN takes to compute. The estimation performance may be enhanced by adding more nodes. When the number of neurons is increased, the current control period must also be extended, which lowers system performance. The parameters, which have been found by trial-and-error methodology, include the learning rate, momentum constant, and slope of the sigmoid function. Depending on the speed, these parameters need to be adjusted. However, figuring out the parameters for each speed is a very difficult task. The idea of ANN-based parameter estimation, which calls for large computations and intricate processes [30]. An ANN-based reactive power-based model reference adaptive system (Q-MRAS) for improving induction motor drive stability has been presented [31]. To make the model reference adaptive system (MRAS) stable in the regenerative mode, an ANN is employed. However, using a neural network takes training, and there is no standard technique for selecting neural networks.

However, the ANN equations to approximate nonlinear systems are not derived by the established methodology but rather by the empirical formula and the trial-and-error approach in most cases. ANN base sensor-less control of induction motor was presented in [32]. A lot of training data and complicated calculations are also required for ANN. A voltage-controlled-fed IM drive system's thyristor firing angle selection using an ANN was proposed in [33]. However, accuracy is only applicable given input data that falls within the parameters set during ANN model training. Beyond such limits, the precision might not be as excellent as anticipated, and there is no feedback or estimate of the torque, speed, and other characteristics that are ultimately developed. A model for adaptive control of five-phase interior permanent magnet motor drives was proposed in [34]. The processing speed and sampling time both have an upper limit on the neural network's number of neurons. More neurons in the neural network make training simpler and faster.

A neural network-based MRAS speed observer technique has been presented in [35]. Neural networks suffer from a lack of design methods. This level of trial and error could extend the training period. A cascading neural network-based flux estimator for induction motor drives speed prediction has been introduced in [36]. The NN-based flux estimator is modelled using the single neuron cascade forward neural network (SNCF-

NN), although this network has a sophisticated multilayer structure. An ANN in [37] uses different operating conditions to identify thermal changes in the stator resistance. When this ANN open-loop model is paired with the MRAS observer, better low speed performance is demonstrated. A 3 level voltage-fed inverter with SVM implemented using a neural network was proposed in [38]. The choice of these switching states has a considerable impact on the inverter's performance. The complexity of space vector pulse width modulation (SVPWM) for multilevel inverters increases as level n increases due to an increase in the number of triangles, switching states, and on-time calculation. With a 600 MHz Pentium-based PC, training typically took half a day, and it required 12,000 epochs to achieve SSE (sum of squared error) 0.008. Be advised that the ANN functions at a higher resolution due to learning or interpolation capabilities. The pulse width signals are established at the output by solving the network at each sample interval. The inverter's low switching frequency contributes to the machine's high ripple torque. It was suggested in [39] to use ANN-based Q-MRAS that is trained using the Bayesian regularization technique. Performance was enhanced at zero and extremely low speeds and lightly loaded situations in all four operating quadrants, according to ANN-based Q-MRAS.

3. ABOUT FUZZY BASED SYSTEM

By employing membership functions with fuzzy rather than crisp membership, type-2 fuzzy systems differ from type-1 fuzzy systems. Although it is expected that membership in type-1 can be stated as a distinct numerical value, this is not always the case. Although the membership values of universal or interval type-2 fuzzy sets are thought to be unclear. In order to describe uncertainty in actual situations, the idea of the "footprint of uncertainty" is developed.

3.1. Type-1 fuzzy logic related sets values

Type-1 related set A values can be expressed as $A = \{(x, \mu_A(x))| \quad \forall_x \in X\}$. A can also be said to be $A = \int_{x \in X} \mu_A(x) \,/\, x$. A two-dimensional function known as a type-1 Gaussian membership function, A(x), is ensured to have a value between 0 and 1 for all X., as shown in Figure 3. There is no doubt in this kind of membership function. In other words, each input data point has a distinct membership value.

3.2. Type-2 fuzzy logic related sets values

Fuzzy sets related to type 2 is expressed as [82]:

$$\check{A} = \{((x,u), \mu_{\tilde{A}}(x,u))| \ \forall_x \in X \ \ \forall_\mu \in J_x \ \subseteq [0,1]\}$$

Here $\mu_{\tilde{A}}(x,u)$ = Fuzzy membership function of type-2 and lies $0 \leq \mu_{\tilde{A}}(x,u) \leq 1$ $\check{A}$ [76] may even expressed as

$$\check{A} = \int_{x \in X} \int_{u \in X} \mu_{\tilde{A}}(x,u)/(x,u) \quad J_x \subseteq [0,1]$$

In addition to a secondary membership value that reflects the probability of primary memberships, each primary membership value also has 1 [40]. The secondary membership functions in interval T2FLSs are uniform functions that only accept values of 1, as opposed to extended T2FLSs, which allow values in the range [0,1]. Since the computations for general T2FLSs are highly difficult, interval T2FLSs are more frequently used in the literature since they are easier to handle. The placements of the membership functions may not be accurately defined if the conditions are so unclear. In situations where the membership grade isn't able to be stated as a precise number between 0 and 1, type-2 fuzzy sets may be a more advantageous choice. Figure 4 can be produced by reducing the standard deviation of the Gaussian function in Figure 3. The membership function in Figure 4 does not have a single value for any given x value. It is not necessary to weight equally all the values at which the vertical line intersects the membership functions. Furthermore, each of those points can be given an amplitude distribution shown in Figure 4.

3.3. Type-2 fuzzy logic system block diagram

Figures 5 and 6 representing the type 1 and 2 related block diagrams of the FLC systems, the main difference between these two is that type reduced block. And remain all are same for the both T1FLC and T2FLC. The Takagi-Sugeno model utilizes the input variables' function in the subsequent part rather than fuzzy sets (like in Mamdani models). The model's order is determined by the function's order; examples include the zeroth order TSK model. Fuzzy IF-THEN rules can be described a type-1 TSK model from (1)-(11).

$$\text{If } x_1 \text{ is } A_{j1} \text{ and } x_2 \text{ is } A_{j2} \text{ and} \ldots\ldots \text{ and } x_n \text{ is } A_{jn} \tag{1}$$

$$\text{Then } u_j = \sum_{i=1}^{n} w_{ij} x_i + b_j \tag{2}$$

The system's ultimate output can be expressed as (3), where f_j is given by (4).

$$u = \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \quad (3)$$

$$f_j = \mu_{A_{j1}}(x_1) * \ldots..* \mu_{A_{jn}}(x_n)\backslash \quad (4)$$

The rule basis, in a first-order type-2 TSK model, for example, is:

IF x_1 is $\widetilde{A_{j1}}$ and x_2 is $\widetilde{A_{j2}}$ and …….and x_n is $\widetilde{A_{jn}}$ (5)

Then $u_j = \sum_{i=1}^{n} w_{ij} x_i + b_j$ (6)

$$U(F_1, \ldots \ldots . F_M) = \int_{f_1} \ldots \ldots . \int_{f_M} T_{j=1}^{M} \mu F_j / \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \quad (7)$$

$$Y_{TSK/A2-C0} = \int_{f^1 \in [\underline{f^1}, \overline{f^1}]} \ldots \ldots \ldots .. \int_{f^M \in [\underline{f}^M, \overline{f}^M]} 1 / \frac{\sum_{j=1}^{M} f_j u_j}{\sum_{j=1}^{M} f_j} \quad (8)$$

$$\underline{f_j}(x) = \underline{\mu}_{Aj1}(x_1) * \ldots \ldots \ldots * \underline{\mu}_{Ajn}(x_n) \quad (9)$$

$$\overline{f}_j(x) = \overline{\mu}_{Aj1}(x_1) * \ldots \ldots \ldots ..* \overline{\mu}_{Ajn}(x_n) \quad (10)$$

$$Y_{TSK\,/A2-C0} = \frac{\sum_{j=1}^{M} \underline{f_j} u_j}{\sum_{j=1}^{M} \underline{f_j} + \sum_{j=1}^{M} \overline{f}_j} + \frac{\sum_{j=1}^{M} \overline{f}_j u_j}{\sum_{j=1}^{M} \underline{f_j} + \sum_{j=1}^{M} \overline{f}_j} \quad (11)$$

