

A NEW DESIGN OF TANGENT HYPERBOLIC FUNCTION GENERATOR WITH APPLICATION TO THE NEURAL NETWORK IMPLEMENTATIONS

Hacer ATAR YILDIZ¹

A CMOS hyperbolic tangent function generator circuit suitable for the implementation of analog neural networks is presented. In order to obtain an accurate yet simple circuit realization, a judiciously chosen symmetrical Padé approximation of the hyperbolic tangent function is proposed. As an illustrative application, we set up an application in which the proposed circuit is used as the nonlinear block of a two-layer neural network. Simulation results using Spectre Simulation tool in Cadence design environment with 0.18 μ m CMOS process verify proper operation of the proposed circuit as well as the neural network built around. These results demonstrate the validity of the theoretical analysis and the feasibility of the proposed circuit.

Key words: Tangent hyperbolic function generator, Padé approximation, Activation function, Neural network implementation

1. Introduction

This An accurate and high-speed nonlinear function generation is highly demanded in many areas of electronics that require computation, instrumentation and control system design [1-3]. Conventionally function generation is realized by using look-up table (LUT) [4], polynomial expansion and approximation methods. [5-6]. One of the major disadvantage of the LUT method is that it requires many hardware resources, thus this is solely used for digital implementations. On the other hand, methods relying on approximations such as Taylor and Padé approximants are mainly intended to the analog circuit realizations, which can be classified into two main groups; circuits realized with transistors operating in sub-threshold mode, usually destined for low frequency applications [7], and circuits using transistors operating in strong inversion region [8]. Besides the well-known limitations of the circuits designed with transistors operating in sub-threshold, the most important limitation of all these solutions is the limited accuracy. In order to increase the accuracy, the approximation order should be increased at the cost of increasing circuit complexity. Obviously, this fact leads to the increased chip area, which makes the circuits unattractive for VLSI design. One of the commonly used nonlinear function is the hyperbolic tangent function, which can be expressed depending on exponential function. In this paper, it is aimed to design hyperbolic tangent function generator with high accuracy and minimal circuit complexity using an appropriate approach.

¹Faculty of Electrical and Electronics Engineering, Istanbul Technical University, 34469 Maslak, Istanbul, Turkey, (haceryildiz@itu.edu.tr)  <https://orcid.org/0000-0003-4490-6878>

For this purpose, circuit realization is provided by using Padé approximation considering the following design criteria:

i) Since the hyperbolic tangent function generator is realized in strong inversion region, it will operate in a much wider frequency range than the sub-threshold circuits proposed in the literature [9]. In neural network applications, signals are usually applied to neuron in the form of pulses, or spikes, which requires wide bandwidth operation of the neurons. These requirements may be difficult to achieve with sub-threshold circuits.

ii) The degree of approximation should be very high to increase accuracy in analog circuits. This causes the design complexity for the implementation of circuit. For this reason, it is necessary to recommend a special approximation for hyperbolic tangent function, rather than a general approximation (Taylor, Padé etc.) for the desired function.

One of the application area of the hyperbolic tangent function is the analog neural network (ANN) implementations. In ANN circuits, one of the most common function used to perform the activation process is a S-shaped function and produces outputs in the scale of -1 to 1, which is perfectly realized with the hyperbolic tangent function. There are many circuit examples designed to perform the hyperbolic tangent function [10-11]. In this work, an ANN that performs character recognition is also used to demonstrate the usefulness of the proposed circuit. For this purpose, the proposed tangent hyperbolic circuit is used to realize the activation process. All simulation results are obtained in UMC 0.18 μ m CMOS technology. The paper is planned as follows: The realization of the hyperbolic tangent function and its circuit implementation is given in Section 2 and Section 3 respectively. Simulation results of the proposed circuits and comparison with the previous works are presented in Section 4. The application of the proposed hyperbolic tangent function generator is described in Section 5. Finally, conclusion part is presented in Section 6.

2. Realization of the Hyperbolic Tangent Function

2.1. The Design of Hyperbolic Tangent Function Generator

As it is well known from the literature, hyperbolic tangent function can be expressed in terms of the exponential function as follows:

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (1)$$

Besides this, exponential function has very effective and well-known approximations [12-13]. Among these, Taylor and Padé approximations constitute two mainstreams in the design of exponential function generators, as the former is a polynomial type approximation,

$$e^{ax} = 1 + \frac{ax}{1!} + \frac{m, n a^2 x^2}{2!} + \dots + \frac{a^n x^n}{n!} + \dots \quad (2)$$

while the latter is a rational type approximations of the following form,

$$e^x = R_{m,n}(x) = \frac{n_m(x)}{d_n(x)} \quad (3)$$

where, m and n are used to define the nominator and denominator degrees of the Padé approximant, respectively [14]. It should be noted that the choice of the polynomial or rational type

approximation affects the performance of the resulting function generator drastically. In general, the use of Padé approximant leads to more complicated but more accurate systems, as the rational function approximates the fast variation of the exponential function for finite x values, thanks to the existing denominator term. The design of hyperbolic tangent function generator circuit is considered based on the following two criteria:

- i) The circuit complexity of the resulting function generator will depend on the mathematical complexity of the approximating function, that is the orders of the numerator and denominator polynomials in Eqn. (3).
- ii) In addition to the simplicity of the circuit design, the accuracy of the function generator's response over wide input and output range is also very important.

