 CMES, vol.115, no.3, pp.295-311, 2018

Modeling the Spike Response for Adaptive Fuzzy Spiking Neurons with Application to a Fuzzy XOR

A. M. E. Ramírez-Mendoza[1, *]

Abstract: A spike response model (SRM) based on the spikes generator circuit (SGC) of adaptive fuzzy spiking neurons (AFSNs) is developed. The SRM is simulated in Matlab™ environment. The proposed model is applied to a configuration of a fuzzy exclusive or (fuzzy XOR) operator, as an illustrative example. A description of the comparison of AFSNs with other similar methods is given. The novel method of the AFSNs is used to determine the value of the weights or parameters of the fuzzy XOR, first with dynamic weights or self-tuning parameters that adapt continuously, then with fixed weights obtained after training, finally with fixed weights and a dynamic gain or self-tuning gain for a fine adjustment of amplitude.

Keywords: Spike response model, spikes generator circuit, fuzzy XOR, adaptive fuzzy spiking neuron, learning algorithm, fuzzy neuron, self-tuning.

1 Introduction

The artificial neurons are mathematical models inspired by the functioning of biological neurons. These had been developed for several decades [McCulloch and Pitts (1943); Widrow, Lehr and Michael (1990); Gupta and Qi (1992); Gupta (1993); Gupta and Rao (1994); Pérez and Ramírez (2001); Ramírez and Pérez (2002); Ghosh-Dastidar and Adeli (2009); Pérez, Garcés, Cabiedes et al. (2009); Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011); Ramírez-Mendoza (2014); Ramírez-Mendoza (unpublished); Ramírez-Mendoza (unpublished)]. Among the operations artificial neurons can perform with the purpose of imitating the neuronal characteristics are memory, learning, axonic delay time, the refractory period and the generation of spikes.

The models of fuzzy neurons are based on the human reasoning, the approximate reasoning and fuzzy logic [Zadeh (1977); Lara-Rosano (2017)]. Also, different models of neural networks and learning methods have been developed [Widrow, Lehr and Michael (1990); Gupta and Qi (1992); Gupta (1993); Gupta and Rao (1994); Pérez and Ramírez (2001); Ramírez and Pérez (2002); Pérez, Garcés, Cabiedes et al. (2009); Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011); Ramírez-Mendoza (2014); Ramírez-Mendoza (unpublished)], as spiking neural networks [Ghosh-Dastidar and Adeli (2009); Chaturvedi

---

[1] CONACYT Research Fellow–State University of Nuevo León (UANL for its acronym in Spanish), Mechanical and Electrical Engineering School (FIME for its acronym in Spanish), Center for Research and Innovation in Aeronautical Engineering (CIIIA for its acronym in Spanish), Carretera a Salinas Victoria km. 2.3, Apodaca, Nuevo León, México.

[*] Corresponding Author: Abigail María Elena Ramírez Mendoza. Email: abigail.ramirezmn@uanl.edu.mx; ameramirezme@conacyt.mx.

(2011); Ramírez-Mendoza (unpublished); Ramírez-Mendoza (unpublished)], with applications in various areas.

The application areas are control [Yu and Rosen (2013)], instrumentation and measurement [Jiang, Guo and Zhang (2015)], electrophysiological recordings [Rossant, Goodman, Platkiewicz et al. (2010)], pattern recognition [Espinosa-Ramos, Cruz-Cortes, and Vazquez (2013)], and Drones and Induction Motors [Ramírez-Mendoza, Covarrubias-Fabela, Amezquita-Brooks et al. (2018)].

Also, neural networks (NNs) had been applied for stochastic analysis of composite materials for aeronautical and aerospace structures of aircrafts, UAVs, helicopters, improving the efficiency of the simulations of the effects of mechanical vibrations, instead of traditional methods such as the algorithm of the finite element [Tawfik, Bishay and Sadek (2018)].

Other application of artificial neural networks (ANNs) with an optimization algorithm is the prediction of the modulus of rupture values of glass fiber reinforced concrete panels as a preliminary criterion for quality check of fabricated products, the results presented in [Yildizel and Öztürk (2016)], exhibit ANNs with back propagation algorithm models have more fitting performance compared with a classic multiple linear regression (MLR) model.

The spiking neuron models are popular for studies of memory, neural coding, neural network dynamics and reproduce spiking behavior of cortical neurons or fast spiking cortical interneurons [Gerstner and Kistler (2002); Izhikevich (2003)]. The simple SRM proposed here is like that of the nonlinear integrate and fire model, the parameters depend on the time since the last output spike, however, the formulation of the equations is different. The main contribution of this paper is a SRM based on a SGC [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011)], of Adaptive Fuzzy Spiking Neurons (AFSNs) with a learning algorithm developed in Ramírez-Mendoza et al. [Ramírez-Mendoza (2014); Ramírez-Mendoza (unpublished); Ramírez-Mendoza (unpublished)].

Spiking neural networks (SNNs) had been developed for imitating the behavior of biological neurons, the ANNs, are based on the training of the weights, but the SNNs also include a delay parameter, the signal delay from one neuron to another is different, giving more flexibility and an additional parameter for the training of the SNN, with applications in pattern recognition and image processing [Rosado-Muñoz, Bataller-Mompeán and Guerrero-Martínez (2012)].

The SRM of the AFSNs proposed in this work, presents also the neuronal characteristics such as the refractory time, axonal delay and the generation of spike trains. An important point is that neuronal characteristics such as refractory time and axonal delay are performed before the generation of the spike trains and therefore, the shape of the spikes remains along the neural network for neuronal coding [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011)].

Learning systems with a hardware spike timing dependent plasticity (STDP) learning rule applied to memristor-based NN, had been implemented in discrete-time where spikes are expressed as pulses, the neurons are synchronous with a global clock [Zheng and Mazumder (2018)], the results demonstrate the efficacy of the learning algorithms.

A SNN third generation neural network model presented in Lashkare et al. [Lashkare, Chouhan, Chavan et al. (2018)], with hardware realization for high integration, the spike frequency depends on the input current. The spike trains frequency of the SGC presented in Ramírez-Mendoza et al. [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011)], is proportional to the amplitude of the response of the activation function of the AFSN.

The operator XOR is very important in the computational programming, it performs a primitive operation used in different encryption algorithms [Bedregal, Reiser and Dimuro (2009)]. Therefore, a fuzzy XOR is proposed as a primitive fuzzy operator to perform different configurations with AFSNs and thus model systems and processes.

As an illustrative example of the good functioning of the spike model, a configuration of a fuzzy XOR is proposed. The fuzzy XOR is very important for the comprehension and future applications of AFSNs in different areas of knowledge such as systems identification, experimental aerodynamics, industrial applications [Ramírez-Mendoza, Covarrubias-Fabela, Amezquita-Brooks et al. (2018)], process control in the design of control rules for proportional integral derivative controllers (PIDs) [Ramírez-Mendoza (unpublished); Ramírez-Mendoza (unpublished)], and navigation systems and trajectory tracking of low-scale unmanned aerial vehicles (UAVs) [Álvarez, Olascoaga, Rivera et al. (2017); Amezquita-Brooks, Liceaga-Castro, González-Sánchez et al. (2017)], among others. The following section presents a description of the AFSNs and the main contribution of this paper the modeling of the spike response of the AFSNs. The complete development of the AFSNs is presented in Ramírez-Mendoza et al. [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011); Ramírez-Mendoza (2014); Ramírez-Mendoza (unpublished); Ramírez-Mendoza (unpublished); Ramírez-Mendoza, Covarrubias-Fabela, Amezquita-Brooks et al. (2018)].