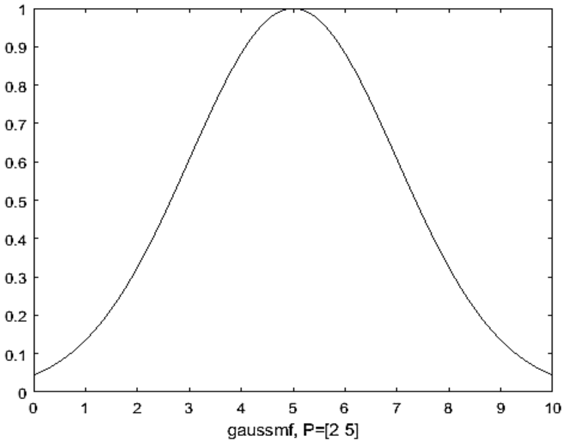

Figure 3. Representation fuzzy membership function with a Gaussian type-1

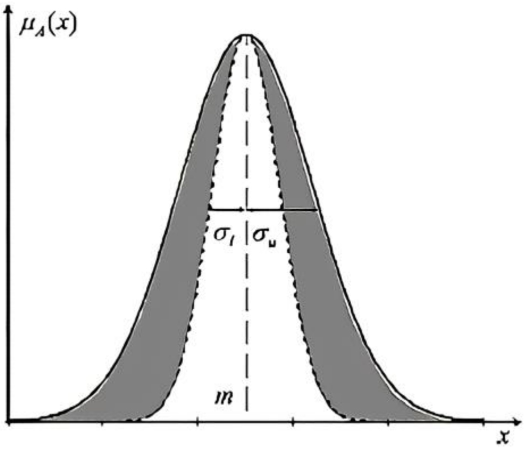

Figure 4. Type-2 fuzzy membership function for Gaussians (FOU)

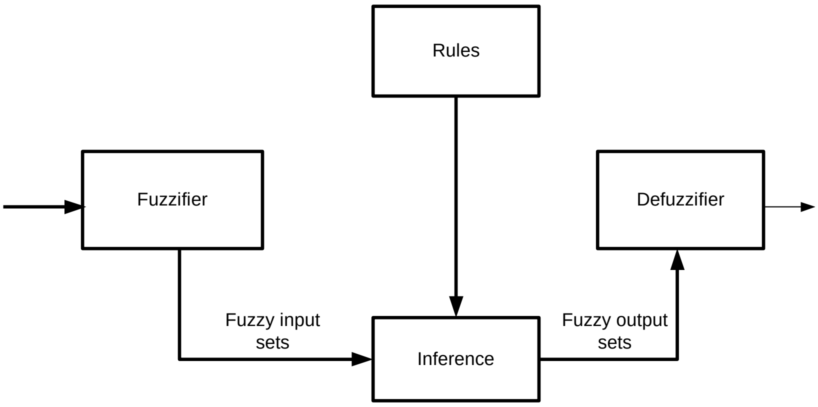

Figure 5. Block diagram representation of type-1 FLC

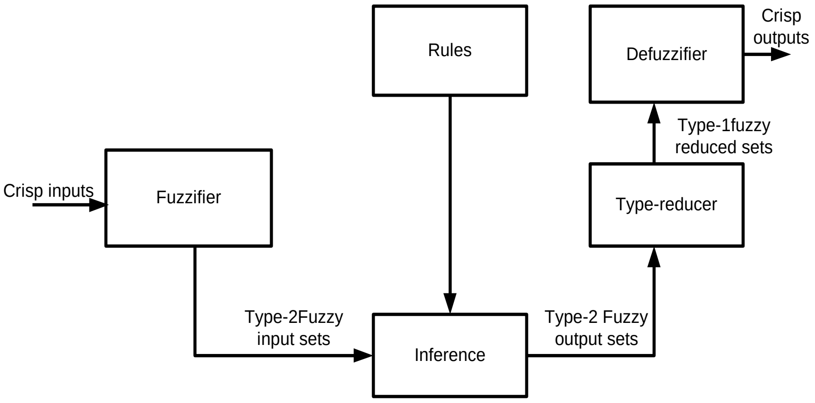

Figure 6. Block diagram representation of type-2 FLC

It has been suggested to use an embedded-based fuzzy system [41], but doing so would require a lot of memory, which might push up project costs. If used with real-time hardware, the computational cost might also go higher, and real-time performance might decrease. Due to its exponential calculus, the Gaussian function used in the suggested method demands processing of significantly more instructions than linear functions do in current methods. The controller's response time was quicker with a higher frequency, but the motor speed tended to oscillate more and/or overshoot considerably. The usage of an embedded-based fuzzy system has been proposed [42] but doing so would require a significant amount of memory, which might increase project expenses. The computational cost may increase, and real-time performance may decline if real-time hardware is used. The Gaussian function utilized in the recommended method requires processing of much more instructions than do linear functions in existing methods because to its exponential calculus. With a higher frequency, the controller responded more quickly, but the motor speed tended to oscillate more and/or overshoot significantly.

The design, simplicity, and operation of FLC for IM drive speed control were examined [8]. Despite having higher performance advantages, the FLC confronts a significant difficulty due to a high computational burden and a high memory space demand, particularly for real-time implementation. Induction motor torque control with multiple objectives and fuzzy prediction is described in [43]. An experimental motor drive test bench has been used to evaluate the suggested FPTC approach. A very basic stator flux and torque estimator's performance is impacted by changes in the stator and rotor time constants. The stator resistance change has a significant impact on the flux observer and causes some oscillations in the torque and flux responses. This problem impacts all model-based techniques, but it is especially prevalent in suggested FPTC, where accurate parameter knowledge is necessary. The optimal global minimum solution and the consistency of algorithm performance are both assessed using all these benchmark functions [44]. The non-separable, low, and high dimensional functions may cause some issues, too. Since lightning is a natural phenomenon, the QLSAF offers a faster convergence rate for solutions than other traditional optimization methods.

A hybrid duty ratio control (HDRC) technique using interval type-2 fuzzy-based DTC (IT2FDTC) has been suggested [50]. The rules of the type-2 fuzzy interval is designed to have a quick settling period and little IMD speed/capacitor voltage overshoots. In general, humanoid specialists robust these rules using a pre-learned technique, meaning that the person trained while learning the IM performance in each mode of operation and modified the rules of MFs accordingly. Induction motor torque management using fuzzy logic was suggested in [45]. However, choosing a voltage vector for the full switching period results in a significant current distortion and very high torque and flux ripples since the torque cannot be accurately regulated. Only the low-speed zone was used for the validation of simulation and experiment results [46]. This research did not use systematic algorithms to come to their conclusions; instead, they used uncertain approaches to choose the dominant rules. To estimate the variations in induction motor stator resistance caused by temperature fluctuations, a fuzzy based resistance estimator has been reported in [47]. It is clear from the results of the two estimators that the PI-based resistance estimator does not provide as good tracking as the fuzzy logic-based estimator does. But it was discovered that the fuzzy resistance estimator had issues with low torque levels (below 2 Nm) and high command flux. Because of the erratic resistance variation, the controller generates incorrect torque and flux. In [48], a new type-2 membership function was put forth to analyze noise reduction. T2FLSs should only be chosen, though, if the system under consideration has a high amount of noise and several uncertainties.