In this paper, the proposed realization of the hyperbolic tangent function is based on the expression in Eqn. (1) and exponential Padé approximants defined in Eqn. (3). By substituting the approximate expression in (3) into (1) and rearranging of the terms, the following simple approximation of the hyperbolic tangent function is obtained:

$$\tanh(x) \approx \frac{(R_{m,n} \times R_{n,m}) - 1}{(R_{m,n} \times R_{n,m}) + 1} \quad (4)$$

Note that the proposed approximation is obtained by considering the following realization of the involved exponential terms in Eqn. (1) as follows:

$$e^{2x} = e^x e^x \approx R_{m,n}(x) R_{n,m}(x) \quad (5)$$

where the first and the second product terms are symmetrical exponential Padé approximants of order $[m/n]$ and $[n/m]$, respectively. It should be noted that owing to the use of the symmetrical Padé approximants, the individual errors of each approximant cancel out each other and leads to more accurate approximation. On the other hand, this choice leads to mathematically simpler expression, which in turn leads to simpler circuit realization. These advantages will be discussed further in Section 2.2.

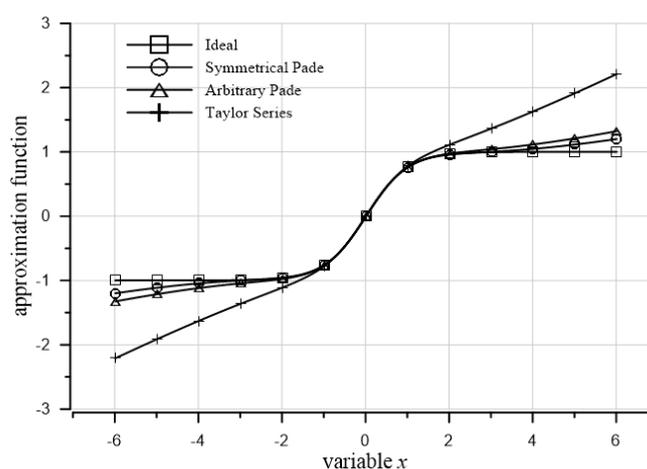


Fig.1 Comparison of Padé approximation with symmetrical pairs, arbitrary Padé approximation and Taylor series approximation

The Matlab simulation results obtained using Taylor and Padé approximants are shown in Fig.1. In order to obtain a meaningful comparison, the degrees of nominator and denominator polynomials of the different approximations are taken as equal. As it can be easily seen from these results, the symmetrical Padé approximation gives a much more accurate result than Taylor approximation in a wide range.

2.2. Advantages of the Proposed Approach

Due to the nature characteristics of the hyperbolic tangent function, symmetrical rational Padé pairs are selected for approximating to the hyperbolic tangent function, and obtained both a much more accurate result and a simpler circuit implementation. Symmetrical Padé approximants can be expressed as follows:

$$R_{m,n}(x) = \frac{n(x)}{d(x)} \quad \text{and} \quad R_{n,m}(x) = \frac{d(-x)}{n(-x)} \quad (6)$$

as a result of which, hyperbolic tangent function assumes the following form:

$$\tanh(x) \approx \frac{A(x) - A(-x)}{A(x) + A(-x)} \quad (7)$$

where $A(x) = n(x)d(-x)$.

Assuming that even and odd parts of the polynomial $A(x)$ are respectively $E(x)$ and $O(x)$, that is $A(x) = E(x) + O(x)$ and $A(-x) = E(x) - O(x)$, the approximation of the hyperbolic tangent function is obtained as follows:

$$\tanh(x) \approx \frac{O(x)}{E(x)} \quad (8)$$

Therefore, all even order terms in the numerator and all odd order terms in the denominator are missing. Therefore, the use of symmetrical Padé pair may lead to the simplest algebraic approximation compared to approximations composed using all other Padé combinations.