2 Adaptive fuzzy spiking neurons description

The AFSNs are fuzzy neurons with synaptic and somatic operations. Fig. 1 shows the block diagram of the AFSN with crisp or non-fuzzy inputs. The AFSN model has dendritic input signals $z_{inj}(k) \in [0,\ 1]$ for unipolar signals, and $z_{inj}(k) \in [-1,\ 1]$ for bipolar signals.

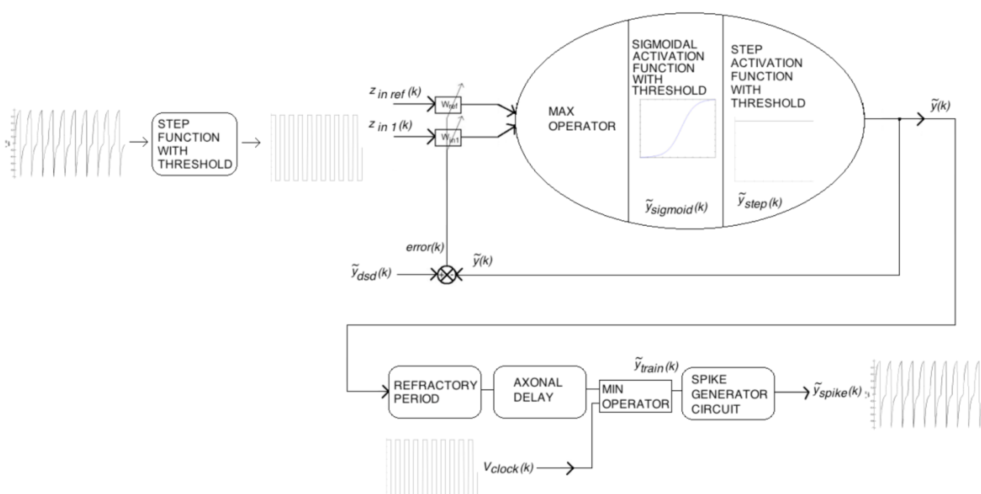

Figure 1: Block diagram of the AFSN model

2.1 Somatic aggregation operation

The somatic aggregation operation is performed according to Gupta [Gupta and Qi (1992); Gupta (1993); Gupta and Rao (1994)], as a Gupta fuzzy integrator (GFI) or generalized fuzzy OR of N inputs formed by N-1 fuzzy OR operators (MAX operators), Eq. (1).

$$\tilde{V}_{max}(\mathrm{k}) = MAX_{j=1}^{N}\min(z_{inj}(k), w_{inj}(k)) \tag{1}$$

Where:

k is the time variable.

$z_{inj}(k)$ are the dendrite inputs.

$w_{inj}(k)$ are the synaptic weights.

$\tilde{V}_{inj}(\mathrm{k}) = \min(z_{inj}(k), w_{inj}(k))$ are the synaptic operations.

2.2 Non-linear somatic operation

For the non-linear somatic operation of the AFSN model, a sigmoid activation function with threshold and a learning algorithm were developed in Ramírez-Mendoza et al. [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011); Ramírez-Mendoza (2014); Ramírez-Mendoza (unpublished); Ramírez-Mendoza (unpublished)], the activation function is defined by Eq. (3),

$$\tilde{V}_{out}(\mathrm{k}) = \max(\tilde{V}_{max}(\mathrm{k}), \lambda_1(k)) \tag{2}$$

- For bipolar signals,

$$\tilde{y}_{sigmoid}(k) = \frac{2}{1+e^{\left(-min\left(\gamma,\tilde{V}_{out}(k)\right)\cdot a\right)}} - 1 \tag{3}$$

Where

$\lambda_1(k)$ is a threshold signal.

$-1 \leq \gamma \leq 1$, is the somatic gain that determines the slope of the activation function, and the learning factor that regulates the speed of learning.

a is a real number, usually, $a > 0$.

- For AFSNs without input spikes, only no fuzzy inputs, and no learning algorithm, $\tilde{y}(\mathrm{k})$ is calculated with Eq. (4) and $\tilde{y}_{train}(\mathrm{k})$ with Eq. (6).

$$\tilde{y}(\mathrm{k}) = \begin{cases} 1 & if\ \tilde{y}_{sigmoid}(k) \geq \lambda_2(k) \\ -1 & if\ \tilde{y}_{sigmoid}(k) < \lambda_2(k) \end{cases} \tag{4}$$

$$\tilde{y}_{trainp}(\mathrm{k}) = \min(\tilde{y}(\mathrm{k}),\ V_{clock}(k)) \tag{5}$$

$$\tilde{y}_{train}(\mathrm{k}) = \begin{cases} 1 & if\ \tilde{y}_{trainp}(k) > 0 \\ -1 & if\ \tilde{y}_{trainp}(k) \leq 0 \end{cases} \tag{6}$$

Where:

$\tilde{y}(\mathrm{k})$ is a fuzzy mapping of the Gupta fuzzy sum-integral $\tilde{V}_{max}(\mathrm{k})$.

$\lambda_2(k)$ is a threshold signal, for example, triangular shape.

$V_{clock}(k)$ is the clock signal.

- For AFSNs with spikes as inputs, that is, pulses as inputs after a step function with threshold, and learning algorithm, $\tilde{y}(\mathrm{k})$ is calculated with Eq. (7).

$$\tilde{y}(\mathrm{k}) = \tilde{y}_{sigmoid}(k) \tag{7}$$

$$\tilde{y}_{train}(\mathrm{k}) = \begin{cases} V_{clock}(k) & if \;\; \tilde{y}(\mathrm{k}) > V_{ref}(k) \\ -1 & if \;\; \tilde{y}(\mathrm{k}) \leq V_{ref}(k) \end{cases} \tag{8}$$

Where

$V_{ref}(k)$ is a reference voltage

Because of the spikes are bipolar signals for $\tilde{y}_{trainbi}(\mathrm{k})$,

$$\tilde{y}_{trainbi}(\mathrm{k}) = \begin{cases} +V & if \;\; \tilde{y}_{train}(\mathrm{k}) > V_{ref}(k) \\ -V & if \;\; \tilde{y}_{train}(\mathrm{k}) \leq V_{ref}(k) \end{cases} \tag{9}$$

Where,

+V is a positive saturation voltage

-V is a negative saturation voltage

$V_{ref}(k)$ is a reference voltage

The error for the adaptive learning algorithm for AFSN model with a Gupta integrator, is defined in Eq. (10).

$$e(k) = \tilde{y}_{dsd}(\mathrm{k}) - \tilde{y}_{trainbi}(\mathrm{k}) \tag{10}$$

Where,

$\tilde{y}_{dsd}(\mathrm{k})$ is the desired output of AFSN_j and is the $\tilde{y}_{trainbi}(\mathrm{k})$ of some other AFSN.