By choosing the most efficient voltage vector, the control techniques may balance the TDCI's dc-link capacitor voltages without the use of an additional controller. Additionally, it takes longer to reach the steady state when capacitor voltages have big spikes caused by changes in load or speed [49]. This presents a so-called "new look" at type-2 fuzzy sets (T2 FSs) and systems and asserts, quite boldly, that the new view is better than the old one. A thorough examination of the connections between this unique representation of a T2 FS and (at the absolute least) the well-known T1, blur, and weight representations is not present in [50], nor is there any discussion of expanding the new representation from one to multiple MF parameters.. Numerous academic works have shown the usefulness of EKF, and several new, improved versions have also been suggested [51]. However, the arbitrary distributions in the system under consideration present challenges for these nonlinear filters. The use of type-2 Fuzzy Classifiers in EEG Analysis for Driving Cognitive Failure Detection has been presented in [52]. It requires additional complexity for secondary membership evaluation for the GT2FS-based classifier in addition to taking the product of the main and secondary MFs at the specified measurement points. However, compared to its IT2FS equivalent, the execution of the product functions and the time needed for secondary membership computation add additional complexity. A general type-2 fuzzy PI controller (zT2-FPI) based on zSlices has been proposed in research [53]. To training IT2 TSK FLSs, the fast-training algorithm T2FELA based on extreme learning strategies is suggested [54]. The suggested T2FELA method, however, enables quick learning of the parameters for the consequents and random production of the preceding' parameters. The type-2 Sugeno fuzzy logic system with subtractive grouping is introduced in Khanesar *et al.* [55]. Type-2 TSK FLSs, on the other hand, have more design parameters for each rule and are difficult to detect than type-1 TSK FLSs.

4. NEURO-FUZZY BASED SYSTEM CONFIGURATION

Building more intelligent decision-making systems is possible thanks to neuro-fuzzy computing [56], which combines the advantages of neural and fuzzy techniques. This integrates into the system the general benefits of artificial neural networks, such as huge parallelism, robustness, and learning in situations with lots of data. Fuzzy logic allows for the modelling of qualitative and imprecise knowledge as well as the transmission of uncertainty. The neuro-fuzzy technique offers the corresponding application-specific benefits in addition to these general benefits. T1NFCs are well renowned for being resistant to changes in parameters and noise, making them an ideal solution to handle induction motor uncertainties and load variations. The T1NFC architecture design, represented in Figure 7. The T1NFC procedure is described from (12) to (20)

$$input^1 = e_\omega = {\omega_r}^* - \omega_r \tag{12}$$

$$input^2 = \Delta e_\omega = e_\omega(k) - e_\omega(k-1) \tag{13}$$

Here $y_1, y_2, \ldots, y_n$ can be given in generalized form as

Rule j (j=1,2--): if e_ω is m_j AND Δe_ω *is* n_j then $y_j is \sum_{j=1}^{A} m_j e_w + n_j \Delta e_w + r_j$

- Layer I: Input layer consists of node member ship functions

$$o_j{}^1 = A_{mj}(e_\omega), \text{ j=1,2} \tag{14}$$

$$o_j{}^1 = A_{nj}(\Delta e_\omega) \text{ j=1} \tag{15}$$

Where A_{mj1} and A_{mj2} expressed as:

$$A_{mj} = e^{\left(-0.5\left(\frac{(m_j - x)^2}{\sigma^2}\right)\right)}$$

$$A_{nj} = e^{\left(-0.5\left(\frac{(n_j - x)^2}{\sigma^2}\right)\right)} \tag{16}$$

- Layer 2: The firing strength of a rule is determined at this output node.

$$o_j^{\,2} = w_i = A_{mj}(e_\omega).A_{nj}(\Delta e_\omega)$$
$$= min\big(A_{mj}(e_\omega), A_{nj}(\Delta e_\omega)\big), j = 1,2.....7 \quad (17)$$

- Layer 3: Each node in this layer computes the weight, which is normalized.

$$o_j^{\,3} = \overline{w_j} = \frac{w_j}{w_1+w_2}, j = 1,2 \quad (18)$$

- Layer 4: Every node in Layer 4's De Fuzzification Layer has a node function that is provided by:

$$y_j\, is \sum_{j=1}^{A} m_j e_w + n_j \Delta e_w + r_j \quad (19)$$

- Layer 5: It is referred to as an output layer since it just has one node that generates the entire output, which contains the weighted sum of all the combined outputs of the preceding layers. The output is then given a:

$$o_j^{\,5} = \frac{\Sigma w_j u_j}{\Sigma w_j}, j = 1,2.....7 \quad (20)$$

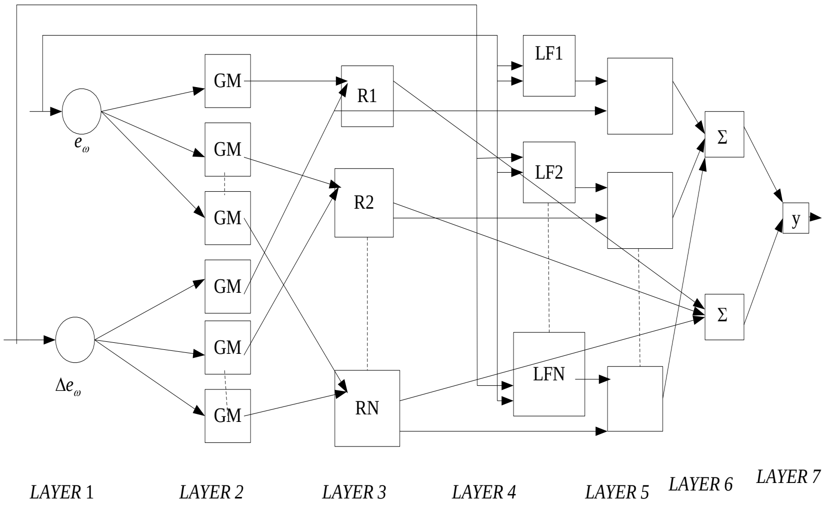

Figure 7. Type-1 NFC architecture related representation

4.1. T2NFC are characterized by fuzzy IF-THEN rules

Type-2 fuzzy values are present in the parameters of the antecedent and consequent parts of the rules. The fuzzy ruleset of the suggested system is expressed in equations from (21) to (27).

IF e_T is m_{1j} AND Δe_T is m_{2j}
THEN $y_j = \sum_{i=1}^{2} w_{ij} x_i + b_j$
Where $x_1 = e_T$, $x_2 = \Delta e_T$ are the inputs
$y_j ism_{1j} e_T + m_{2j} \Delta e_T + b_j$

- Layer 1: This layer's nodes each function as precise input variables. This layer is only fed input variables. Keep in mind that this layer has no weights that need to be changed.
- Layer 2: This layer consists of node membership functions. The upper and lower membership functions degrees, along with an undetermined standard deviation, define the range.

$$o_j^{\,2} = \bar{\mu}_{m1j} = e^{\left(-\frac{1}{2}\left(\frac{(x_j-c)^2}{\bar{\sigma}^2}\right)\right)} \text{ j=1,2 ......7}$$
$$o_j^{\,2} = \bar{\mu}_{m2j} = e^{\left(-\frac{1}{2}\left(\frac{(x_j-c)^2}{\bar{\sigma}^2}\right)\right)} \text{ j=1,2 ...... 7}$$

$$o_j{}^2 = \underline{\mu}_{m1j} = e^{\left(-\frac{1}{2}\left(\frac{\left(x_j-c\right)^2}{\underline{\sigma}^2}\right)\right)} \text{ j=1,2 ...... 7}$$

$$o_j{}^2 = \underline{\mu}_{m2j} = e^{\left(-\frac{1}{2}\left(\frac{\left(x_j-c\right)^2}{\underline{\sigma}^2}\right)\right)} \text{ j=1,2 ...... 7} \qquad (21)$$

- Layer 3: Every node in this layer calculates the firing strength of a rule with the least error or least change in error between any two input weights using the prod t-norm operator.