On the other hand, in order to demonstrate the usefulness of the proposed method, the error function of the proposed approximation can be obtained as follows:

$$\varepsilon_1 \triangleq e^x - R_{m,n} = e^x - \frac{n(x)}{d(x)} \quad (9)$$

$$\varepsilon_2 \triangleq e^x - R_{n,m} = e^x - \frac{d(-x)}{n(-x)} \quad (10)$$

where ε_1 and ε_2 are the error functions of the Padé approximants.

Similarly, the error function of the hyperbolic tangent function can be calculated as follows:

$$\varepsilon_{\tanh} \triangleq \tanh(x) - \frac{(R_{n,m}R_{m,n}) - 1}{(R_{n,m}R_{m,n}) + 1} \quad (11)$$

$$\varepsilon_{\tanh} \cong \frac{e^x}{(1 + e^{2x})^2} (\varepsilon_1 + \varepsilon_2) \quad (12)$$

It should be also noted that, the error functions ε_1 and ε_2 are symmetrical for $x = 0$. Therefore, ε_1 and ε_2 cancel each other for small x and ε_{\tanh} remains small. Although for large values of x , the error functions lose their symmetric properties, the $\varepsilon_1 + \varepsilon_2$ still remains smaller than either ε_1 or ε_2 . The corresponding errors are given in Fig.2.

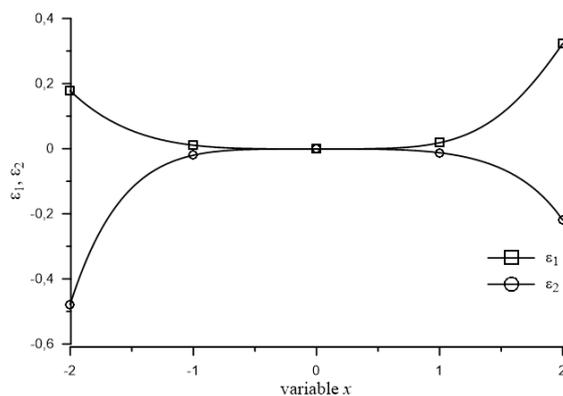


Fig.2 Error functions, ε_1 , ε_2 (see Eqns. 9 and 10) for $m=2$, $n=1$.

Therefore, although the error is small in the case of symmetrical Padé approximants, still the values of the parameters m and n is important and affects the approximation error.

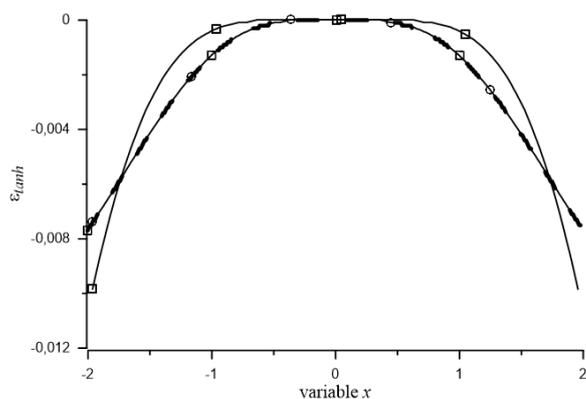


Fig. 3 Comparison of the error functions, ε_{tanh} (see Eqn. 11) (straight line: $m=3$, $n=1$, dashed line: $m=2$, $n=1$).

Obviously, there is a trade-off between the complexity of the approximating function and the circuit complexity. The value $n+m$ can be considered to estimate the complexity of the Padé approximation. For more complicated function, i.e for larger values of $m+n$, the approximation error is smaller, but the resulting circuit is more complicated.

However, as m and n are increased, the rate of error reduction may not be as high as expected. To be specific, let us consider two different cases, in the first one of which is used Padé approximants with $m=2$, $n=1$ and in the second one of which with $m=3$, $n=1$. Therefore, it is compared the errors of the different symmetrical Padé approximants in Fig.3. For a limited input range, the symmetrical Padé pair for $m=2$, $n=1$ approximates hyperbolic tangent function with smaller error compared to the more complicated approximation, $m=3$, $n=1$. This provides an important advantage as the resulting circuit would be considerably simpler. Obviously, for larger input range, more complicated approximation yields better results, but having more complicated circuit realization. Nevertheless, the degrees of the symmetrical Padé approximants in Eqn. (4) are set to $m=2$ and $n=1$. With these parameters, the proposed approximation for the hyperbolic tangent function is obtained as follows:

$$\tanh(x) \approx \frac{x^3 + 15x}{6x^2 + 15} \quad (13)$$

3. Implementation of the Hyperbolic Tangent Function Generator

Many squarer circuits proposed in the literature realizes the following function [15-18]:

$$I_Y = \frac{\alpha I_X^2}{I_B} + \beta I_B \tag{14}$$

where I_B is a scaling biasing current. The architecture of the proposed hyperbolic tangent function generator is depicted in Figure 4. The structure consists of two current mode squarers, each realizing the function in Eqn. (14) and a current mode analog multiplier, which realizes the following function:

$$I_{out} = \frac{I_a I_b}{I_c} \tag{15}$$

Assuming that the parameters of the current mode squarers are set to $\alpha_1=\alpha_2=1$ and $\beta_1=15, \beta_2 =15/6$ and the currents driving the subcircuits are chosen as:

$$I_a = \frac{I_{in}^2}{I_B} + 15I_B, \quad I_b = \frac{I_{in}}{6}, \quad I_c = \frac{I_{in}^2}{I_B} + \frac{15}{6} I_B$$

the output current is found as:

$$I_{out} = \frac{I_{in}(I_{in}^2 + 15I_B^2)}{6(I_{in}^2 + \frac{15}{6} I_B^2)} = \frac{I_{in}^3 + 15I_{in}I_B^2}{6I_{in}^2 + 15I_B^2} \tag{16}$$

On the other hand, approximation of the hyperbolic tangent function in Eqn. (4) is obtained for $m=2, n=1$ as follows:

$$I_{out} = \frac{I_{in}(I_{in}^2 + 15I_B^2)}{6(I_{in}^2 + \frac{15}{6} I_B^2)} = \frac{I_{in}^3 + 15I_{in}I_B^2}{6I_{in}^2 + 15I_B^2} \tag{17}$$

Therefore, by comparing Eqns. (16) and (17), it is seen that the circuit is indeed realizes approximate hyperbolic tangent function for $x=I_{in}/I_B$ and $y=I_{out}/I_B$. As it can be seen from Fig.4, the circuit adopts three input currents. These currents are realized using a current steering/replicator block implemented using simple current mirrors. Concurrently, the scaling coefficient of 1/6 can be realized by adjusting the transistor sizes of the involved current mirrors, without having to use any additional circuit.

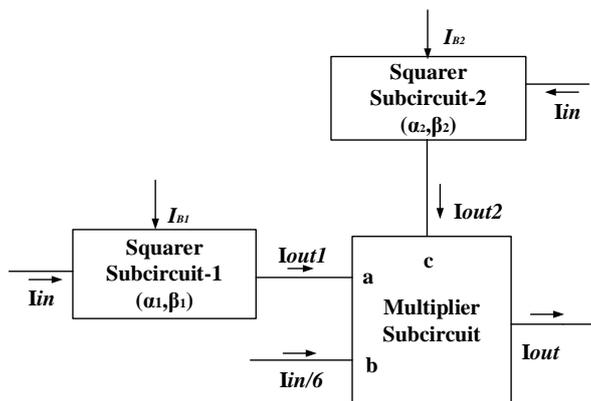


Fig.4 Architecture which realizes hyperbolic tangent function generator for $m=2, n=1$ ($\beta_1=15$ and $\beta_2=15/6$).

The hyperbolic tangent function generator has been redesigned using an alternative symmetrical Padé pair with the following parameters: $m=3$ and $n=1$ for comparative purposes. The hyperbolic tangent function obtained using the approach employed in Section 2. is given as follows:

$$y = \tanh(x) \approx \frac{7x^3 + 42x}{x^4 + 24x^2 + 24} \tag{18}$$

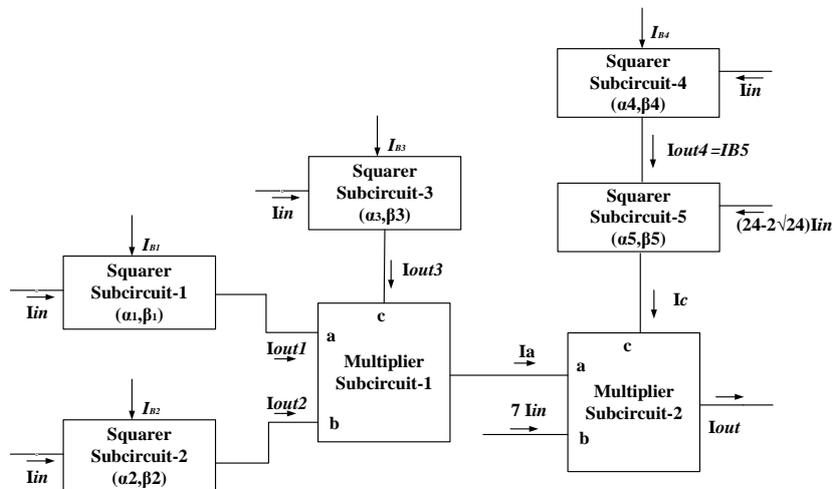


Fig.5 Architecture which realizes hyperbolic tangent function generator with the symmetrical Padé pair for $m=3, n=1$

For $x=I_{in}/I_B$ and $y=I_{out}/I_B$, the hyperbolic tangent function can be obtained in terms of currents as follows:

$$I_{out} = \frac{7I_B^2(I_{in}^3 + 6I_{in}I_B^2)}{I_{in}^4 + 24I_{in}^2I_B^2 + 24I_B^4} \tag{19}$$

If the hyperbolic tangent function is designed using squarer and multiplier subcircuits, the architecture of the circuit is obtained as in Fig.5. As can be seen from this figure, this circuit consists of 5 squarer and 2 multiplier subcircuits and therefore this realization in this way requires more complicated circuit.