The learning algorithm for bipolar signals requires a reference input to each AFSN, the input reference value is the maximum value in the interval [-1, 1] for bipolar signals. The ideal values of the weights are unknown, even though, are calculated by the adaptive learning algorithm, based on proposed initial values.

2.3 Spike model based on the spikes generator circuit

Based on Eq. (9), the spikes are obtained according to Eq. (11), the mathematical model of the SGC [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011)],

$$\tilde{y}_{spike}(\mathrm{k}) = R_3 \cdot I_s \cdot \left(e^{\frac{K \cdot V_D(k)}{T_K}} - 1\right) - \frac{R_3}{R_1} \cdot \tilde{y}_{trainbi}(\mathrm{k}) - \left(\frac{R_2}{R_1} + 1\right) \cdot R_3 \cdot C_1 \cdot \frac{d\tilde{y}_{trainbi}(\mathrm{k})}{dk} \tag{11}$$

A spike model is developed based on the SGC Fig. 1 and Eq. (11). The positive part of the spike model [Desoer and Kuh (1969); Esfandiari and Lu (2014)], is defined by the equations:

$$\frac{d\tilde{y}_{spike}(\mathrm{k})}{dk} + \left(\frac{(R_2+R_3)}{R_2 \cdot R_3 \cdot C}\right) \cdot \tilde{y}_{spike}(\mathrm{k}) = \frac{R_2 \cdot I_D(k) + \tilde{y}_{trainbi}(\mathrm{k})}{R_2 \cdot C} \tag{12}$$

$$\tilde{y}_{spike}(\mathrm{k}) = \frac{(R_2 \cdot I_D(k) + \tilde{y}_{trainbi}(\mathrm{k})) \cdot R_3}{R_2 + R_3} \cdot \left(1 - e^{-\left(\frac{R_2 + R_3}{R_2 \cdot R_3 \cdot C}\right) \cdot k}\right) \tag{13}$$

The negative part of the spike model [Esfandiari and Lu (2014)], is defined by the equations:

$$\frac{d\tilde{y}_{spike}(\mathrm{k})}{dk} + \left(\frac{1}{\frac{(R_1+R_2)\cdot R_3\cdot C}{R_1+R_2+R_3}}\right)\cdot \tilde{y}_{spike}(\mathrm{k}) \; = 0 \tag{14}$$

$$\tilde{y}_{spike}(\mathrm{k}) \; = \; \tilde{y}_{trainbi}(\mathrm{k}) \cdot e^{-\left(\frac{1}{\frac{(R_1+R_2)\cdot R_3\cdot C}{R_1+R_2+R_3}}\cdot k\right)} \tag{15}$$

- Based on Eqs. (13) and (15), if only the spikes are available, $\tilde{y}_{trainbi}(\mathrm{k})$ could be obtained with Eq. (16).

$$\tilde{y}_{trainbi}(\mathrm{k}) = \begin{cases} 1 & if \;\; \tilde{y}_{spike}(\mathrm{k}) \; > \; V_{ref}(k) \\ -1 & if \;\; \tilde{y}_{spike}(\mathrm{k}) \; \leq \; V_{ref}(k) \end{cases} \tag{16}$$

Where,

k is the time variable [seconds].

R_1, R_2, R_3 are resistances [Ω].

C is a capacitance [F].

I_D is the diode current [A].

The comparison between the traditional models of adaptive neurons, fuzzy adaptive neurons and fuzzy adaptive spiking neurons with the AFSN model is shown in Tab. 1.

Table 1: Comparison of Neuron Models

Neuron Model	Learning Algorithm	Fuzzy	Pulse train proportional in frequency to the neuron response.	Spike Response
Neural Networks with back propagation algorithms.	✓			
Fuzzy Adaptive Neurons of Gupta.	✓	✓		
Spiking Neural Networks	✓			✓
Adaptive Fuzzy Spiking Neurons	✓	✓	✓	✓

3 Results of the simulation of the Fuzzy XOR based on the AFSNs

In this section, a configuration for realizing a fuzzy XOR based on the AFSNs is proposed. To obtain the fuzzy XOR, it is proposed that the weights adapt continuously, that is, dynamically, all the parameters are adapted online. Also, with values of fixed weights obtained after training. Finally, with fixed weights and a dynamic gain for a fine adjustment of amplitude and a greater definition in the results. The simulation results of the spike model are presented in Matlab^{TM} environment.

3.1 Fuzzy XOR configuration

As illustrative example, a configuration of a fuzzy XOR is proposed with base on the fuzzy logic, the equation that defines a non-fuzzy XOR is Eq. (17),

$$x \oplus y = (x + y) \cdot (x' + y') \tag{17}$$

Expressing the Eq. (17), with fuzzy operations, Eq. (18),

$$\mu_{X \oplus Y}(x \oplus y) = \min\big(\max(\mu_X(x), \mu_Y(y)), \max(-\mu_X(x), -\mu_Y(y))\big) \;\; \forall\, x, y \in U \tag{18}$$

where X, Y are fuzzy sets of the universe of discourse U.

The proposed configuration of a fuzzy XOR is based on the AFSNs, the block diagram is shown in Fig. 2.

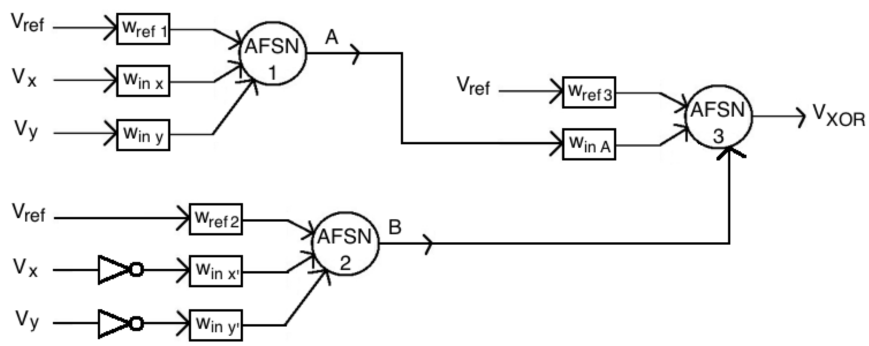

Figure 2: Block diagram of the fuzzy XOR configuration

According to the proposed model of the AFSN [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011); Ramírez-Mendoza (2014); Ramírez-Mendoza (unpublished); Ramírez-Mendoza (unpublished)], the equations that determine the fuzzy XOR are:

$$\tilde{V}_{\max AFSN\,1}(\mathrm{k}) = max\Big(\min\big(V_{ref}(k), w_{ref\,1}(k)\big), \min\big(V_x(k), w_{in\,x}(k)\big), \min\big(V_y(k), w_{in\,y}(k)\big)\Big) \tag{19}$$

$$\tilde{V}_{out\;AFSN\,1}(\mathrm{k}) = \max(\tilde{V}_{\max AFSN\,1}(\mathrm{k}), \lambda_1(k)) \tag{20}$$