$$o_j^3 = \overline{w_\iota} = \bar{\mu}_{m1j}(e_T)\bar{\mu}_{m2j}(\Delta e_T) \qquad (22)$$

$$o_j{}^3 = \underline{w_i} = \underline{\mu}_{m1j}(e_T)\ \underline{\mu}_{m2j}(\Delta e_T) \qquad (23)$$

$$\overline{W_\iota} = \frac{\overline{w_\iota}}{\sum_{i=1}^{M}\overline{w_\iota}}\ and\ , \underline{W_i} = \frac{w_i}{\sum_{i=1}^{M}\underline{w_i}} \qquad (24)$$

- Layer 4: The outputs of the linear functions in the subsequent parts for the two inputs are in this layer.

$$o_j{}^4 = y_j = m_{1j}e_T + m_{2j}\Delta e_T + b_j \qquad (25)$$

- Layer 5: This layer calculates the product of the linear functions and the membership degrees ($\underline{W_i}$) and ($\overline{W_\iota}$).

$$o_j{}^5 = \ y_j \ = q\sum_{i=1}^{M} y_j\underline{W_i} + (1-q)\sum_{i=1}^{M} y_j\overline{W_\iota} \qquad (26)$$

- Layer 6: This layer contains two summation blocks. One of these blocks calculates the layer's output signal sum, and the other block computes the layer's layer 3 output signal sum.
- Layer 7: The output can be determined here as:

$$u = \frac{q\sum_{i=1}^{M} y_j\underline{W_i}}{\sum_{i=1}^{M}\underline{W_i}} + \frac{(1-q)\sum_{i=1}^{M} y_j\overline{W_\iota}}{\sum_{i=1}^{M}\overline{W_\iota}} \qquad (27)$$

The preceding approach, unfortunately, came into issues because it needed a lot of data, took a while to train, and required a lot of memory to implement in real-time for both linear and nonlinear functions. It has been suggested in [57], [58] to use an interval type-2 mutual subset hood fuzzy neural inference system (IT2MSFuNIS). However, it addresses issues with time-series prediction, function approximation, and control. A self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) based on the Takagi-Sugeno-Kang (TSK) model was suggested [59]. The model generated in this study is entirely online, nevertheless, because data normalization requires upper and lower boundaries, such as those between [-1, 1]. This model's lack of rule management mechanisms is a further problem. As a result, this model is unable to handle non-stationary growth due to a lack of rule reduction modules [60].

The ANFIS model is a reproach to training-generated adaptive fuzzy systems. Recent years have seen some interesting research on fuzzy systems [88], [89]. When the ANFIS model is used to solve problems in the real world, training the model's parameters is one of the major problems that arise. The majority of the ANFIS training methods are based on gradient descent (GD) approaches, where the gradient computation in each step is tractable since the chain rule applied may result in numerous local minima of the issue. The neuro-fuzzy model based on type-2 fuzzy sets was introduced in [61] as a novel approach for regulating a nonlinear system.

A type-2 singleton fuzzifier-based neuro-fuzzy controller for mobile robot navigation was proposed [62]. This type of fuzzifier, nevertheless, might not always be sufficient, particularly when there are load disturbances and induction motor parameter changes. In this research [63], the computation is slow and requires a lot of learning time in addition to the membership function having recurrent terms.

Type-II membership functions were introduced in the prior part of this study [64], and wavelets were included in the subsequent portion to further improve convergence. The challenge of selecting the mother wavelet function for WNN is one of its main limitations. Wavelet functions are created using a mother wavelet function and several fundamental transformations. Not all functions may be employed as wavelet mothers; a wavelet function needs to meet several requirements to be available and eventually mature into a decent wavelet transform function. For the BRB detection in a three-phase IM, [65] a general neuro-fuzzy model-based fault detection technique. The general neuro-fuzzy model and the custom threshold levels make up the fault detector. According to the discrepancy between the generic model's output and an actual torque-speed connection, variable thresholds were chosen. These are then utilized to take machine variations into consideration. As less experimental data is required to construct the fault detector, this strategy solves a practical drawback of model-based approaches. High performance vector-controlled motors have successfully used constant parameter fuzzy logic (CPFL) controllers [66]. The error and change of error are the two inputs for a constant parameter fuzzy logic (CPFL) controller. The torque current command produced by the CPFL controller must be changed in accordance with the speed error and change in speed error. A CPFL approach that is off-line optimized has been employed by many researchers [67].

A method for creating a type-2 neural-fuzzy system from an input-output set was proposed [96]. The dataset is divided into clusters using a fuzzy clustering algorithm. Then, from each cluster, a type-2 fuzzy TSK rule is derived. Because there are fewer membership functions in this manner, the output accuracy of the control is decreased. With simplified regulations, the fundamental problem is that the system's performance suffers at slower and reverse working speeds [68]. In motor applications including speed estimation, harmonics reduction, and torque ripple minimization, adaptive neuro-fuzzy inference systems have been functioning satisfactorily. However, the AI-based controllers have limitations in terms of the significant data requirements, extended learning, and lengthy training periods [69]. The fuzzy logic speed controller-based optimization strategies were suggested in [70] to improve the scalar control and vector control for an IM drive. However, the appropriate architecture, the ideal number of membership functions (MFs), and the proper creation of fuzzy rules all contribute to FLC's correctness. The type-2 fuzzy set is incorporated in either the antecedent or the consequent part of the type-2 neuro-fuzzy system, or both. The selection of the best structure and parameter identification are the main obstacles in the creation of type-2 neuro-fuzzy systems. Derivative, derivative-free, or hybrid training algorithms will be used [71]. The interval type-2 FNN (IT2FNN) parameters are tuned using a sliding mode incremental learning algorithm in [72], where an adaptive learning rate with an adaptation rule is generated. The continuous nature of the adaptation laws suggested in this study is one aspect that needs to be considered. However, an ideal sampling time should be selected for the method's computer simulation. An extremely large value for the sample time could result in system instability, making the selection of the ideal sampling time problematic.

Many researchers working in the drives field have used soft computational techniques, including artificial neural networks, fuzzy logic, and neuro-fuzzy [74]. These techniques are well-known intelligent control techniques. However, much work needs to be done in order to mature the basic technologies as well as the drives sector. The use of type-2 fuzzy logic control has seen increasing interest due to the need to account for uncertainty. [75] developed interval type-2 fuzzy for path planning and control with obstacle detection. The gradient-based technique performs well when the system under study exhibits very slow dynamics fluctuations. However, because partial derivatives are used in gradient-based algorithms (such dynamic backpropagation), the speed of convergence may be slow, especially when the search space is complicated [72]. The study of ANFIS techniques for use in various power system issues has expanded significantly. This research presents a power system stabilizer (PSS) based on a fuzzy basis function network (FBFN), which has been published in [78] to enhance power system dynamic stability. Each ANFIS unit identifies one failure mode when the numerous Adaptive Neuro-fuzzy Inference System is employed for fault diagnosis [77]. Various learning strategies have been described in the literature for the automatic construction of fuzzy sets.

5. GRADIENT DESCENT-BASED LEARNING ALGORITHMS

5.1. Back-propagation algorithms

The type-2 FNN structure has been described [78], and the gradient descent approach is used to derive the structure's parameter update rules. Even though the type-2 FNN has fewer rules than the type-1 FNN, it performs better overall. The system's disadvantage is that because the algorithm's convergence values are chosen to be minimal, learning proceeds at a very slow rate. To execute the steepest-descent technique and fine-tune the parameters of T2FLSs have been presented [79]. However, its implementation is based on four assumptions. 1. Neither parameters nor rules are shared. 2. The antecedent and consequent MFs' formulas are not known in advance. 3. Using mathematical formulas, derivatives required for a steepest descent tuning technique must be calculated. 4. Type-reduction focused on sets is employed. For the antecedent portion of the interval T2FNN, the dynamical optimal training algorithm and genetic algorithm were integrated to figure out the best spread and learning rate. The weighting factors in the subsequent phase of the T2FNN as well as the parameters of the antecedent type-2 MFs were tuned using these equations. As a result, these equations could be inaccurate and produce false results. As a result, it may be necessary to repeat any subsequent comparisons that were inaccurate [80].

