International Journal of Engineering

Journal Homepage: www.ije.ir

Memristor Crossbar-based Hardware Implementation of Type-2 Fuzzy Membership Function and On-chip Learning

S. Haghzad Klidbary*[a], M. Javadian[b], R. Omidi[c], R. P. R. Hasanzadeh[d]

[a] *Department of Electrical and Computer Engineering, University of Zanjan, Zanjan, Iran*
[b] *Computer Department, Kermanshah University of Technology, Kermanshah, Iran*
[c] *Department of Electrical Engineering, University of Zanjan, Zanjan, Iran*
[d] *Department of Electrical Engineering, University of Guilan, Rasht, Iran*

PAPER INFO

Paper history:
Received 7 March 2021
Received in revised from 7 August 2021
Accepted 9 August 2021

Keywords:
Fuzzy Inference System
Fuzzy Membership Function
Type-1 Fuzzy
Type-2 Fuzzy
Hardware Implementation
Memristor-crossbar Structures

ABSTRACT

Utilizing fuzzy techniques, especially fuzzy type-2, is one of the most widely used methods in machine learning to model uncertainty. In addition to algorithm provision, the hardware implementation capability, and proper performance in real-time applications are other challenges. The use of hardware platforms that have biological similarities and are comparable to human neural systems in terms of implementation volume has always been considered. Memristor is one of the emerging elements for the implementation of fuzzy logic based algorithms. In this element, by providing current and selecting the appropriate direction for the applied current, the resistance of the memristor (memristance) will increase or decrease. Various implementations of type-1 fuzzy (T1F) systems exist, but no implementation of type-2 fuzzy (T2F) systems has been done based on memristors. In this paper, memristor-crossbar structures are used to implement type-2 fuzzy membership functions. In the proposed hardware, the membership functions can have any shape and resolution. Our proposed implementation of type-2 fuzzy membership function has the potential to learn (On-Chip learning capability regardless of the host system). Besides, the proposed hardware is analog and can be used as a basis in the construction of evolutionary systems. Furthermore, the proposed approach is applied to memristor emulator to demonstrate its correct operation.

***doi**: 10.5829/ije.2021.34.09C.15*

1. INTRODUCTION

Nowadays, the design of artificial intelligence-based computing machines that can intelligently perform human-like tasks is one of the main goals of researchers. Therefore, various algorithms have been proposed. The proposed algorithms can be categorized into the artificial neural networks, evolutionary algorithms and fuzzy systems. Neural Networks seek to find structures similar to the neural structures of living organisms, while the evolutionary algorithms try to seek swarm intelligence in nature. On the other hand, the fuzzy systems tries to find functional concepts identical to the human brain.

Fuzzy logic theory can be considered as a human effort to create a tool to deal with the inherent inaccuracy of systems. "Fuzzy logic" was first introduced in 1965. The word fuzzy means inaccurate, vague, and ambiguous. The idea of fuzzy sets was based on linguistic variables. In fuzzy logic, the problem is modeled as a group of "IF-THEN" rules, unlike the classical methods which model the problem by expressing complex mathematical relations. Fuzzy logic was utilized in various applications [1-5]]. When the efficiency of type-1 fuzzy (T1F) sets are approved in many research papers, Zadeh et al. [3] presented the second series of fuzzy sets. In this set, each degree of membership was also a member of a fuzzy set and named it the type-2 fuzzy (T2F) set. The T2F set was introduced in 1975 and is an extension of T1F sets. Some of the reasons for creating the T2F set are stated below [6, 7]: (1) The inability to

*Corresponding Author Institutional Email: s.haghzad@znu.ac.ir (S. H. Klidbary)

Please cite this article as: S. Haghzad Klidbary, M. Javadian, R. Omidi, R. P. R. Hasanzadeh, Memristor Crossbar-based Hardware Implementation of Type-2 Fuzzy Membership Function and On-chip Learning, International Journal of Engineering, Transactions C: Aspects, Vol. 34, No. 09, (2021) 2180-2188

mathematically describe systems in which the data-generating is known to be time-varying (mobile telecommunications); (2) The inability in the mathematical description of non-stationary noise; (3) Inability in the mathematical description of features in pattern recognition applications that have non-stationary statistical properties; (4) Knowledge obtained from a group of experts (expertise extraction) those using questionnaires that involve uncertain words; (5) Use linguistic expressions that have a non-measurable domain.

Currently, the T2F set has found many applications in the fields with high uncertainty factor. These applications include pre-processing of radiographic images [8], estimating the health of the newborn [9], controlling mobile robots [10], edge detection in digital photographs [11]. According to the increasing application of T2F sets in real-world engineering problems, there is a need for a fast hardware for implementation. Therefore, providing an efficient hardware for implementation of these algorithms would be beneficial for engineering problems. Because, the advantages of the soft computing algorithms will be observable when they can be used as hardware in the form of an intelligent system with high computational power and efficiency.

In hardware implementation, inspiration from nature and the implementation of brain-like structures have received much attention. The brain is structurally composed of a large number of neurons and synapses. It can be concluded that the density of neurons in the brain is very high by considering the dimensions of the brain. So far, many computational structures were developed based on CMOS technology [12-14]. Since this technology faces challenges at the nanoscale, it is not a proper option for implementing Bio-Inspired systems and achieving high volumes of processing elements. Therefore, it can be said that the manufacturing of more advanced technologies, more processing power, more compact circuit, and less power consumption are the most critical issues in the construction of an artificial brain. After the implementation of the memristor in HP (Hewlett Packard) Lab., the hope for achieving a computing platform comparable to the neural system of living organisms has increased.