$$\tilde{y}_{sigmoid\;AFSN\,1}(k) = \frac{2}{1 + e^{\left(-min\left(\gamma_1, \tilde{V}_{out\;AFSN\,1}(k)\right)\cdot a_1\right)}} - 1 \tag{21}$$

$$\tilde{V}_{\max AFSN\,2}(\mathrm{k}) = max\Big(\min\big(V_{ref}(k), w_{ref\,2}(k)\big), \min\big(-V_x(k), w_{in\,x'}(k)\big), \min\big(-V_y(k), w_{in\,y'}(k)\big)\Big) \tag{22}$$

$$\tilde{V}_{out\;AFSN\,2}(\mathrm{k}) = \max(\tilde{V}_{\max AFSN\,2}(\mathrm{k}), \lambda_2(k)) \tag{23}$$

$$\tilde{y}_{sigmoid\;AFSN\,2}(k) = \frac{2}{1 + e^{\left(-min\left(\gamma_2, \tilde{V}_{out\;AFSN\,2}(k)\right)\cdot a_2\right)}} - 1 \tag{24}$$

$$V_A(k) = \tilde{y}_{sigmoid\;AFSN\,1}(k) \tag{25}$$

$$\tilde{V}_{\max AFSN\,3}(\mathrm{k}) = max\Big(\min\big(V_{ref}(k), w_{ref\,3}(k)\big), \min\big(V_A(k), w_{in\,A}(k)\big)\Big) \tag{26}$$

$$\tilde{V}_{out\;AFSN\,3}(\mathrm{k}) = \max(\tilde{V}_{\max AFSN\,3}(\mathrm{k}), \lambda_{3\,1}(k)) \tag{27}$$

$$\gamma_3(k) = \tilde{y}_{sigmoid\;AFSN\,2}(k) \tag{28}$$

$$\tilde{y}_{sigmoid\;AFSN\,3}(k) = \frac{2}{1 + e^{\left(-min\left(\gamma_3, \tilde{V}_{out\;AFSN\,3}(k)\right)\cdot a_3\right)}} - 1 \tag{29}$$

$$\tilde{V}_{XOR}(k) = \tilde{V}_{x\oplus y}(k) = \tilde{y}_{sigmoid\ AFSN\ 3}(k) \tag{30}$$

$$\tilde{y}_{step\ AFSN\ 3}(\mathrm{k}) = \begin{cases} 1 & if\ \ \tilde{y}_{sigmoid\ AFSN\ 3}(k) \geq \lambda_{3\ 2}(k) \\ -1 & if\ \ \tilde{y}_{sigmoid\ AFSN\ 3}(k) < \lambda_{3\ 2}(k) \end{cases} \tag{31}$$

where $\lambda_{3\ 2}(k)$ is a triangular wave.

$$\tilde{y}_{trainp}(\mathrm{k}) = \min(\tilde{y}_{step\ AFSN\ 3}(\mathrm{k}),\ V_{clock}(k)) \tag{32}$$

$$\tilde{y}_{train}(\mathrm{k}) = \begin{cases} 1 & if\ \ \tilde{y}_{trainp}(k) > 0 \\ -1 & if\ \ \tilde{y}_{trainp}(k) \leq 0 \end{cases} \tag{33}$$

$$\tilde{y}_{trainbi}(\mathrm{k}) = \begin{cases} +V & if\ \ \tilde{y}_{train}(\mathrm{k}) > V_{ref}(k) \\ -V & if\ \ \tilde{y}_{train}(\mathrm{k}) \leq V_{ref}(k) \end{cases} \tag{34}$$

For the positive part of the spike model,

$$\tilde{y}_{spike}(\mathrm{k}) = \frac{(R_2 \cdot I_D(k) + \tilde{y}_{trainbi}(\mathrm{k})) \cdot R_3}{R_2 + R_3} \cdot \left(1 - e^{-\left(\frac{R_2 + R_3}{R_2 \cdot R_3 \cdot C}\right) \cdot k}\right) \tag{35}$$

For the negative part of the spike model,

$$\tilde{y}_{spike}(\mathrm{k}) = \tilde{y}_{trainbi}(\mathrm{k}) \cdot e^{-\left(\frac{1}{\frac{(R_1 + R_2) \cdot R_3 \cdot C}{R_1 + R_2 + R_3}} \cdot k\right)} \tag{36}$$

Table 2: Configuration of the Fuzzy XOR operator

Number of AFSNs	3
AFSNs for Input layer of the Fuzzy XOR operator	2
AFSNs for Output layer of the Fuzzy XOR operator	1
Sigmoidal Activation Function	3
Step Activation Function	1
Spike Generator	1

3.2 Simulation results

The results of the simulations for the fuzzy XOR configuration is presented, first with dynamic parameters, second with static parameters and in third place with the static parameters and a dynamic gain at the output of each neuron before generating the trains of spikes. The dynamic parameters of the fuzzy XOR, are determined by the learning algorithm of AFSNs in line. The static parameters of the fuzzy XOR, are previously determined by training the learning algorithm of AFSNs. The result of the fuzzy XOR with the configuration here proposed, then is applied a step activation function, for obtaining the spike trains for illustrating the SRM.

The input signals $V_x(k)$ and $V_y(k)$ for the example are non-fuzzy and are shown in Fig. 3. For the simulations, the sampling period is $T_{sampling} = 0.1\ \mu s$.

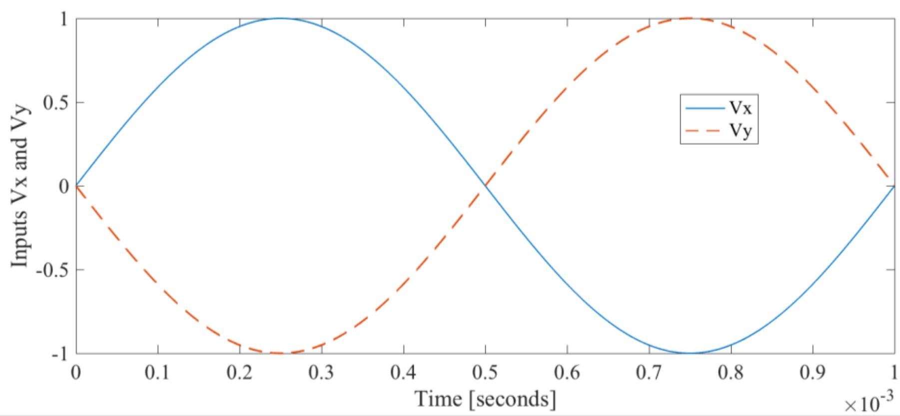

Figure 3: Input signals $\boldsymbol{V_x(k)}$ and $\boldsymbol{V_y(k)}$

3.2.1 Fuzzy XOR configuration with dynamic parameters

The fuzzy XOR $\tilde{V}_{x\oplus y}(k)$ of the input signals $V_x(k)$ and $V_y(k)$ is obtained with the fuzzy XOR configuration proposed here, with the parameters adapting dynamically, three additional neurons are required, for determine the weights $w_{inAFSN_i}(k)$ of the input signals of the three neurons of Fig. 2.
The initial conditions for $k = 0$ are,

- $a = 2$
- $G_{AFSN_3} = 4; \quad \tilde{V}_{XOR}(k) = \tilde{V}_{x\oplus y}(k) = \tilde{y}_{sigmoid\ AFSN\ 3}(k) * G_{AFSN_3}$
- $w_{refAFSN_i}(k) = 0,\ w_{inAFSN_i}(k) = 0$
- $V_{refAFSN_i}(k) = 1,\ k = 0 \dots n$
- $\lambda_{1\ AFSN_i}(k) = -1,\ k = 0 \dots n$
- $\gamma_{AFSN_1}(k) = \gamma_{AFSN_2}(k) = 1.$
- $\gamma_{AFSN_3}(k) = \gamma_3(k) = \tilde{y}_{sigmoid\ AFSN\ 2}(k)$

The result of the simulation $\tilde{V}_{x\oplus y}(k)$ after 7 iterations, is shown in Fig. 4.