5.2. Levenberg-Marquart algorithm

The Levenberg-Marquardt algorithm-based T2FNN was proposed in [81]. The technique makes use of second order derivatives to speed up training. There was also a discussion of a direct way for computing the Jacobian matrix, which is the trickiest part of applying the Levenberg-Marquardt algorithm. The scalar is crucial when updating the Levenberg-Marquardt algorithm's parameter rules. The weight equations change to gradient descent, which is sluggish learning, if the scalar is sufficiently large. The LM is a lot quicker than the GD algorithm, which is the foundation of the conventional BP method. For type-2 fuzzy systems, a new diamond-shaped membership function has been presented [82]. A few definite values between 0 and 1 and several ambiguous values make up the suggested membership function. It has been demonstrated that the lower membership function must be considered with lower grades to achieve a superior noise reduction property.

5.3. Kalman filter-based algorithm

Only a 1.2% improvement in identification rate was seen when nine hidden neurons were used. It should be noted that the identification rate for the traditional RBF approach increases to the fuzzy situation but does not surpass it when the number of hidden neurons is further increased [83]. For robot manipulators powered by artificial muscles, a novel reliable method known as radial base function network type 2 fuzzy sliding mode control (RBFT2FSMC) has been developed [84]. However, there are several drawbacks to using this architecture. At first, as the number of links to be regulated increases, so does the complexity and unpredictability of the MIMO RBFNN dynamics. Second, as the number of joints grows, computation time will become increasingly crucial. A heat exchange procedure on the apparatus CE117 process trainer was subjected to a modified interval type-2 fuzzy T-S modelling method that was suggested in [85]. Because of the trade-off between a sufficient level of accuracy and the cost of calculation, the number of fuzzy rules should be carefully selected. A comprehensive discussion of a few of these T2FLS optimization techniques can be found in [86]. A comparison of bio-inspired algorithms used for T1, and T2 fuzzy logic controller (FLC) optimization was provided [87]. A genetic algorithm-based strategy for designing a type-2 FLS was made in [88].

Interval type-2 fuzzy logic controllers' genetic learning and performance evaluation have been given in [89]. It was found that the type-2 FLC's control surface is more complex. In comparison to the other 3 controllers, FLC2 has a larger computational cost. Nevertheless, real-time implementation is limited by the requirement for expensive computing. Interval type-2 fuzzy neural networks (IT2FNN) design approach and real-coded genetic algorithm optimization of the network have been reported [63]. However, in the event of higher dimensional data, the network design may run into problems. A description of an optimization technique based on the degree of uncertainty for the membership functions of type-2 fuzzy systems are provided [90]. The proposed design approach for type-2 fuzzy models aims to take full advantage of the membership function uncertainty. However, in more difficult cases, a longer search procedure would be necessary for optimization, and additional research might be required to improve the effectiveness of the design process.

5.4. Artificial Bee Colony

The Bee Colony Optimization algorithm (BCO) includes a new method for designing type-1 and type-2 fuzzy controllers [91]. However, the results show that as the number of follower bees in the search space increases and there are more iterations, the computing time increases. Utilizing the simulation of annealing (SA) described in [92], an optimal design of IT2FLS was provided. By minimizing the objective function, the parameters of the consequent components of the IT2FLS were optimized using SA. The Mackey Glass time series was then predicted using the optimized model by determining the ideal IT2FLS configuration. With the help of an IT2FLS, a universal T2FLS was created, utilizing the SA algorithm [93]. The proposal's main goal was to minimize the calculations required to obtain the best FOU with IT2FLS. However, in certain cases, the accuracy losses in the conversion step exceeded these gains from SMF learning, producing outcomes that were comparable to those of the IT2FLS.

The continual nature of the adaptation laws suggested in this study is one aspect that needs to be considered. However, an ideal sampling time should be selected for the method's computer simulation. An extremely large value for the sample time could result in system instability, making the selection of the ideal sampling time difficult [94]. A new type-2 fuzzy wavelet neural network (FWNN) structure was suggested [95] that combines wavelet function in type-2 fuzzy logic inference structure. However, variables that fluctuate outside of predetermined limits won't be considered. Interval type-2 fuzzy logic systems (IT2FLSs) can now be optimized utilizing two different types of tabu search (TS) [96]. The interval type-2 fuzzy logic system (IT2FLS) rule base membership functions parameters are optimized by TS algorithms. Two benchmark datasets were classified using IT2FLS and directed tabu search (DTS), which directs TS moves using pattern search, and short-term tabu search (STS). In comparison to the STS with IT2FLS, the DTS has performed better. However, this work's optimization approach did not include the computation related to T2FLS. Wu and Tan created an IT2FLS with the use of a coevolutionary technique in [97]. The interpretability of IT2-

FIS and the evolution of uncertainty must be addressed, and the co - evolution of IT2-FIS requires a greater computational cost. The complexity of IT2FLS affects how long the algorithm takes to run. IT2FLS coevolution is approximately 26.4 times slower than T1FLS co - evolution. With hybrid learning algorithm methodologies, three interval type-2 fuzzy neural network (IT2FNN) designs were presented [98]. However, having some intuitive aspects that make the resulting interval type-2 fuzzy rules simple to understand is a challenging task in training IT2FNN.

6. CONCLUSION

This study provides a review of soft computing methods used in induction motor analysis and control. As the most energy-intensive machine in the sector, induction motors can help to minimize peak power demand and lower energy costs by adopting energy conservation. If the motor speed can be changed in accordance with the load and the ripple content is decreased, a substantial amount of energy can be saved. By creating suitable controls, the motor's speed may be managed. This paper's main contribution is a thorough analysis of soft computing techniques, such as artificial neural networks (ANN), fuzzy logic, and neuro-fuzzy controllers, in terms of their precision, complexity, classification and regression abilities, convergence times, self-organizing capabilities, advantages, and disadvantages. This review has made some relevant and well-chosen recommendations for the continued technological advancement of IM controllers. Additional study on type-2 FLCT, type-2 ANN, and type-2 neuro-fuzzy based SVPWM switching approaches for various inverter configurations should be conducted in order to improve the accuracy of ANN, ANFIS, and FLC. Intelligent controllers like FLCT (type-2), ANN, and type-2 neuro-fuzzy controllers are recommended to be included in IM to reduce overshoot, settling time, and steady state inaccuracy. By choosing the proper membership function and rules, Neuro-fuzzy and FLC performance can be enhanced. The accuracy of the ANN can also be improved by selecting the right hidden layer neurons. This review has made some noteworthy and well-chosen recommendations for the continued technological advancement of IM controllers. To find the best values and least amount of error for ANN, Neuro-fuzzy, and FLC, a variety of optimization approaches, such as the genetic algorithm, particle swarm optimization, lighting search algorithm, and backtracking search algorithm, may be used (type-2). It is important to investigate the study on determining the proper value of the PID control parameters (Kp, Ki, and Kd). To lower the cost of the control system's production, the designed controller can be used with multiple DC motors or multiple permanent magnet synchronous motor drives. These suggestions would significantly advance the design and implementation of soft computing controllers and give manufacturers and researchers a clear direction for the development of IM in the future.

REFERENCES

[1] A. Ba-Razzouk, A. Chériti, G. Olivier, and P. Sicard, “Field-oriented control of induction motors using neural-network decouplers,” *IEEE Trans. Power Electron.*, vol. 12, no. 4, pp. 752–763, 1997.

[2] S. Chakraverty and S. Mall, “Artificial neural networks for engineers and scientists: Solving ordinary differential equations,” *Artif. Neural Networks Eng. Sci. Solving Ordinary Differ. Equations*, pp. 1–150, 2017.