Memristor is a nonlinear passive electrical element whose resistance is controlled by voltage. This element, by having a nanoscale size, small volume, low power consumption, and also identical behavior similar to the biological synapse, has been able to find an appropriate position in the hardware implementation of computational algorithms. This element attracted the attention of many researchers in various fields such as artificial neural networks and fuzzy systems. In this paper, the membership functions of T2F sets are implemented on the memristor-crossbar structure. Also, a circuit to type-reduction in this T2F set is given.

The rest of this paper is organized as follows. The main concepts of memristor element are reviewed in section 2. In section 3, the T2F set is discussed. The memristor-crossbar hardware implementation is presented in section 4. In section 5, the simulation results are reported. Finally, section 6 includes summaries, conclusions, and suggestions for future work.

2. MEMRISTOR BASICS

In addition to the three previously known fundamental electrical elements: capacitors, inductors, and resistance, in 1976, Chua and kang [22] introduced the fourth element by mathematical proof. This element, as a passive element, is the relationship between magnetic flux and electric charge [15, 16]. Chua [23] called this element as memristor, which stands for memory and resistor. Before 2008, no successful implementation of this element was reported, where the main reason was that the memristor property is visible in nanoscale. This element was successfully realized in mid-2008 [17]. After that, many researches have been done about memristor and its applications [18-21]. Memristor acts like a linear resistor, except that it has memory. This means that the memristor's properties at any given time depend on how the current or the voltage has been applied to it in the past. Figure 1 shows the relationships between different electrical parameters.

The feature of the memristor is that if the current passes through the memristor in one direction, its resistance will increase and if the direction of the current changes, the resistance value will decrease. So the memristor is called a memory resistor. In the following, the description of the memristor, which is provided by HP, is given by Chua [23].

2. 1. The Memristor Modeling This memristor is comprised of a very thin TiO_2 with the width, D, sandwiched between two platinum electrodes. The

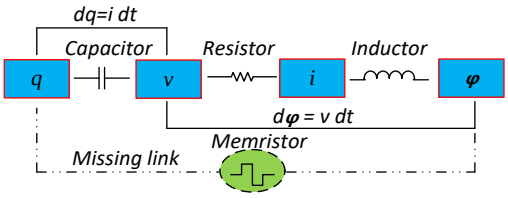

Figure 1. A representation of the four basic electrical elements, which are the resistor, the capacitor, the inductor, and the memristor. Memristor is the fourth fundamental element; the missing link represents the relationship between magnetic flux and electric charge. Theoretically, the information stored in memristors will be stored indefinitely if no current passes through it. Unlike a capacitor, it will not need to be refreshed

semiconductor layer has two regions, one with oxygen impurities and R_{on} resistance and the second without impurities with R_{off} resistance. By applying an external voltage, the border between the doped and undoped regions can be displaced (Figure 2(a)). This will change the memristance value of the memristor.

Memristance value varies between R_{on} and R_{off}. The memristance value reaches its minimum (equal to R_{on}) when the doped region extends to D ($w \to D$). The memristance value reaches its maximum (equal to R_{off}) when the undoped region extends to D ($w \to 0$). The mathematical model of the memristor, assuming the uniformity of the electric field inside the memristor, is as follows:

$$R_M(w) = R_{on} \times \frac{w(t)}{D} + R_{off} \times \left(1 - \frac{w(t)}{D}\right) \quad (1)$$

$$w(t) = w_0 + \frac{\mu_v R_{on}}{D} q(t) \quad (2)$$

where W_0 is the initial width of the doped region w, μ_v is ion mobility and $q(t)$ is the electrical charge that passes through the element.

A crucial point in a memristor is that memristor properties are visible in nanoscale. This characteristic of memristor makes it possible to implement hardware similar to the brain, which has a high density of neurons and synapses. The ability to store data in analog form, non-volatile memory, nanoscale fabrication and higher speed due to the nature of analog are some of the advantages of this element. Using the memristor alone is not optimal, and such as mentioned in Figure 3, this element is used as a crossbar structure. A crossbar consists of a series of horizontal wires passing over vertical ones. A memristor is placed at the intersection of two wires that cross each other. By applying proper voltage over any pair of vertical and horizontal wires, the memristor at that intersection can be accessed. Some implementations of fuzzy systems on memristor structures are given below. The memory property of memristor has been used, and by utilizing this feature, the four primary operations (addition, subtraction, multiplication, and division) have been implemented by Merrikh-Bayat and Shouraki [22]. Amer et al. [23] discussed on the implementation of Center Of Gravity (COG). The combination of memristor and resistor is used to make two operators, minimum and maximum [24]. The implementation of T1F membership functions on memristor-crossbar structures is given by Merrikh-Bayat et al. [25]. Merrikh-Bayat and Shouraki [26] presented, a new neuro-fuzzy computational structure. The implementation of ink drop spread (IDS) in 2D plans on the memristor-crossbar structure is presented by Merrikh-Bayat et al. [27]. There are also other references related to the implementation of the ALM (Active Learning Method) algorithm on memristor-crossbar [28-30]. Memristor has also been used as a synapse in neuromorphic computing [31-34]. Figure 3 shows the memristor-crossbar structure and its applications as a synapse is given.

As we know, fuzzy systems can be considered as human efforts to create a tool to deal with the inherent inaccuracies of systems. T2F set is an extension of the T1F set which has been able to provide proper results due to its ability to deal with the uncertainty of system rules and parameters. According to the researches, it can be stated that so far, in addition to all the implemented cases expressed on the memristor-crossbar structures, no implementations of T2F membership functions have been observed on the memristor structure. In the following, we will first give an overview of the T2F set.