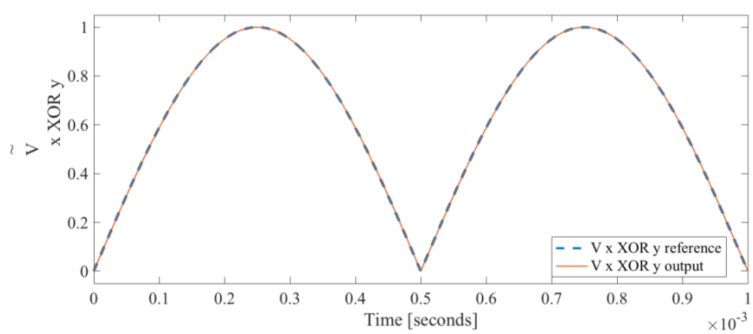

Figure 4: Simulation results of the fuzzy XOR configuration $\boldsymbol{\tilde{V}_{x\oplus y}(k)}$ with dynamic parameters

3.2.2 Fuzzy XOR configuration with static parameters

The result of the simulation of $\tilde{V}_{x\oplus y}(k)$ for the input signals $V_x(k)$ and $V_y(k)$ of the fuzzy XOR configuration with static parameters, obtained by training with three additional and independent neurons $AFSN_4$, $AFSN_5$, $AFSN_6$, to determine the input weights $w_{in\ AFSN_i}(k)$, after 3 iterations, is shown in Fig. 5.
The initial conditions for $k = 0$ are,

- $a = 1$
- $G_{AFSN_3} = 4.4047;\ \tilde{V}_{XOR}(k) = \tilde{V}_{x\oplus y}(k) = \tilde{y}_{sigmoid\ AFSN\ 3}(k) * G_{AFSN_3}$
- $w_{ref AFSN_i}(k) = -1,\ k = 0 \dots n, fuzzy\ XOR\ configuration\ neurons$
- $w_{inAFSN_i}(k) = 0,\ fuzzy\ XOR\ configuration\ neurons$
- $w_{ref AFSN_i}(k) = 1,\ k = 0 \dots n,\ additional\ and\ independent\ neurons$
- $w_{inAFSN_i}(k) = 0,\ additional\ and\ independent\ neurons$
- $V_{ref AFSN_i}(k) = 1,\ k = 0 \dots n$
- $\lambda_{1\ AFSN_i}(k) = -1,\ k = 0 \dots n$
- $\gamma_{AFSN_1}(k) = \gamma_{AFSN_2}(k) = 1,\ k = 0 \dots n, fuzzy\ XOR\ configuration\ neurons$
- $\gamma_{AFSN_3}(k) = \gamma_3(k) = \tilde{y}_{sigmoid\ AFSN\ 2}(k),\ fuzzy\ XOR\ configuration\ neuron$
- $\gamma_{AFSN_4}(k) = \gamma_{AFSN_5}(k) = \gamma_{AFSN_6}(k) = 1,$
  $additional\ and\ independent\ single\ neurons$

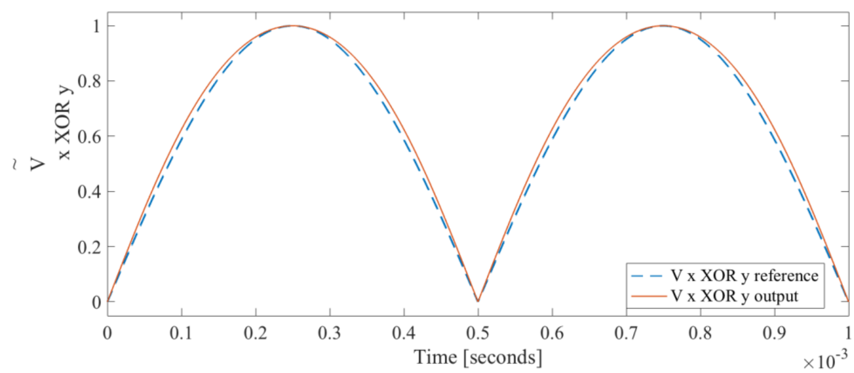

Figure 5: Simulation results of the fuzzy XOR configuration $\tilde{\boldsymbol{V}}_{\boldsymbol{x\oplus y}}(\boldsymbol{k})$ with static parameters

The weights obtained for the fuzzy XOR configuration, after training are:

- $w_{in\ x\ AFSN_1}(k) = w_{in\ y\ AFSN_1}(k) = 1$
- $w_{in\ x'\ AFSN_2}(k) = w_{in\ y'\ AFSN_2}(k) = 1$
- $w_{in\ A\ AFSN_3}(k) = 1$

3.2.3 Fuzzy XOR configuration with static parameters and a dynamic gain in each AFSN

The result of the simulation of $\tilde{V}_{x\oplus y}(k)$ for the input signals $V_x(k)$ and $V_y(k)$ of the fuzzy XOR configuration with static parameters previously obtained in Section 3.2.2, and a

dynamic gain in each AFSN determined with an additional and independent neuron $AFSN_4$, after 1 iteration, is shown in Fig. 6.
The initial conditions for $k = 0$ are,

- $a = 1$
- $T_{spike} = T_{clock} = 2\ \mu s$
- $G_{AFSN_i}(k) = \left|w_{inAFSN_4}(k) * 2.3\right| = 0;\ \tilde{y}_{sigmoid\ AFSN\ i}(k) * G_{AFSN_i}, i = 1 \dots 3$
- $w_{refAFSN_i}(k) = -1,\ k = 0 \dots n, fuzzy\ XOR\ configuration\ neurons$
- $w_{inAFSN_i}(k) = 1,\ k = 0 \dots n,\ i = 1 \dots 3,\ fuzzy\ XOR\ configuration\ neurons$
- $w_{refAFSN_4}(k) = 1,\ k = 0 \dots n,\ additional\ and\ independent\ neuron$
- $w_{inAFSN_4}(k) = 0,\ additional\ and\ independent\ neuron$
- $V_{refAFSN_i}(k) = 1,\ k = 0 \dots n, i = 1 \dots 4$
- $\lambda_{1\ AFSN_i}(k) = -1,\ k = 0 \dots n, i = 1 \dots 4$
- $\lambda_{2\ AFSN_3}(k) = \lambda_{3\ 2}(k)$ is a triangular wave of 100 KHz of frequency, amplitude of 1.
- $\gamma_{AFSN_1}(k) = \gamma_{AFSN_2}(k) = 1,\ k = 1 \dots n$
- $\gamma_{AFSN_3}(k) = \gamma_3(k) = \tilde{y}_{sigmoid\ AFSN\ 2}(k)$
- $\gamma_{AFSN_4}(k) = 1, additional\ and\ independent\ neuron$