[3] M. Cirrincione, M. Pucci, and G. Vitale, *Power Converters and AC Electrical Drives with Linear Neural Networks.* CRC Press, 2012.

[4] B. K. Bose, “Neural network applications in power electronics and motor drives - An introduction and perspective,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 14–33, 2007.

[5] B. Yegnanarayana, *Artificial Neural Networks*. PHI Learning Pvt. Ltd, 2009.

[6] M. A. Rahman and M. A. Hoque, “On-line adaptive artificial neural network based vector control of permanent magnet synchronous motors,” *IEEE Trans. Energy Convers.*, vol. 13, no. 4, pp. 311–318, 1998.

[7] M. P. Kazmierkowski, R. Krishnan, and F. Blaabjerg, “Control in Power Electronics: Selected Problems,” *Control Power Electron. Sel. Probl.*, pp. 1–518, 2003.

[8] B. Karanayil, M. F. Rahman, and C. Grantham, “Online stator and rotor resistance estimation scheme using artificial neural networks for vector controlled speed sensorless induction motor drive,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 167–176, 2007.

[9] C. B. Butt and M. A. Rahman, “Intelligent speed control of interior permanent magnet motor drives using a single untrained artificial neuron,” *IEEE Trans. Ind. Appl.*, vol. 49, no. 4, pp. 1836–1843, 2013.

[10] A. Saxena, S. Dutta, and A. Chitra, “Artificial Neural Network Controller for Vector ControlledInduction Motor Drive,” *Int. J. Comput. Appl.*, vol. 46, no. 14, 2012.

[11] A. Denai M. and A. Attia S., “Fuzzy and Neural Control of an Induction Motor,” *Int. J. Appl. Math. Comput. Sci.*, vol. 12, no. 2, pp. 221–233, 2002.

[12] A. Bechouche, H. Sediki, D. O. Abdeslam, and S. Haddad, “Identification of induction motor at standstill using artificial neural network,” *IECON Proc. (Industrial Electron. Conf.*, pp. 2908–2913, 2010.

[13] M. E. Elbuluk, L. Tong, and I. Husain, “Neural-network-based model reference adaptive systems for high-performance motor drives and motion controls,” *IEEE Trans. Ind. Appl.*, vol. 38, no. 3, pp. 879–886, 2002.

[14] Tien-Chi Chen and Tsong-Terng Sheu, “Model reference neural network controller for induction motor speed control,” *IEEE Trans. Energy Convers.*, vol. 17, no. 2, pp. 157–163, Jun. 2002.

[15] F. J. Lin, R. J. Wai, W. Der Chou, and S. P. Hsu, “Adaptive backstepping control using recurrent neural network for linear induction motor drive,” *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 134–146, 2002.

[16] C. M. Lin and C. F. Hsu, “Neural-network-based adaptive control for induction servomotor drive system,” *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 115–123, 2002.

[17] S. M. Gadoue, D. Giaouris, and J. W. Finch, “A Neural Network based stator current MRAS observer for speed sensorless induction motor drives,” *IEEE Int. Symp. Ind. Electron.*, pp. 650–655, 2008.

[18] M. I. Marei, M. F. Shaaban, and A. A. El-Sattar, “A speed estimation unit for induction motors based on adaptive linear combiner,” *Energy Convers. Manag.*, vol. 50, no. 7, pp. 1664–1670, 2009.

[19] D.-S. Huang, Y. Gan, P. Gupta, and M. M. Gromiha, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, vol. 6839. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[20] R. Krishnan and A. S. Bharadwaj, “A review of parameter sensitivity and adaptation in indirect vector controlled induction motor drive systems,” *PESC Rec. - IEEE Annu. Power Electron. Spec. Conf.*, vol. 2, pp. 560–566, 1990.

[21] O. Deperlioglu and U. Kose, “An educational tool for artificial neural networks,” *Comput. Electr. Eng.*, vol. 37, no. 3, pp. 392–402, 2011.

[22] L. Frosini and G. Petrecca, “Neural networks for load torque monitoring of an induction motor,” *Appl. Soft Comput.*, vol. 1, no. 3, pp. 215–223, 2001.

[23] D. Bae, D. Kim, H. K. Jung, S. Y. Hahn, and C. S. Koh, “Determination of induction motor parameters by using neural network based on FEM results,” *IEEE Trans. Magn.*, vol. 33, no. 2 PART 2, pp. 1924–1927, 1997.

[24] M. Ikeda and T. Hiyama, “ANN based designing and cost determination system for induction motor,” *IEE Proc. Electr. Power Appl.*, vol. 152, no. 6, pp. 1595–1602, 2005.

[25] C. H. Tsai and M. F. Yeh, “Application of CMAC neural network to the control of induction motor drives,” *Appl. Soft Comput. J.*, vol. 9, no. 4, pp. 1187–1196, 2009.

[26] D. S. Reay, Y. Dessouky, and B. W. Williams, “The use of neural networks to enhance sensorless position detection in switched reluctance motors,” in *SMC’98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.98CH36218)*, vol. 2, pp. 1774–1778.

[27] W. S. Oh, B. K. Bose, K. M. Cho, and H. J. Kim, “Self tuning neural network controller for induction motor drives,” *IECON Proc. (Industrial Electron. Conf.*, vol. 1, pp. 152–156, 2002.

[28] S. Weerasooriya and M. A. El-Sharkawi, “Identification and Control of a DC Motor Using Backpropagation Neural Networks,” *IEEE Trans. Energy Convers.*, vol. 6, no. 4, pp. 663–669, 1991.

[29] J. R. Raol and H. Madhuranath, “Neural network architectures for parameter estimation of dynamical systems,” *IEE Proc. Control Theory Appl.*, vol. 143, no. 4, pp. 387–394, 1996.

[30] G. Mino-Aguilar, J. M. Moreno-Eguilaz, B. Pryymak, J. Peracaula, and J. A. Beristain, “A comparative analysis of two neural-network-based estimators for efficiency optimization of an induction motor drive,” *Int. Power Electron. Congr. - CIEP*, pp. 33–38, 2006.

[31] C. H. Tsai, “Neural network application for flux and speed estimation in the sensorless decoupling induction motor drive,” *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, vol. 6, pp. 5297–5303, 2006.

[32] B. Karanayil, M. F. Rahman, and C. Grantham, “Stator and rotor resistance observers for induction motor drive using fuzzy logic and artificial neural networks,” *IEEE Trans. Energy Convers.*, vol. 20, no. 4, pp. 771–780, 2005.

[33] D. Vukadinovic, M. Basic, and L. Kulisic, “Stator resistance identification based on neural and fuzzy logic principles in an induction motor drive,” *Neurocomputing*, vol. 73, no. 4–6, pp. 602–612, 2010.

[34] A. Rubaai, R. Kotaru, and M. D. Kankam, “Online training of parallel neural network estimators for control of induction motors,” *IEEE Trans. Ind. Appl.*, vol. 37, no. 5, pp. 1512–1521, 2001.

[35] S. H. Kim, T. S. Park, J. N. Yoo, and G. T. Park, “Speed-sensorless vector control of an induction motor using neural network speed estimation,” *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 609–614, 2001.

[36] J. Campbell and M. Sumner, “Practical sensorless induction motor drive employing an artificial neural network for online parameter adaptation,” *IEE Proc. Electr. Power Appl.*, vol. 149, no. 4, pp. 255–260, 2002.

[37] S. Maiti, V. Verma, C. Chakraborty, and Y. Hori, “An adaptive speed sensorless induction motor drive with artificial neural network for stability enhancement,” *IEEE Trans. Ind. Informatics*, vol. 8, no. 4, pp. 757–766, 2012.