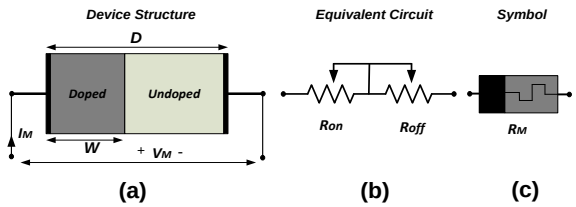

Figure 2. (a) the structure of the memristor, which consists of two regions respectively doped (low resistanceR_{on}) and undoped (high resistanceR_{off}). By applying a voltage (V_m) with appropriate polarity, this border can be changed, (b) circuit equivalent of memristor, (c) The symbol used for the memristor with R_M memristance

3. TYPE-2 FUZZY(T2F) SET

In the T1F set, a certain degree of membership between zero and one is considered. This specificity of the degree of membership challenges the uncertainty. After the aforementioned problems, the T2F set was proposed. The T2F set acts as a probability density function and provides uncertainty about the degree of membership using the variance around that degree of membership. Therefore, it can be mentioned that the T1F set models the first-degree of uncertainty, and the T2F set models the second-degree of uncertainty. Therefore, it can be

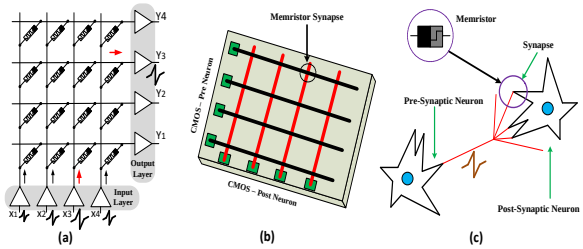

Figure 3. (a) Memristor-crossbar based circuit as a hardware implementation of spiky neural network (SNN), (b) the memristor-crossbar structure, (c) biological synapses

mentioned, the T2F system is also resistant to uncertainties that arise in fuzzy rules or system parameters. Equation (3) is the T1F set and Equations (4) and (5) present T2F set.

$$A = \sum_i \mu_A (x_i)/_{x_i} \ , x_i \in X \tag{3}$$

$$\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) | \forall\, x \in X\} \tag{4}$$

$$\mu_{\tilde{A}}(x) = \sum_i f_x(u_i)/_{u_i} \ , \ u_i \in J_{x_i} \ , J_{x_i} \subseteq [0\ ,1] \tag{5}$$

where x is primary variable, u is secondary variable, $\mu_{\tilde{A}}(x)$ presents the secondary membership function and $J_{x_1} \subseteq [0,1]$ is a set of primary membership degrees. T2F membership functions are described by a characteristic called FOU (footprints of uncertainty), which is a finite range of uncertainties and an expression of the union of primary membership functions. The FOU is enclosed between two T1F membership functions, which in Figure 4, the uncertainty effect by upper membership function (UMF) and lower membership function (LMF) is shown. It is worth mentioning that the values that secondary membership functions take are called the degree of secondary membership. FOU is utilized to eliminate $3D$ display problem of a T2F set that displays borders in a two-dimensional form. Figure 4(a) shows the T1F set, Figure 4(b) shows the T1F set without uncertainty, Figure 4(c) shows the interval T2F membership function and Figure 4(d) indicates the general T2F membership function. Commonly used methods such as Point-Valued Representation (PVR), Vertical-Slice Representation (VSR), Wavy Slice Representation (WSR) or Embedded T2FS Representation, and Horizontal Slice Representation (HSR) or α-plane Representation can be mentioned to express T2F sets. Computations in the T2F set require an algebraic operation and its dedicated fuzzy inference system, just like the T1F set. However, there are extensive researches in this field [35, 36].

After the inference operation, the output is a T2F set, and finally, the output must be brought to the crisp space. In the T2F set, like the T1F set, we need to use the defuzzification procedure. The mapping of a T2F set to a T1F set is called type reduction. The type reduction methods are the same as the methods developed in type-1 systems.

Among the most important methods of type reduction, Generalized Centroid and Karnik-Mendel (KM) can be mentioned. The Generalized Centroid method is the first method where in this method each embedded set has a center of gravity. In the following Equation, the calculation of this method is given and this Equation is calculated for each Embedded Set.

$$C_{\tilde{A}} = \left\{ \left(\frac{\sum_{i=1}^{N} x_i.u_i}{\sum_{i=1}^{N} u_i}, \mu_{\tilde{A}}(x_1)(u_1) * \ldots * \mu_{\tilde{A}}(x_N)(u_N) \right) | \forall i \in \{1, \ldots, N\} : x_i X\, u_i\, J_{x_i} \subseteq U \right\}, \quad t-norm(*) \tag{6}$$

The T-Norm operator is usually chosen to be minimal so that the membership degree obtained is not too small. Of course, by increasing the number of embedded sets, the calculations increase significantly. Another method is Vertical Slice Representation (Vertical Slice Centroid) and the focus of this paper is on this method. In this method, de-fuzzyization takes place in each vertical section, and finally a T1F set is obtained, which can be easily converted to a crisp number. The relation of this method are simple and have been avoided due to brevity.

In the following, the proposed memristor-crossbar hardware for implementation of the T2F membership function set and the type reduction section will be presented.

4. PROPOSED HARDWARE

As mentioned in the section 3, the memristor is a flexible circuit element whose resistance can be controlled by the current flow. In this paper, memristor-crossbar is used as a hardware platform for implementation due to their small size and high density in small volume compared to other elements and also the low energy required for changing the memristance value.

Utilizing memristor-crossbar structures has two phases (or states) of writing and reading. In the writing phase, information is stored on the memristor, which is called the learning phase. In the reading phase, which is called the usage or test phase, the stored information will be read. Intermediate circuits perform these phases on memristor-crossbar. The proposed hardware consists of four sections: control, memristor-crossbar, type-reduction (from T2F set to T1F set), and defuzzification.