3.1. Current-Mode Squarer Subcircuit

As it can be seen from the Fig.6, it is considered the squarer subcircuit-1 which is well-known circuit from the literature and operates in current mode [17]. This circuit is based on trans-linear loop (M1-M4), and all transistors used in the circuits are assumed to operate in saturation mode. According to this, the current equation for saturation region can be given as follows:

$$I_{DS} = \frac{1}{2} \mu_n C_{ox} \frac{W}{L} (V_{GS} - V_t)^2 \tag{20}$$

where the transistor parameters have the usual meanings.

By applying Kirchoff's voltage law around the loop consisting of transistors M1-M4, using the square relationship given in Eqn. (20) and assuming that transistors are perfectly matched, we obtain the following relation for the transistors' drain currents:

$$V_{GS1} + V_{GS2} = V_{GS3} + V_{GS4} \tag{21}$$

$$\sqrt{I_{DS1}} + \sqrt{I_{DS2}} = \sqrt{I_{DS3}} + \sqrt{I_{DS4}} \tag{22}$$

It is assumed that, the transistors have the same transconductance value. By substituting (20) into (21) and considering $I_{DS1}=I_{DS2}=I_{B1}$, the output current is obtained as follows:

$$I_{out1} = \alpha_1 \frac{I_{in}^2}{I_{B1}} + \beta_1 I_{B1} \tag{23}$$

where $\alpha_1=1$ and $\beta_1=15$.

Note that since the I_B is the biasing current, the value of $\beta_1 I_B$ is provided easily by scaling the corresponding current. The output of the second squarer circuit can be also obtained as follows:

$$I_{out2} = \alpha_2 \frac{I_{in}^2}{I_{B2}} + \beta_2 I_{B2} \tag{24}$$

where $\alpha_2=1$ and $\beta_2 = 15/6$, this value can be obtained by appropriate scaling of the biasing current.

3.2. Multiplier Subcircuit

This subcircuit which performs the multiplication operation, can easily be realized with the following equation [19]:

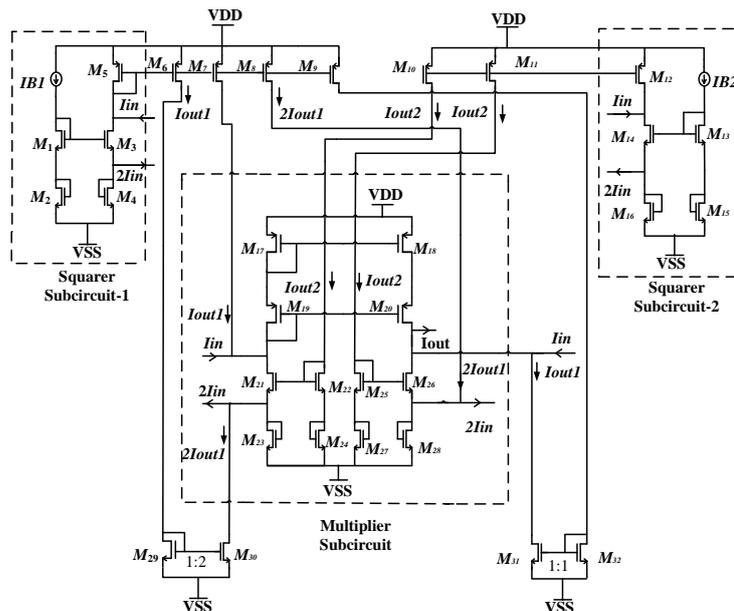


Fig. 6. The realization of proposed hyperbolic tangent function generator

$$(X + Y)^2 - (X - Y)^2 = 4XY \tag{25}$$

As can be easily seen also from the Fig.6, this circuit is based on two trans-linear loops and achieves squaring operations by using these loops. The first loop formed by the M_{21} - M_{24} transistors provides the squarer function $(X+Y)^2$ and the second loop formed by the M_{25} - M_{28} transistors provides the squarer function $(X-Y)^2$. The equations performed by the first and second loops can be given as follows, depending on the current values:

$$I_{out} = \frac{I_{in} I_{out1}}{I_{out2}} \tag{26}$$

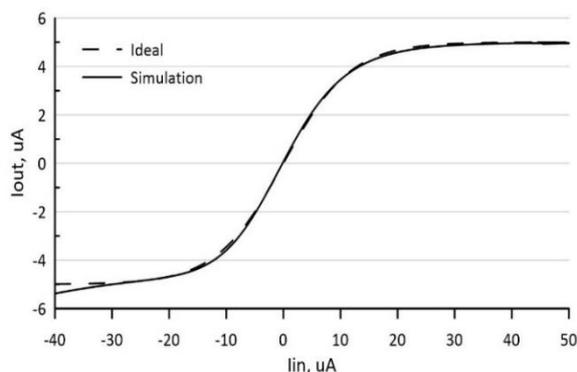
By substituting the values I_{in} , I_{out1} and I_{out2} into (26) and setting $I_B=I_{B1}=I_{B2}$, the final equation is obtained as below:

$$I_{out} \approx \frac{I_{in} (I_{in}^2 + 15I_B^2)}{6I_B (I_{in}^2 + \frac{15}{6}I_B^2)} \tag{27}$$

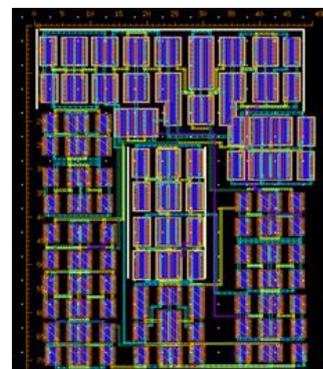
It can be said that the proposed circuit easily performs the hyperbolic tangent function characteristic by appropriately selecting the current values.

4. The Comparison and Simulation Results

In order to demonstrate the usefulness of the proposed circuit, the hyperbolic tangent function generator is designed for implementing in 0.18 μm UMC technology, under supply voltage of $\pm 1\text{V}$. For the proper operation, the biasing current is chosen for $I_B=5\mu\text{A}$. In order to evaluate simulation results, it is compared the nonlinear transfer characteristic of the circuit with the ideal hyperbolic tangent function, defined as $I_{\text{out}}=\alpha \cdot \tanh(\beta I_{\text{in}})$.



(a)



(b)

Fig. 7. a) Simulated circuit output characteristic and ideal characteristic for hyperbolic tangent function. b) The proposed hyperbolic tangent function circuit with 0.18 μm CMOS technology layout.

As it can be seen from Fig.7a, the ideal characteristic obtained for $\alpha=1.088$ and $\beta=7800$ is quite compatible with the simulation results. The average error is found to be as 3%. According to these results, it can easily be said that the proposed hyperbolic tangent function generator gives a much more accurate result compared to the other analog circuit generator in reference [9,22], at the cost of having more complicated circuitry. Besides this, in order to provide a fair comparison with other circuits in the literature in terms of the area, accuracy and power consumption, it is obtained the circuit layout as shown in Fig.7b. Crucial performance figures of the proposed circuit and the recently presented works are tabulated in Table 1.

Table.1 Comparison table with the previous works

| Structure | Hybrid[21] | Hyperbolic tangent [9] | Hyperbolic tangent[22] | proposed |
|-------------------------|-------------|------------------------|------------------------|-------------------|
| Area(μm^2) | 3646 | 39 | 90 | 3500 |
| Maximum error | 3.6% | 19.7% | 29.33% | 3% |
| Power consumption | NA | 62.5 μW | 137 μW | 244 μW |
| Technology | 180-nm CMOS | 180-nm CMOS | 180-nm CMOS | 180-nm CMOS |

As it is known from the literature, in such circuits, digital only solutions (LUT) suffers from large chip area and from large power consumption [20]. As a remedy, hybrid realization has been proposed, which achieves better accuracy while providing acceptable chip area and power consumption [21].

Accordingly, as shown in Table 1, the proposed circuit offers a more advantageous solution in terms of the area and the accuracy compared to the hybrid circuit. Beside this, it is clear that the proposed circuit provides much better error performance compared to the other analog circuits. In addition, it is also found the power consumption of the proposed circuit as 244 μW .

5. Application

In today state-of-the art technologies, the realization of artificial neural networks (ANN) featuring large number of neurons and huge number of interconnection between them poses significant challenges to the integrated circuit designers.