[38] M. Wlas, Z. Krzemiński, J. Guziński, H. Abu-Rub, and H. A. Toliyat, “Artificial-neural-network-based sensorless nonlinear control of induction motors,” *IEEE Trans. Energy Convers.*, vol. 20, no. 3, pp. 520–528, 2005.

[39] A. Gastli and M. M. Ahmed, “ANN-based soft starting of voltage-controlled-fed IM drive system,” *IEEE Trans. Energy Convers.*, vol. 20, no. 3, pp. 497–503, 2005.

[40] L. Guo and L. Parsa, “Model reference adaptive control of five-phase IPM Motors based on neural network,” *2011 IEEE Int. Electr. Mach. Drives Conf. IEMDC 2011*, pp. 563–568, 2011.

[41] S. M. Gadoue, D. Giaouris, and J. W. Finch, “Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers,” *IEEE Trans. Ind. Electron.*, vol. 56, no. 8, pp. 3029–3039, 2009.

[42] A. Venkadesan, S. Himavathi, K. Sedhuraman, and A. Muthuramalingam, “Design and field programmable gate array implementation of cascade neural network based flux estimator for speed estimation in induction motor drives,” *IET Electr. Power Appl.*, vol. 11, no. 1, pp. 121–131, 2017.

[43] S. K. Mondal, J. O. P. Pinto, and B. K. Bose, “A neural-network-based space-vector PWM controller for a three-level voltage-fed inverter induction motor drive,” *IEEE Trans. Ind. Appl.*, vol. 38, no. 3, pp. 660–669, 2002.

[44] B. Sai Shiva, V. Verma, and Y. A. Khan, “Q-MRAS-Based Speed Sensorless Permanent Magnet Synchronous Motor Drive with Adaptive Neural Network for Performance Enhancement at Low Speeds,” *Innov. Soft Comput. Inf. Technol.*, pp. 103–116, 2019.

[45] M. Suetake, I. N. Da Silva, and A. Goedtel, “Embedded DSP-based compact fuzzy system and its application for induction-motor V/f speed control,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 3, pp. 750–760, 2011.

[46] N. Farah *et al.*, “A Novel Self-Tuning Fuzzy Logic Controller Based Induction Motor Drive System: An Experimental Approach,” *IEEE Access*, vol. 7, pp. 68172–68184, 2019.

[47] Q. A. Tarbosh *et al.*, “Review and Investigation of Simplified Rules Fuzzy Logic Speed Controller of High Performance Induction Motor Drives,” *IEEE Access*, vol. 8, pp. 49377–49394, 2020.

[48] C. A. Rojas, J. R. Rodriguez, S. Kouro, and F. Villarroel, “Multiobjective Fuzzy-Decision-Making Predictive Torque Control for an Induction Motor Drive,” *IEEE Trans. Power Electron.*, vol. 32, no. 8, pp. 6245–6260, 2017.

[49] M. A. Hannan *et al.*, “A Quantum Lightning Search Algorithm-Based Fuzzy Speed Controller for Induction Motor Drive,” *IEEE Access*, 2017.

[50] N. Venkataramana Naik, S. P. Singh, and A. K. Panda, “An Interval Type-2 Fuzzy-Based DTC of IMD Using Hybrid Duty Ratio Control,” *IEEE Trans. Power Electron.*, vol. 35, no. 8, pp. 8443–8451, 2020.

[51] A. Berzoy, J. Rengifo, and O. Mohammed, “Fuzzy Predictive DTC of Induction Machines with Reduced Torque Ripple and High-Performance Operation,” *IEEE Trans. Power Electron.*, vol. 33, no. 3, pp. 2580–2587, 2018.

[52] M. R. Douiri and M. Cherkaoui, “Comparative study of various artificial intelligence approaches applied to direct torque control of induction motor drives,” *Front. Energy*, vol. 7, no. 4, pp. 456–467, 2013.

[53] S. Mir, M. E. Elbuluk, and D. S. Zinger, "PI and fuzzy estimators for tuning the stator resistance in direct torque control of induction machines," *IEEE Trans. Power Electron.*, vol. 13, no. 2, pp. 279–287, Mar. 1998.

[54] V. Naik N and S. P. Singh, "A Novel Interval Type-2 Fuzzy-Based Direct Torque Control of Induction Motor Drive Using Five-Level Diode-Clamped Inverter," *IEEE Trans. Ind. Electron.*, vol. 68, no. 1, pp. 149–159, Jan. 2021.

[55] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Analysis of the noise reduction property of type-2 fuzzy logic systems using a novel type-2 membership function," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 41, no. 5, pp. 1395–1406, 2011.

[56] M. B. Ozek and Z. H. Akpolat, "A software tool: Type-2 fuzzy logic toolbox," *Comput. Appl. Eng. Educ.*, vol. 16, no. 2, pp. 137–146, 2008.

[57] N. Venkataramana Naik, A. Panda, and S. P. Singh, "A Three-Level Fuzzy-2 DTC of Induction Motor Drive Using SVPWM," *IEEE Trans. Ind. Electron.*, 2016.

[58] N. Venkataramana Naik and S. P. Singh, "A Comparative Analytical Performance of F2DTC and PIDTC of Induction Motor Using DSPACE-1104," *IEEE Trans. Ind. Electron.*, 2015.

[59] N. V. Naik and S. P. Singh, "Improved torque and flux performance of type-2 fuzzy-based direct torque control induction motor using space vector pulse-width modulation," *Electr. Power Components Syst.*, vol. 42, no. 6, pp. 658–669, 2014.

[60] T. Ramesh and A. K. Panda, "Direct flux and torque control of three phase induction motor drive using PI and fuzzy logic controllers for speed regulator and low torque ripple," *2012 Students Conf. Eng. Syst. SCES 2012*, 2012.

[61] B. Subudhi, A. Kumar, and D. Jena, "dSPACE implementation of fuzzy logic based vector control of induction motor," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, 2008.

[62] L. X. Wang, "A new look at type-2 fuzzy sets and type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 3, pp. 693–706, 2017.

[63] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Extended kalman filter based learning algorithm for type-2 fuzzy logic systems and its experimental evaluation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4443–4455, 2012.

[64] K. Subramanian, R. Savitha, and S. Suresh, "A metacognitive complex-valued interval type-2 fuzzy inference system," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 9, pp. 1659–1672, 2014.

[65] A. Saha, A. Konar, and A. K. Nagar, "EEG Analysis for Cognitive Failure Detection in Driving Using Type-2 Fuzzy Classifiers," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 1, no. 6, pp. 437–453, 2017.

[66] T. Kumbasar and H. Hagras, "A Self-Tuning zSlices-Based General Type-2 Fuzzy PI Controller," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 4, pp. 991–1013, 2015.

[67] Z. Deng, K. S. Choi, L. Cao, and S. Wang, "T2fela: Type-2 fuzzy extreme learning algorithm for fast training of interval type-2 TSK fuzzy logic system," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 4, pp. 664–676, 2014.

[68] R. Qun, L. Baron, and M. Balazinski, "Type-2 takagi-sugeno-kang fuzzy logic modeling using subtractive clustering," *Annu. Conf. North Am. Fuzzy Inf. Process. Soc. - NAFIPS*, pp. 120–125, 2006.

[69] Q. Ren, M. Balazinski, and L. Baron, "Uncertainty prediction for tool wear condition using type-2 TSK fuzzy approach," *Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern.*, pp. 660–665, 2009.

[70] Q. Ren, L. Baron, M. Balazinski, and K. Jemielniak, "Acoustic emission signal feature analysis using type-2 fuzzy logic system," *Annu. Conf. North Am. Fuzzy Inf. Process. Soc. - NAFIPS*, 2010.

[71] Q. Rena, M. Balazinskia, L. Barona, and K. Jemielniakb, "TSK fuzzy modeling for tool wear condition in turning processes: An experimental study," *Eng. Appl. Artif. Intell.*, vol. 24, no. 2, pp. 260–265, 2011.