The control section can be a microprocessor or FPGA or DSP, which is responsible for controlling the timing

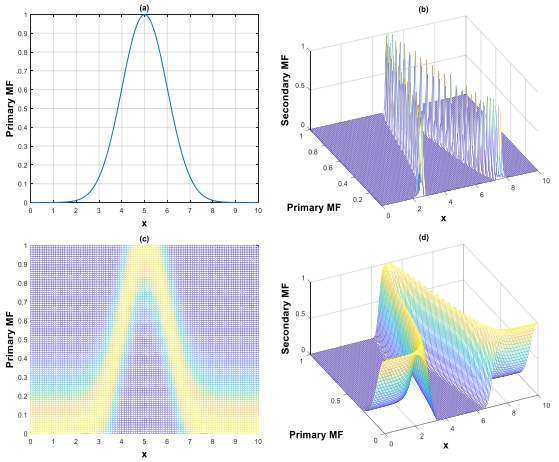

Figure 4. (a) type-1 membership function, (b) type-2 membership function without uncertainty, (c) interval T2F membership function and (d) general T2F membership function

and order of the applied signals. The two sections of type-reduction and defuzzification circuits are composed of resistors, Op-Amps, diodes, and transistors. In this section, we focus more on the memristor-based structure (for the simplicity, a 4x4 memristor-crossbar structure, as illustrated in Figure 5 is used).

For each type-2 function, there is a memristor-crossbar with R_x rows and R_u columns (R_x and R_u are the quantization levels). Each memristor is equivalent to one memory (pixel). These memristors are selected using de-multiplexers. In the training phase, a pulse is applied to the desired column, and the row corresponding to the value u is grounded as shown in Figure 6. At the training phase, the signal p is zero, and all the transistors connecting the rows of crossbar to the Op-Amps are disconnected, which causes the output to be zero volts at the beginning of the training (so these transistors act as isolators).

It can be mentioned that by using the ladder structure [28, 29], changes can be made as a Gaussian function in the memristance of memristors. As mentioned in section 3, Vertical Slice Representation (Vertical Slice Centroid) for T2F set is considered here. In this method, de-fuzzyization takes place in each vertical section, and finally a T1F set is obtained. In the type-reduction section, the maximum function is used. The circuit of this function is shown in Figure 7.

According to Figure 7, there are two series Op-Amps per input line, which is the same number for a system with R_u quantization level. In the test phase, the signal p is one then by applying V_{Read} to the desired rows and columns of the memristor-crossbar, the output voltage of each line is calculated as follows:

$$v_{OP1} = -V_{Read} \times \frac{R_f}{R_{m_{ij}}} \quad (7)$$

$$v_{o1} = V_{Bias} + v_{OP1} \quad (8)$$

where $R_{m_{ij}}$ indicates the memristance value of the memristor in the i_{th} row and the j_{th} column. Assuming the initial memristance values of all memristors are R_{on} and the values $R_f = R_{on}$, and also assuming $V_{Read} = V_{Bias}$, the output values are zero when no training has been done, which specifies the initial condition. In the learning process, memristance of memristors may increase, after which the absolute value of $-V_{Read}(\frac{R_f}{R_{m_{ij}}})$ begins to decrease, resulting in larger v_{o1} values. Therefore, for the first column ($x = 1$), the output voltage will be obtained as follows:

$$v_1 = max(V_{o1}, \dots, V_{o4}) - V_{Don} \quad (9)$$

Using the same hardware as Figure 7 for all columns, voltages will be obtained for each input. We can also use hardware such as weighted average (WA) instead of the connector block as shown in Figure 7.

4. 1. The Defuzzification Stage After type-reduction, the resulting outputs (voltage) indicate the membership degree in the corresponding inputs. The output is a vector, which is considered as a fuzzy number, so the output should be transformed into a crisp number

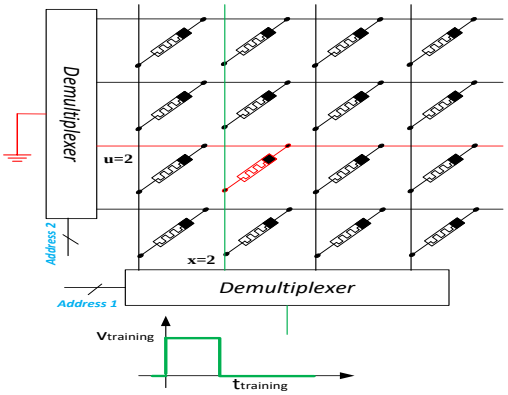

Figure 6. Selecting the desired memristor using de-multiplexers while applying a learning pulse (assumption of training for the second row ($u = 2$) and second column ($x = 2$))

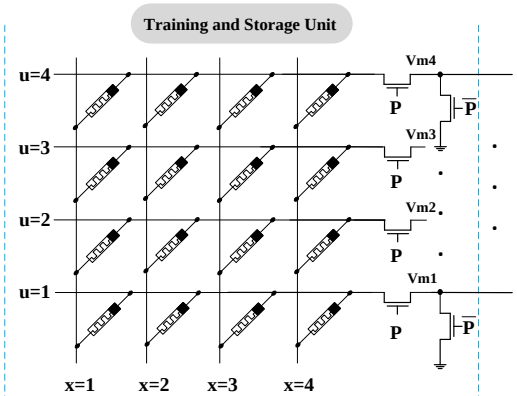

Figure 5. Proposed memristor-crossbar hardware for storing, learning, and testing operations

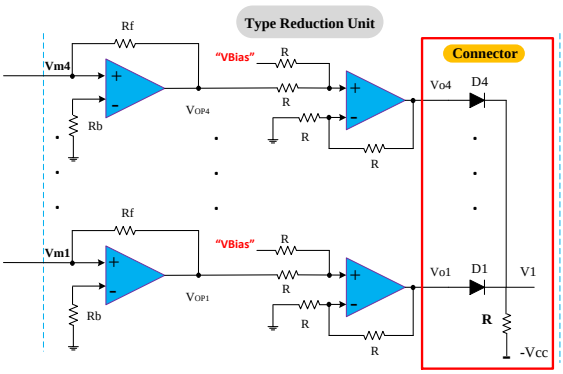

Figure 7. The type-reduction circuit, assuming the output v_1 for this circuit is $V_1 = max(V_{o1}, \dots, V_{o4}) - V_{Don}$

by an algorithm such as a WA. To implement the WA section as the defuzzification section, the hardware circuit that is shown in Figure 8 is used. The outputs v_{w1} and v_{w2} are calculated for n pairs $(x_i, \mu(x_i))$ according to the following equations.