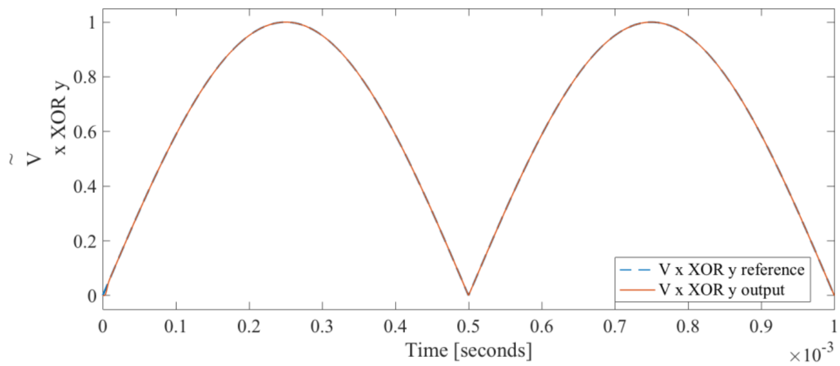

Figure 6: Simulation results of the fuzzy XOR configuration $\tilde{\boldsymbol{V}}_{\boldsymbol{x}\oplus\boldsymbol{y}}(\boldsymbol{k})$ with static parameters and a dynamic gain in each AFSN

3.2.4 Results of the simulation of the Spike trains with the Spike model

Based on the results obtained from the fuzzy XOR configuration $\tilde{V}_{x\oplus y}(k)$ with weights or static parameters and a dynamic gain in each AFSN, the spike trains will be generated agree with the spike model proposed in equations Eqs. 35 and 36.
Spike model with constant I_D diode current,

- The values of the capacitors and the resistors are:
  $R_1 = 100\ K\Omega$
  $R_2 = 100\ K\Omega$
  $R_3 = 110\ \Omega$
  $C_1 = 2.64\ nF$
- The values of the parameters of the diode are:

$V_D = 0.7\ V$
$K = 11600$
$T_K = 298\ K$
$I_D = 10\ mA$

- It is considered an ultra-fast high conductance diode with a reverse recovery time of $4\ ns$, also a high-speed precision operational amplifier with a slew rate of $50\ V/\mu s$.

The simulation results are shown in Figs. 7 and 8 for the input $V_x(k)$ and $V_y(k)$ of Fig. 3. Fig. 9 shows the results for the fuzzy XOR of the input $V_x(k)$ and $V_y(k)$ of Fig. 3, but $V_y(k)$ = $V_x(k)$.

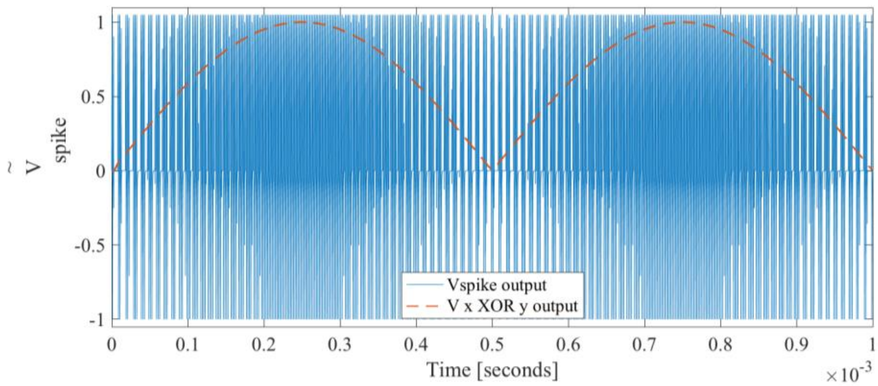

Figure 7: Simulation results of the spike model for the fuzzy XOR configuration $\tilde{V}_{x\oplus y}(k)$ with static parameters and a dynamic gain in each AFSN

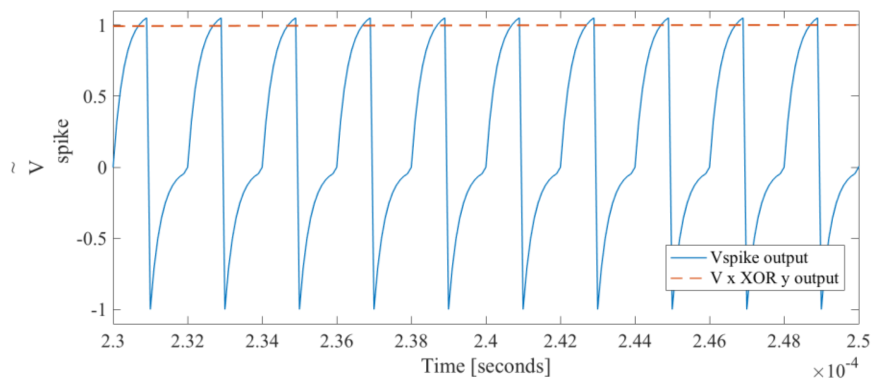

Figure 8: Simulation results of the of the spike model

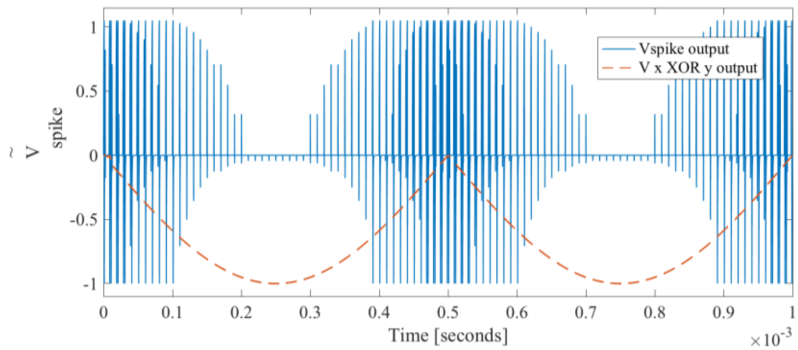

Figure 9: Simulation results of the spike model for the fuzzy XOR configuration $\tilde{\boldsymbol{V}}_{\boldsymbol{x}\oplus\boldsymbol{y}}(\boldsymbol{k})$ with static parameters and a dynamic gain in each AFSN, for the input signals $\boldsymbol{V}_{\boldsymbol{x}}(\boldsymbol{k}) = \boldsymbol{V}_{\boldsymbol{y}}(\boldsymbol{k})$

Spike model with dynamic I_D diode current,

- The values of the capacitors and the resistors are:
  $R_1 = 1\ K\Omega$
  $R_2 = 1\ K\Omega$
  $R_3 = 100\ \Omega$
  $C_1 = 2.2\ nF$
- The values of the parameters of the diode are:
  $V_D = 0.7\ V$
  $K = 11600$
  $T_K = 298\ K$
  $I_s = 0.015\ pA$
- It is considered an ultra-fast high conductance diode with a reverse recovery time of $4\ ns$, also a high-speed precision operational amplifier with a slew rate of $50\ V/\mu s$.