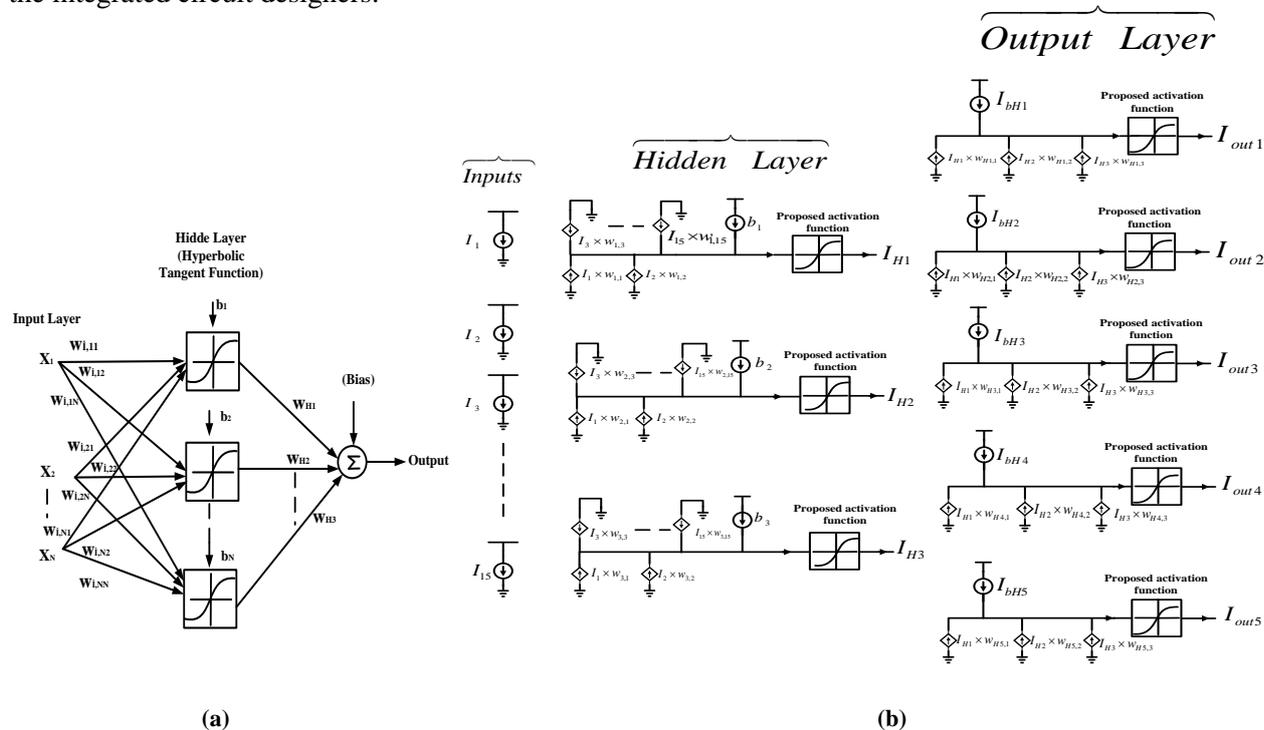


Fig.8.a) General structure of analog neural network. b) The analog neural network circuit performing character recognition