$$v_{w1} = -(\mu_1 \times \frac{n\times R}{\frac{n}{1}\times R} + \mu_2 \times \frac{n\times R}{\frac{n}{2}\times R} + \cdots + \mu_n \times \frac{n\times R}{\frac{n}{n}\times R}) = -(\mu_1 + \mu_2 \times 2 + \cdots + \mu_n \times n) = -\sum_{i=1}^{n} \mu_i \times i = -\sum_{i=1}^{n} \mu_i \times x_i \quad (10)$$

$$v_{w2} = -(\mu_1 + \mu_2 + \cdots + \mu_n) = -\sum_{i=1}^{n} \mu_i \quad (11)$$

The obtained outputs are applied into an analog voltage divider, and the output is obtained according to the following equation.

$$v_{wA} = -\sum_{i=1}^{n} \mu_i \times x_i \,/ -\sum_{i=1}^{n} \mu_i \quad (12)$$

Voltage v_{wA} will be a number between 1 and the quantization level (n) of the input. This voltage can be converted to an analog voltage using a digital to analog converter (DAC).

It should be noted, the number of memristors can be increased in order to improve the accuracy, i.e., by increasing the number of quantization levels, the accuracy can be improved. In this implementation, the values are coded to memristance of memristors, so any desired value can be programmed by applying the appropriate voltage on memristors.

5. SIMULATION RESULTS

For investigating the accuracy of the proposed hardware, we conducted a T2F as shown in Figure 9. We have simulated the proposed circuit in HSPICE 2009 with TSMC 350nm technology. In this simulation, the memristor model proposed by Biolek et al. [37] has been used. Before simulating this hypothetical function, the first step is to examine the performance of each part of the circuit. One of the most important elements in the proposed circuit is the Op-Amp, the frequency responses of the designed Op-Amp are shown in Figures 10 and 11.

As shown in those figures, the bandwidth of the Op-Amp is 346 MHz and has a phase margin of 65°. Also, the DC gain of this amplifier is about 78 dB. The output switching of this Op-Amp is from -0.86 to 0.86 volt, which will determine and limit the output changes in the proposed circuit. The values of the parameters used in the designed circuit are given in the Table 1.

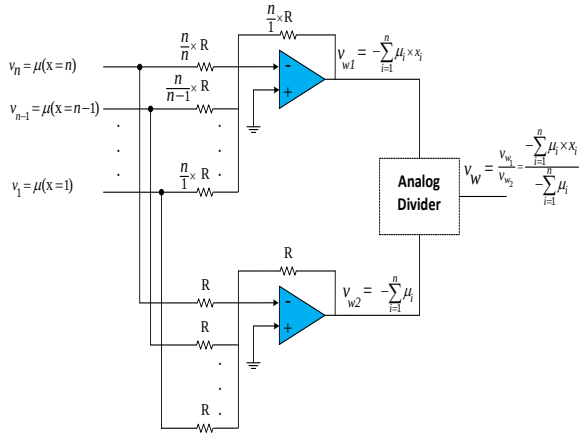

Figure 8. Hardware implementation of the WA method as the defuzzification procedure for $R_x = n$

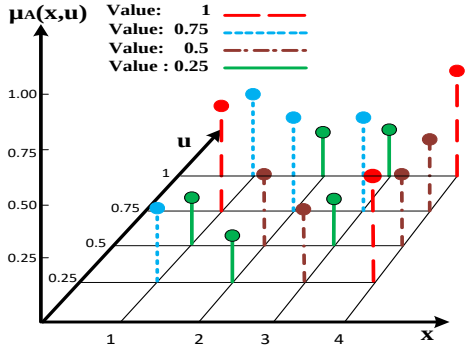

Figure 9. Example of an T2F for discrete universe of discourse

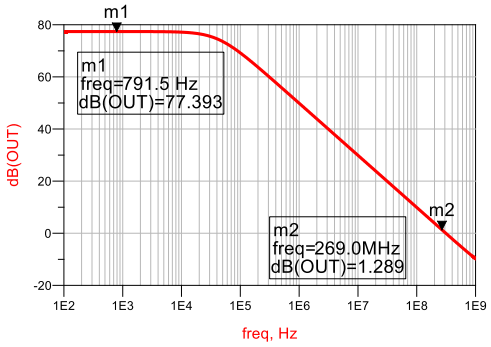

Figure 10. Frequency response of designed Op-Amp (amplitud)

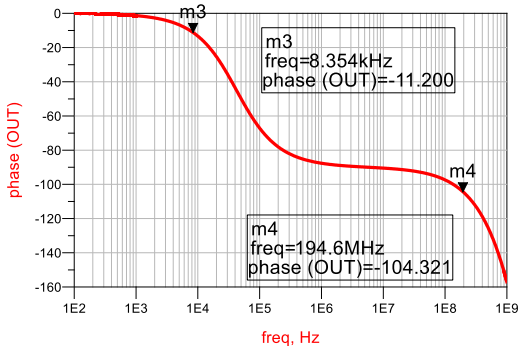

Figure 11. Frequency response of designed Op-Amp (Phase)

TABLE 1. The values of the parameters used in the designed circuit.