The simulation results are shown in Figs. 10 and 11 for the inputs $V_x(k)$ and $V_y(k)$ of Fig. 3. Fig. 12 shows the results for the fuzzy XOR of the inputs $V_x(k)$ and $V_y(k)$ of Fig. 3, but $V_y(k) = V_x(k)$.

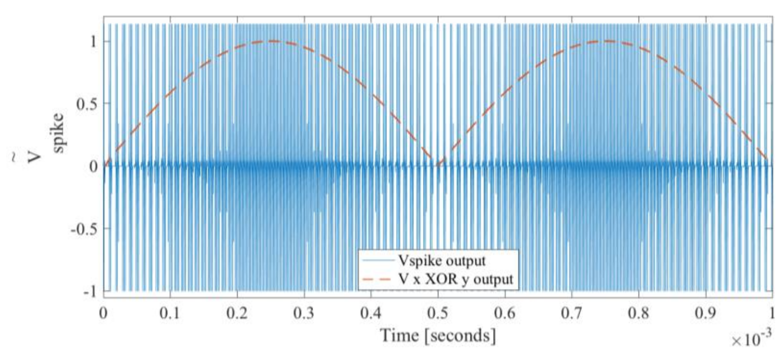

Figure 10: Simulation results of the spike model for the fuzzy XOR configuration $\tilde{V}_{x\oplus y}(k)$ with static parameters and a dynamic gain in each AFSN

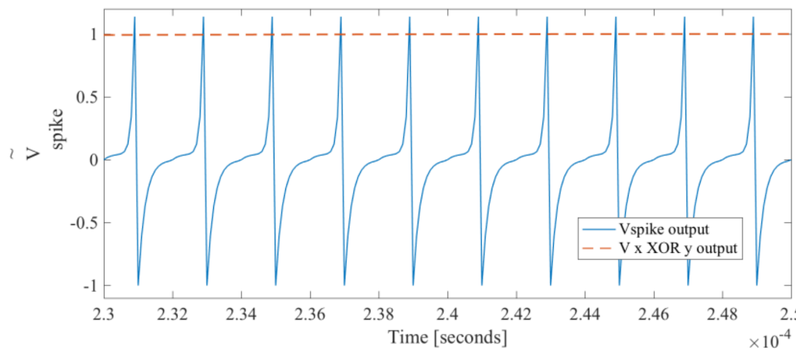

Figure 11: Simulation results of the of the spike model

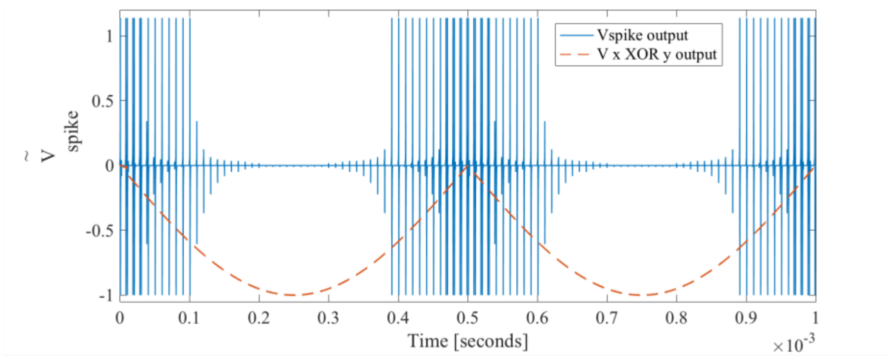

Figure 12: Simulation results of the spike model for the fuzzy XOR configuration $\boldsymbol{\widetilde{V}_{x\oplus y}(k)}$ with static parameters and a dynamic gain in each AFSN, for the input signals $\boldsymbol{V_x(k) = V_y(k)}$

4 Conclusions and comments

A SRM was obtained based on the spike generator circuit [Ramírez-Mendoza, Pérez-Silva and Lara-Rosano (2011)], this model describes the response behavior of the spikes inspired by biological spiking neurons.

One of the great advantages of the spike response model is that the process of information of the time firing pattern or pattern of spike firing with respect to time, is performed based on the trains of pulses obtained from the square pulses of the step activation function, proportional in pulse width to the MAX Operator response of the inputs to the neuron, weighted with the MIN operator. That is to say, the neuronal characteristics such as the refractory time, the axonic delay and the generation of spikes, are performed based on bipolar pulses, so that the information of the pattern of time spike firing is preserved despite the aforementioned neuronal characteristics, also the shape of the response spike model for each pulse. It should also be noted that the frequency of the spike trains are directly proportional to the width of the pulses obtained from the step-type activation function and therefore to the Gupta-type fuzzy integral (MAX operator generalized), of the neuron inputs weighted by means of the synaptic MIN operator.

However, a disadvantage of the AFSNs method could be that many systems or industrial processes have inputs or parameters that do not fit the fuzzy values within the interval [0, 1] for unipolar AFSNs, or [-1, 1] for bipolar AFSNs. The solution to these challenges has been to scale the values of the inputs to carry out the information process in a fuzzy way and then, the output response is rescaled to the interval of values of the system or industrial process [Ramírez-Mendoza, Covarrubias-Fabela, Amezquita-Brooks et al. (2018)].
A potential issue for fuzzy XOR connectives is in quantum computing [Ávila, Schmalfuss, Reiser et al. (2015)], however, systems modeled exclusively with AFSNs should be able to be performed based on basic fuzzy logic operators such as MIN, MAX, INV and the XOR fuzzy operator, similarly as the systems based on classic or non-fuzzy logical operators, INV, AND, OR, XOR; which represents a challenge because many linear and non-linear systems models require other types of mathematical operations such as trigonometric, exponential, derivative and integral.

The SRM and the AFSNs method will allow more precise simulations with applications in several areas such as process control, instrumentation, experimental aerodynamics, navigation and trajectory tracking of UAVs, among others. The proposed example, the fuzzy XOR, demonstrates that the basic fuzzy logic operators can be implemented with the AFSNs, and the results of the simulation show that the dynamic gain or self-tuning gain, of the curve fitting can be used for systems modeling optimization applications and computational modeling [Dutta, Murthy, Kim et al. (2017)].

Acknowledgement: This work was supported by UNAM and CONACYT Research Fellows-UANL, FIME, CIIIA, program. I also appreciate the valuable comments of Eduardo Liceaga-Castro for the writing of this article.

References

Álvarez, U.; Olascoaga, A.; Rivera, D.; Ramírez-Mendoza, A.; García, O. et al. (2017): *Active Disturbance Rejection Control for Micro Air Vehicles Using Frequency Domain Analysis*. Congreso Nacional de Control Automático, Nuevo León, México.