[72] A. Zadeh L., "Fuzzy sets. Information and Control," *Inf. Control*, vol. 8, no. 1, pp. 338–353, 1965.

[73] E. Hisdal, "The IF THEN ELSE statement and interval-valued fuzzy sets of higher type," *Int. J. Man. Mach. Stud.*, vol. 15, no. 4, pp. 385–455, 1981.

[74] E. H. Mamdani and N. Baaklini, "Prescriptive method for deriving control policy in a fuzzy-logic controller," *Electron. Lett.*, vol. 11, no. 25–26, p. 625, 1975.

[75] J. M. Mendel, R. I. John, and F. Liu, "Interval Type-2 Fuzzy Logic Systems Made Simple," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 6, pp. 808–821, Dec. 2006.

[76] Q. Liang and J. M. Mendel, "Interval type-2 fuzzy logic systems: Theory and design," *IEEE Trans. Fuzzy Syst.*, vol. 8, no. 5, pp. 535–550, 2000.

[77] M. Zamani, H. Nejati, A. T. Jahromi, A. Partovi, S. H. Nobari, and G. N. Shirazi, "Toolbox for Interval Type-2 Fuzzy Logic Systems," in *Proceedings of the 11th Joint Conference on Information Sciences (JCIS)*, 2008.

[78] N. N. Karnik, J. M. Mendel, and Qilian Liang, "Type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 6, pp. 643–658, 1999.

[79] H. Annison, "Book review: Uncertain rule-based fuzzy logic systems: introduction and new directions," *Criminol. Crim. Justice*, vol. 11, no. 3, pp. 277–278, Jul. 2011.

[80] N. N. Karnik and J. M. Mendel, "Operations on type-2 fuzzy sets," *Fuzzy Sets Syst.*, vol. 122, no. 2, pp. 327–348, 2001.

[81] L. A. Zadeh, "Toward extended fuzzy logic-A first step," *Fuzzy Sets Syst.*, vol. 160, no. 21, pp. 3175–3181, 2009.

[82] J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro-Fuzzy and Soft Computing-A Computational Approach to Learning and Machine Intelligence [Book Review]," *IEEE Trans. Automat. Contr.*, vol. 42, no. 10, pp. 1482–1484, 2005.

[83] R. N. Mishra and K. B. Mohanty, "Real time implementation of an ANFIS-based induction motor drive via feedback linearization for performance enhancement," *Eng. Sci. Technol. an Int. J.*, vol. 19, no. 4, pp. 1714–1730, 2016.

[84] V. Sumati and C. Patvardhan, "Interval Type-2 Mutual Subsethood Fuzzy Neural Inference System (IT2MSFuNIS)," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 1, pp. 203–215, 2018.

[85] E. Kayacan, O. Kaynak, R. Abiyev, J. Toørresen, M. Hoøvin, and K. Glette, "Design of an adaptive interval type-2 fuzzy logic controller for the position control of a servo system with an intelligent sensor," *2010 IEEE World Congr. Comput. Intell. WCCI 2010*, 2010.

[86] J. M. Mendel, "Computing Derivatives in Interval Type-2 Fuzzy Logic Systems," *IEEE Trans. Fuzzy Syst.*, vol. 12, no. 1, pp. 84–98, 2004.

[87] C. H. Wang, C. S. Cheng, and T. T. Lee, "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 34, no. 3, pp. 1462–1477, 2004.

[88] M. A. Khanesar, E. Kayacan, M. Teshnehlab, and O. Kaynak, "Levenberg Marquardt algorithm for the training of type-2 fuzzy neuro systems with a novel type-2 fuzzy membership function," *IEEE SSCI 2011 Symp. Ser. Comput. Intell. - T2FUZZ 2011 2011 IEEE Symp. Adv. Type-2 Fuzzy Log. Syst.*, no. Lm, pp. 88–93, 2011.

[89] M. A. Khanesar, M. Teshnehlab, E. Kayacan, and O. Kaynak, "A novel type-2 fuzzy membership function: Application to the prediction of noisy data," *CIMSA 2010 - IEEE Int. Conf. Comput. Intell. Meas. Syst. Appl. Proc.*, no. 1, pp. 128–133, 2010.

[90] O. Poleshchuk and E. Komarov, "A Fuzzy Nonlinear Regression Model for Interval Type-2 Fuzzy Sets," vol. 8, no. 6, pp. 840–844, 2014.

[91] F. C. H. Rhee and B. I. Choi, "Interval type-2 fuzzy membership function design and its application to radial basis function neural networks," *IEEE Int. Conf. Fuzzy Syst.*, vol. 1, no. c, 2007.

[92] R. Amar, H. Mustapha, and T. Mohamed, "Decentralized RBFNN type-2 fuzzy sliding mode controller for robot manipulator driven by artificial muscles," *Int. J. Adv. Robot. Syst.*, vol. 9, 2012.

[93] P. Wang, N. Li, and S. Li, "Interval type-2 fuzzy T-S modeling for a heat exchange process on CE117 Process Trainer," *Proc. 2011 Int. Conf. Model. Identif. Control. ICMIC 2011*, pp. 457–462, 2011.

[94] O. Castillo and P. Melin, “Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review,” *Inf. Sci. (Ny).*, vol. 205, pp. 1–19, 2012.

[95] O. Castillo, R. Martínez-Marroquín, P. Melin, F. Valdez, and J. Soria, “Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot,” *Inf. Sci. (Ny).*, vol. 192, pp. 19–38, 2012.

[96] S. Park and H. Lee-Kwang, “A designing method for type-2 fuzzy logic systems using genetic algorithms,” *Annu. Conf. North Am. Fuzzy Inf. Process. Soc. - NAFIPS*, vol. 5, no. C, pp. 2567–2572, 2001.

[97] D. Wu and W. Wan Tan, “Genetic learning and performance evaluation of interval type-2 fuzzy logic controllers,” *Eng. Appl. Artif. Intell.*, vol. 19, no. 8, pp. 829–841, 2006.

[98] K. J. Park, S. K. Oh, and W. Pedrycz, “Design of interval type-2 fuzzy neural networks and their optimization using real-coded genetic algorithms,” *IEEE Int. Conf. Fuzzy Syst.*, no. 2, pp. 2013–2018, 2009.

BIOGRAPHIES OF AUTHORS

Gadwala Durgasukumar received his B.Tech. and M.Tech. degree in electrical engineering from JNTU Hyderabad and PhD degree in power electronics and drives from Indian Institute of Technology, Roorkee, India in 2012. He is currently professor in the Department of Electrical & Electronics Engineering, Vignan Institute of technology and science, Hyderabad, India. His research interests include electromagnetic fields, power electronics and drives, soft computing techniques, electrical circuits. He has authored of many research articles/books related to electromagnetic fields, power electronics and drives, soft computing techniques, and electrical circuits. He can be contacted at email: durgasukumar@gmail.com.

Repana Ramanjan Prasad received his B.Tech. and M.Tech. degree in electrical engineering from JNTU Hyderabad. He is currently pursuing PhD at the Department of Electrical Engineering at Vignan’s University, Guntur, India. His research interests include Power electronics and drives and soft computing techniques. He can be contacted at email: prasad243@gmail.com.

Srinivasa Rao Gorantla received his B.Tech. degree in electrical and electronics engineering from ANU Vijayawada, India, M.Tech. degree in electrical and electronics engineering from Anna University, Chennai, India and Ph.D. degree in electrical engineering from JNTU Kakinada, India. Gorantla is currently working as a professor in the Department of Electrical & Electronics Engineering, Vignan's University, India. His research interests include hybrid electric vehicles, reactive power compensation, and fuel cells. He has authored of many research articles/books related to hybrid electric vehicles, reactive power compensation, and fuel cells. He can be contacted at email: srn.gorantla@gmail.com.