# Parameters	Parameter	Value
1	D	$10 * 10^{-9}\ m$
2	μ_v	$5 * 10^{-14}\ \frac{m^2}{VS}$
3	R_{on}	100Ω
4	R_{off}	$16\ K\Omega$
5	R_b	$10\ K\Omega$
6	R	$1\ K\Omega$
7	R_f	$8\ K\Omega$

To read the memristor state and encode it to voltage, the V_{Read} must be applied very quickly (similar to the impulse function) so that the memristor state does not change. The pulse width is about $1\mu s$, which can be easily produced by digital and analog systems.

The voltage of Op-Amp input (v_{m1}) is zero due to the virtual ground of the positive base of the Op-Amp and the negative feedback. In the first stage, the input voltage value is given to the Op-Amp according to Figure 12. The resulted Op-Amp output is shown in Figure 13.

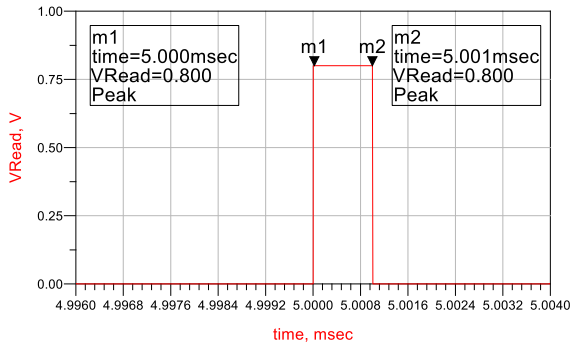

Figure 12. Reading voltage (V_{Read}) for training and pulse width

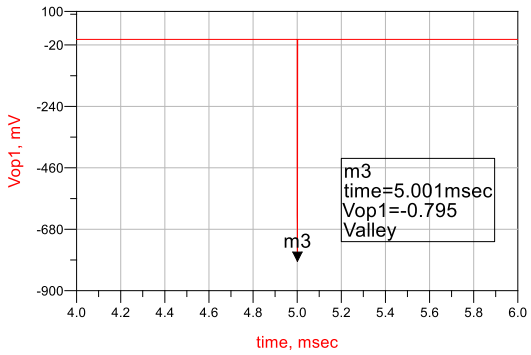

Figure 13. The output voltage propotional to the first stage of Op-Amp

The value of circuit parameters are designed in a way that the output of the second Op-Amp stage be zero, when no training is done. The voltage of V_{Bias} is equal to the input voltage V_{Read} and the values of the voltages are shown in the Figures 14 and 15.

After performing the training process and applying training pulses with the appropriate pulse width, the values stored in the memristor-crossbar structure is shown as Figure 16.

As shown in those figures, the outputs of Figures 14 and 15 are in accordance with Equations (7) and (8).

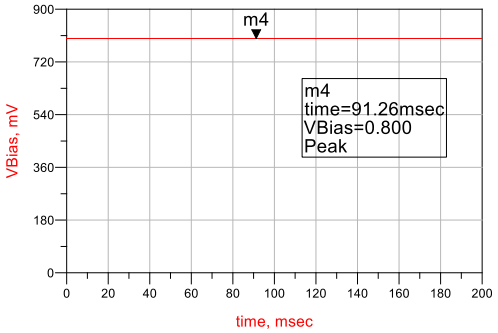

Figure 14. Bias node voltage waveform (V_{Bias})

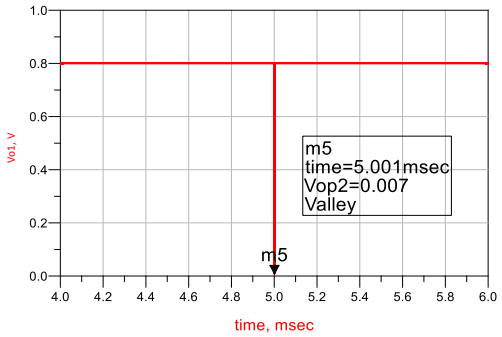

Figure 15. The output voltage propotional to the first stage of Op-Amp

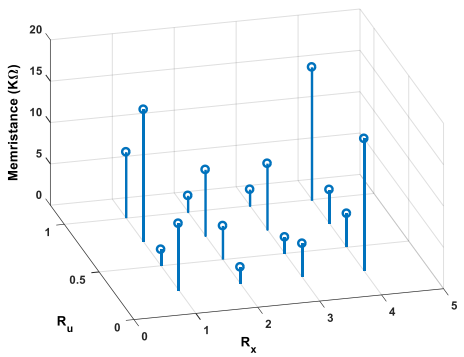

Figure 16. Memristance of memristors in memristor-crossbar

Also, the pattern in Figure 16 is similar to Figure 9. Therefore, it can be claimed that the implementation has been done with great accuracy.

Finally, it should be mentioned that providing the hardware that has learning ability is another challenge that Bi-Inspired systems are facing. The learning capability enables the hardware to be used for a variety of applications. Today, the parameters are mainly obtained by a host system, and then these parameters are implemented on the hardware. Therefore, the challenge of learning on independent hardware is considered in this paper. Moreover, in our proposed hardware, the changes in the memristance of memristors can be done at any time of system implementation while the membership degree values are programmed in the memristance values of memristors.

6. CONCLUSION

Due to the development and growing use of soft computing methods to solve complex engineering problems, it is necessary to provide hardware platforms souitable for the implementation of these algorithms. Memristor is a resistor with memory where memristor-crossbar structures are used to implement artificial neural networks, fuzzy systems, and neuro-fuzzy systems. Therefore, the goal of a suitable hardware platform has been achieved to some extent using nanoscale memristor-crossbar structures. In this paper, first, the T2F membership functions were investigated, and then the memristive hardware related to this type of fuzzy set was presented. Finally, it can be noted that, utilizing the proposed hardware structure, any number of membership functions with any shape and resolution can be implemented without the need for a host system. This feature become more important in adaptive systems. Future work includes the complete hardware implementation of a type-2 fuzzy computing system.