Amezquita-Brooks, L.; Liceaga-Castro, E.; González-Sánchez, M.; García-Salazar, O.; Martínez-Vázquez, D. (2017): Towards a standard design model for quad-rotors: A review of current models, their accuracy and a novel simplified model. *Progress in Aerospace Sciences*, vol. 95, pp. 1-23.

Ávila, A.; Schmalfuss, M.; Reiser, R.; Kreinovich, V. (2015): Fuzzy Xor classes from quantum computing. *14th International Conference on Artificial Intelligence and Soft Computing, Part II, LNAI 9120*, pp. 305-317.

Bedregal, B. C.; Reiser, R. H. S.; Dimuro, G. P. (2009): Xor-implications and E-implications: Classes of fuzzy implications based on fuzzy Xor. *Electronic Notes in Theoretical Computer Science*, vol. 247, pp. 5-18.

Chaturvedi, S. (2011): Review of spiking neural network architecture for feature extraction and dimensionality reduction. *Fourth International Conference on Emerging Trends in Engineering & Technolog*, pp. 317-322.

Desoer, C. A.; Kuh, E. S. (1969): *Basic Circuit Theory*. Tokyo, Japan, McGraw-Hill Kogausha, Ltd., pp. 544-549.

Dutta, S.; Murthy, A. R.; Kim, D.; Samui, P. (2017): Prediction of compressive strength of self-compacting concrete using intelligent computational modeling. *Computers, Materials & Continua*, vol. 53, no. 2, pp. 157-174.

Esfandiari, R. S.; Lu, B. (2014): *Modeling and Analysis of Dynamic Systems*. CRC Press Taylor & Francis Group, pp. 239-265.

Espinosa-Ramos, J. I.; Cruz-Cortes, N.; Vazquez, R. A. (2013): Creation of spiking neuron models applied in pattern recognition problems. *International Joint Conference on Neural Networks*, pp. 1-8.

Gerstner, W.; Kistler W. M. (2002): *Spiking Neuron Models*. Cambridge University Press, pp. 480.

Ghosh-Dastidar, S.; Adeli, H. (2009): Spiking neural networks. *International Journal of Neural Systems*, vol. 19, no. 4, pp. 295-308.

Gupta, M. M. (1993): Fuzzy logic, neural networks and virtual cognitive systems. *Second International Symposium on Uncertainty Modeling and Analysis*, pp. 90-97.

Gupta, M. M.; Qi, J. (1992): On fuzzy neuron models. In Lotfi A. Zadeh et al (Eds.), *Fuzzy Logic for the Management of Uncertainty*, Wiley-Interscience, pp. 479-491.

Gupta, M. M.; Rao, D. H. (1994): On the principles of fuzzy neural networks. *Fuzzy Sets and Systems*, vol. 61, pp. 1-18.

Izhikevich, E. M. (2003): Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569-1572.

Jiang, L.; Guo, Z.; Zhang, B. (2015): Scalar calibration of aeromagnetic data using BPANN and LS algorithms based on fixed-wing UAV platform. *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 7, pp. 1968-1976.

Lara-Rosano, F. (2017): A Heuristic framework for non-conscious reasoning. *Applied Sciences, Concept Paper*, vol. 7, no. 1161.

Lashkare, S.; Chouhan, S.; Chavan, T.; Bhat, A.; Kumbhare, P. et al. (2018): PCMO RRAM for integrate-and-fire neuron in spiking neural networks. *Electron Device Letters*, vol. 39, no. 4, pp. 484-487.

Li, X.; Yu, W.; Lara-Rosano, F. (2000): Dynamic knowledge inference and learning under adaptive fuzzy petri net framework. *IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews*, vol. 30, no. 4, pp. 442-450.

McCulloch, W. S.; Pitts, W. (1943): A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133.

Pérez, S. J. L.; Garcés, M. A.; Cabiedes, C. F.; Miranda, V. A. (2009): Electronic model of a Dubois fuzzy integration neuron. *Journal of Applied Research and Technology*, vol. 7, no. 1, pp. 73-82.

Pérez, J. L.; Ramírez, A. (2001): Two new models of integrative fuzzy neuron. *Instrumentation & Development*, vol. 5, no. 3, pp. 140-145.

Ramírez-Mendoza, A. (2014): Study of the response of the connection of Adaptive Fuzzy Spiking Neurons with self-synapse in each single neuron. *11th International Conference on Electrical Engineering, Computing Science and Automatic Control*, pp. 1-6.

Ramírez-Mendoza, A. (unpublished): Design of the law of control of a PID controller based on Adaptive Fuzzy Spiking Neurons.

Ramírez-Mendoza, A. (unpublished): Fault-Tolerant PID Control Law Design based on Adaptive Fuzzy Spiking Neurons.

Ramírez-Mendoza, A.; Covarrubias-Fabela, J.; Amezquita-Brooks, L.; Hernández-Alcantara, D. (2018): Parameter identification using fuzzy neurons: Application to drones and induction motors. *DYNA*, vol. 93, no. 1, pp. 75-81.

Ramírez, A.; Pérez, J. L. (2002): A Fuzzy gupta integrator neuron model with spikes response and axonal delay. *Advances in Artificial Intelligence & Engineering Cybernetics*, pp. 12-16.

Ramírez-Mendoza, A.; Pérez-Silva, J. L.; Lara-Rosano, F. (2011): Electronic implementation of a fuzzy neuron model with a gupta integrator. *Journal of Applied Research and Technology*, vol. 9, no. 3, pp. 380-393.

Rosado-Muñoz, A.; Bataller-Mompeán, M.; Guerrero-Martínez, J. (2012): FPGA implementation of spiking neural networks. *Proceedings of the 1st IFAC Conference on Embedded Systems, Computational Intelligence and Telematics in Control-CESCIT*, pp. 139-144.

Rossant, C.; Goodman, D. F. M.; Platkiewicz, J.; Brette, R. (2010): Automatic fitting of spiking neuron models to electrophysiological recordings. *Frontiers in Neuroinformatics*, vol. 4, no. 2, pp. 1-10.

Tawfik, M. E.; Bishay, P. L.; Sadek, E. A. (2018): Neural network-based second order reliability method (NNBSORM) for laminated composite plates in free vibration. *Computer Modeling in Engineering & Sciences*, vol. 115, no. 1, pp. 105-129.

Widrow, B.; Lehr, M. A. (1990): 30 years of adaptive neural networks: Perceptron, madaline, and backpropagation. *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1415-1442.

Yildizel, S. A.; Öztürk, A. U. (2016): A Study on the estimation of prefabricated glass fiber reinforced concrete panel strength values with an artificial neural network model. *Computers, Materials & Continua*, vol. 52, no. 1, pp. 41-52.

Yu, W.; Rosen, J. (2013): Neural PID control of robot manipulators with application to an upper limb exoskeleton. *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 673-684.

Zadeh, L. A. (1977): *Theory of Fuzzy Sets, Encyclopedia of Computer Science and Technology*. Marcel Dekker, Nueva York, E.U.A.

Zheng, N.; Mazumder, P. (2018): Learning in memristor crossbar-based spiking neural networks through modulation of weight-dependet spike-timing-dependent plasticity. *Transactions on Nanotechnology*, vol. 17, no. 3, pp. 520-532.