7. REFERENCES

1. Tanaka, K. and Wang, H.O., "Fuzzy control systems design and analysis: A linear matrix inequality approach, John Wiley & Sons, (2004).
2. Terano, T., Asai, K. and Sugeno, M., "Applied fuzzy systems, Academic Press, (2014).
3. Zadeh, L.A., Fu, K.-S. and Tanaka, K., "Fuzzy sets and their applications to cognitive and decision processes: Proceedings of the us–japan seminar on fuzzy sets and their applications, held at the university of california, berkeley, california, july 1-4, 1974, Academic press, (2014).
4. Srikrishna, A., Reddy, B.E. and Srinivas, V.S., Detection of lesion in mammogram images using differential evolution based automatic fuzzy clustering, in Computational intelligence techniques in health care. 2016, Springer.61-68. doi.org/10.1007/978-981-10-0308-0_5
5. Klidbary, S.H., Shouraki, S.B., Ghaffari, A. and Kourabbaslou, S.S., "Outlier robust fuzzy active learning method (alm)", in 2017 7th international conference on computer and knowledge engineering (ICCKE), IEEE. (2017), 347-352. doi: 10.1109/ICCKE.2017.8167903
6. Mendel, J.M., Uncertain rule-based fuzzy systems, in Introduction and new directions. 2017, Springer.684. doi.org/10.1007/978-3-319-51370-6
7. Mendel, J.M. and John, R.B., "Type-2 fuzzy sets made simple", ***IEEE Transactions on Fuzzy Systems***, Vol. 10, No. 2, (2002), 117-127, doi: 10.1109/91.995115.
8. John, R.I., Innocent, P.R. and Barnes, M., "Neuro-fuzzy clustering of radiographic tibia image data using type 2 fuzzy sets", ***Information Sciences***, Vol. 125, No. 1-4, (2000), 65-82, doi.org/10.1016/S0020-0255(00)00009-8.
9. Ozen, T. and Garibaldi, J.M., "Investigating adaptation in type-2 fuzzy logic systems applied to umbilical acid-base assessment", in Proceedings of 2003 European Symposium on Intelligent Technologies (EUNITE 2003). (2003), 289-294.
10. Figueroa, J., Posada, J., Soriano, J., Melgarejo, M. and Rojas, S., "A type-2 fuzzy controller for tracking mobile objects in the context of robotic soccer games", in The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ'05., IEEE. (2005), 359-364. doi: 10.1109/FUZZY.2005.1452420
11. Melin, P., Mendoza, O. and Castillo, O., "An improved method for edge detection based on interval type-2 fuzzy logic", ***Expert Systems with Applications***, Vol. 37, No. 12, (2010), 8527-8535, doi.org/10.1016/j.eswa.2010.05.023.
12. Kim, D., " An implementation of fuzzy logic controller on the reconfigurable fpga system", ***IEEE Transactions on industrial Electronics***, Vol. 47, No. 3, (2000), 703-715, doi: 10.1109/41.847911.
13. Klidbary, S.H., Shouraki, S.B. and Linares-Barranco, B., "Digital hardware realization of a novel adaptive ink drop spread operator and its application in modeling and classification and on-chip training", ***International Journal of Machine Learning and Cybernetics***, Vol. 10, No. 9, (2019), 2541-2561, doi: doi.org/10.1007/s13042-018-0890-x.
14. Hung, D.L., "Dedicated digital fuzzy hardware", ***IEEE Micro***, Vol. 15, No. 4, (1995), 31-39, doi: 10.1109/40.400640.
15. Chua, L.O. and Kang, S.M., "Memristive devices and systems", ***Proceedings of the IEEE***, Vol. 64, No. 2, (1976), 209-223, doi: 10.1109/PROC.1976.10092..
16. Chua, L., "Memristor-the missing circuit element", ***IEEE Transactions on Circuit Theory***, Vol. 18, No. 5, (1971), 507-519, doi: 10.1109/TCT.1971.1083337.
17. Strukov, D.B., Snider, G.S., Stewart, D.R. and Williams, R.S., "The missing memristor found", ***Nature***, Vol. 453, No. 7191, (2008), 80-83, doi: doi.org/10.1038/nature06932.
18. Waser, R. and Aono, M., Nanoionics-based resistive switching memories, in Nanoscience and technology: A collection of reviews from nature journals. 2010, World Scientific.158-165.
19. Kuekes, P., "Material implication: Digital logic with memristors", in Memristor and memristive systems symposium. Vol. 21, (2008).
20. Tarkhan, M., Maymandi-Nejad, M., Haghzad Klidbary, S. and Bagheri Shouraki, S., "A bridge technique for memristor state programming", ***International Journal of Electronics***, Vol. 107, No. 6, (2020), 1015-1030, doi.org/10.1080/00207217.2019.1692371.
21. Mouttet, B., " Proposal for memristors in signal processing", in International Conference on Nano-Networks, Springer. Vol., No., (2008), 11-13. doi: 10.1007/978-3-642-02427-6_3
22. Merrikh-Bayat, F. and Shouraki, S.B., "Memristor-based circuits for performing basic arithmetic operations", ***Procedia Computer***

Science, Vol. 3, (2011), 128-132, doi.org/10.1016/j.procs.2010.12.022.

23. Amer, S., Madian, A.H. and Emara, A.S., " Memristor-based center-of-gravity (COG) defuzzifier circuit", in 2015 European Conference on Circuit Theory and Design (ECCTD), IEEE. (2015), 1-4. doi: 10.1109/ECCTD.2015.7300099
24. Amer, S., Madian, A., ElSayed, H. and Emara, A., "Effect of the memristor threshold current on memristor-based min-max circuits", in 2016 5th International Conference on Modern Circuits and Systems Technologies (MOCAST), IEEE. (2016), 1-4. doi: 10.1109/MOCAST.2016.7495104
25. Merrikh-Bayat, F., Shouraki, S.B. and Merrikh-Bayat, F., "Memristor crossbar-based hardware implementation of fuzzy membership functions", in 2011 eighth international conference on fuzzy systems and knowledge discovery (FSKD), IEEE. Vol. 1, (2011), 645-649.
26. Merrikh-Bayat, F. and Shouraki, S.B., "Memristive neuro-fuzzy system", ***IEEE Transactions on Cybernetics***, Vol. 43, No. 1, (2012), 269-285, doi.
27. Merrikh-Bayat, F., Shouraki, S.B. and Rohani, A., "Memristor crossbar-based hardware implementation of the ids method", ***IEEE Transactions on Fuzzy Systems***, Vol. 19, No. 6, (2011), 1083-1096, doi: 10.1109/TFUZZ.2011.2160024.
28. Klidbary, S.H. and Shouraki, S.B., "A novel adaptive learning algorithm for low-dimensional feature space using memristor-crossbar implementation and on-chip training", ***Applied Intelligence***, Vol. 48, No. 11, (2018), 4174-4191, doi.org/10.1007/s10489-018-1202-6.
29. Klidbary, S.H., Shouraki, S.B. and Afrakoti, I.E.P., "An adaptive efficient memristive ink drop spread (IDS) computing system", ***Neural Computing and Applications***, Vol. 31, No. 11, (2019), 7733-7754, doi.org/10.1007/s00521-018-3604-0.
30. Afrakoti, I.E.P., Shouraki, S.B., Bayat, F.M. and Gholami, M., "Using a memristor crossbar structure to implement a novel adaptive real-time fuzzy modeling algorithm", ***Fuzzy Sets and Systems***, Vol. 307, (2017), 115-128, doi.org/10.1016/j.fss.2016.02.016.
31. Adhikari, S.P., Yang, C., Kim, H. and Chua, L.O., "Memristor bridge synapse-based neural network and its learning", ***IEEE Transactions on Neural Networks and Learning Systems***, Vol. 23, No. 9, (2012), 1426-1435, doi: 10.1109/TNNLS.2012.2204770.
32. Hasan, R., Taha, T.M. and Yakopcic, C., "On-chip training of memristor crossbar based multi-layer neural networks", ***Microelectronics Journal***, Vol. 66, (2017), 31-40, doi.org/10.1016/j.mejo.2017.05.005.
33. Li, T., Duan, S., Liu, J., Wang, L. and Huang, T., "A spintronic memristor-based neural network with radial basis function for robotic manipulator control implementation", ***IEEE Transactions on Systems, Man, and Cybernetics: Systems***, Vol. 46, No. 4, (2015), 582-588, doi: 10.1109/TSMC.2015.2453138.
34. Alibart, F., Zamanidoost, E. and Strukov, D.B., "Pattern classification by memristive crossbar circuits using ex situ and in situ training", ***Nature Communications***, Vol. 4, No. 1, (2013), 1-7, doi: 10.1038/ncomms3072.
35. Karnik, N.N. and Mendel, J.M., "Introduction to type-2 fuzzy logic systems", in 1998 IEEE international conference on fuzzy systems proceedings. IEEE world congress on computational intelligence (Cat. No. 98CH36228), IEEE. Vol. 2, (1998), 915-920.
36. Karnik, N.N., Mendel, J.M. and Liang, Q., "Type-2 fuzzy logic systems", ***IEEE Transactions on Fuzzy Systems***, Vol. 7, No. 6, (1999), 643-658, doi: 10.1109/FUZZY.1998.686240.
37. Biolek, D., Di Ventra, M. and Pershin, Y.V., "Reliable spice simulations of memristors, memcapacitors and meminductors", ***arXiv preprint arXiv:1307.2717***, (2013).

Persian Abstract

چکیده

استفاده از تکنیک های فازی، به ویژه فازی نوع-۲، یکی از روش های بسیار پرکاربرد در یادگیری ماشین برای مدل سازی عدم قطعیت است. علاوه بر ارائه الگوریتم، قابلیت پیاده سازی سخت افزاری و همچنین عملکرد مناسب در کاربردهای زمان واقعی از چالش های مهم هستند. استفاده از بسترهای سخت افزاری که شباهت های بیولوژیکی دارند و از نظر حجم پیاده سازی با سیستم های عصبی انسان قابل مقایسه هستند همیشه مورد توجه بوده است. ممریستور یکی المان های نوظهور برای پیاده سازی الگوریتم های مبتنی بر منطق فازی است. در این المان، با اعمال جریان و انتخاب جهت مناسب برای جریان اعمال شده ، مقاومت ممریستور (ممریستنس) افزایش یا کاهش می یابد. پیاده سازی های مختلفی از سیستم های فازی نوع-۱ وجود دارد، اما هیچ پیاده سازی از سیستم های فازی نوع-۲ مبتنی بر مموریستور انجام نشده است. در این مقاله، از ساختارهای متقاطع ممریستور برای پیاده سازی توابع عضویت فازی نوع-۲ استفاده شده است. در سخت افزار پیشنهادی، توابع عضویت می توانند از هر شکل و رزولوشنی برخوردار باشند. سخت افزاری پیشنهادی که برای توابع عضویت فازی نوع-۲ ارائه شده است، قابلیت یادگیری دارد (قابلیت یادگیری روی تراشه بدون نیازی به سیستم میزبان). علاوه بر این، سخت افزار پیشنهادی آنالوگ بوده و می تواند به عنوان مبنایی در ساخت سیستم های تکاملی مورد استفاده قرار گیرد. ضمناً، برای بررسی عملکرد سخت افزار پیشنهادی، شبیه سازی هایی مبتنی بر ممریستورانجام داده شده است.