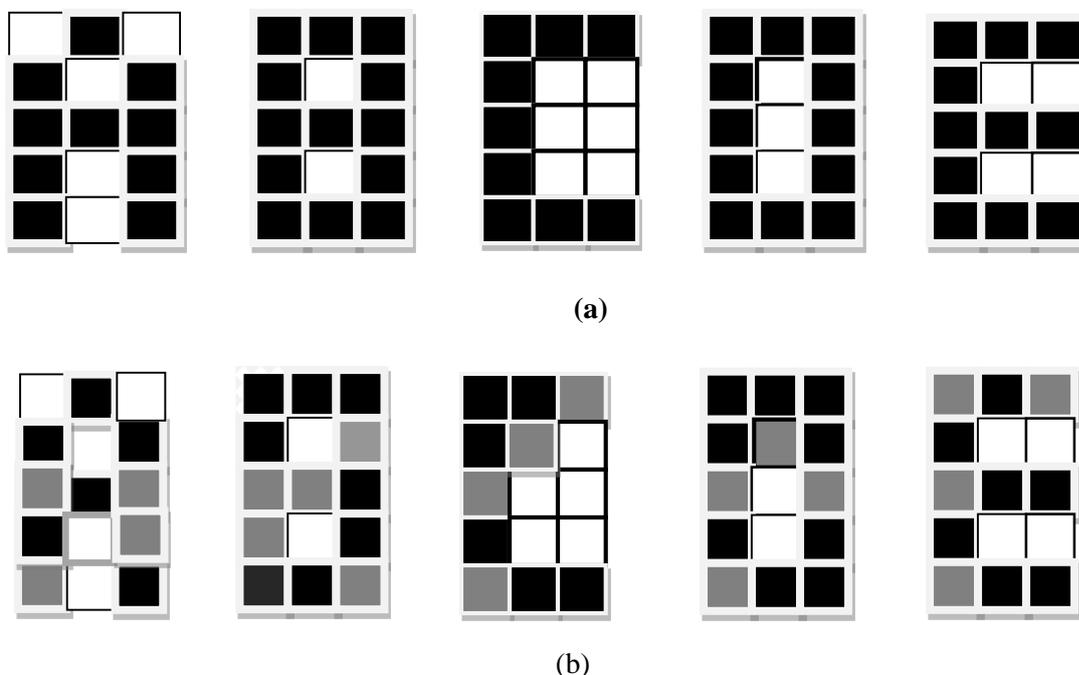


Fig.9. a) Representation of the regular characters with a 5x3 matrix. b) Examples of noisy inputs where the network generates correct outputs (black color 10uA, white color 0uA, gray levels represent the level between 0u -10u of output current)

Therefore, many studies on the application and implementation of the ANNs are presented in the literature [22-23]. The general scheme of ANN is given in Fig.8a. In order to illustrate the usefulness of this work, the proposed hyperbolic tangent function generator is used as an activation function in ANN circuit in Fig.8a. The required weighted summing operation are realized using current mirrors in which the weights are set by adjusting transistor aspect ratios. The resulting circuit is shown in Fig.8b. This circuit with one hidden layer operates in current mode and is used for the character recognition from the inputs set up to represent the letters of the alphabet from the letter "A" to the letter "E". These characters are expressed in a 5X3 matrix and represented by black and white patterns. The representation of these letters are shown in Fig.9. The training of the network and derivation of the proper weights are derived using MATLAB. These values are inserted to the proposed CMOS circuit using Cadence simulation program. Thus, by applying the input set that provides each character, the characters corresponding to the output values are obtained. It has also been tested whether the ANN is capable of generalization and fault tolerant. For this purpose, some of the incorrect input values are deliberately applied to the network and it is determined that the network generates correct results despite this error. Therefore, it can be said that for this ANN, where the proposed CMOS circuit performs the activation function, it is a fault tolerant circuit.

6. Conclusion

In this study, a new circuit that provides tangent hyperbolic function characteristics is proposed using Padé approximation. With the proposed approach, it is aimed to realize the tangent hyperbolic function algebraically simple. Thus, the circuit implementation of the tanh function is also reduced to a much simple form. In order to demonstrate the usefulness of the proposed circuit, this circuit is used as an activation function in an ANN with 15 inputs, 5 outputs and performing character recognition. The proposed circuit is based on the translinear principle and consists of sub circuits such as squarer and multiplier. By comparing the proposed circuit with those given in the literature, it can easily be said that the circuit still covers a small area while providing an accuracy of 3%. The proposed circuit has been applied to a simple character recognition problem, which is a well-known problem in the literature, and its usefulness and applicability have been proven.

References

- [1] Weng, J.H., Lin, D.Y. (2014). Sine function generator. *Int. Sym. on Bio. and Bioinf. (ISBB)*, 11-14.
- [2] Enqvist, M., Schoukens, J., Pintelon, R. (2007). Detection of unmodeled nonlinearities using correlation methods. *IEEE Instrumentation and Measurement Tech. Conference (IMTC)*, Poland.
- [3] Esfahani, A.F., Dreesen, P., Tiels, K., Noël, J., Schoukens, J. (2018). Parameter reduction in nonlinear state-space identification of hysteresis. *Mechanical Sys. and Signal Proces.*, 104, 884-895.
- [4] Wang, M., Liu, B., Foroosh, H. (2018). Look-up table unit activationfunction for deep convolutional neural networks. *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 1225–1233.
- [5] Gomar, S., Mirhassani, M., Ahmadi, M. (2016). Precise digital implementations of hyperbolic tanh and sigmoid function. *50th Asilomar Conference on Signals, Systems and Computers*, 1586–1589.
- [6] Kutluoglu, D., Toker, A., Ozoguz, S. (2017). CMOS Current Mode Exponential Function Generator Circuit Using Padé Approximation. *10th Int. Conf. on Elec.l and Elec. Eng. (ELECO)*, 1202-1206.

- [7] Robles, M.C., Serrano, L. (2009). Accurate differential tanh(nx) implementation. *Int. J. Circ. Theor. Appl.* 37, 613-629.
- [8] Abuelma'atti, M.T., Al-Yahia N.M. (2008). An Improved Universal CMOS Current-Mode Analog Function Synthesizer: Performance Analysis. *International Journ. of Electronics*, 95(1), 1127-1148.
- [9] Shamsi, J., Amirsoleimani, A., Mirzakuchaki, S., Ahmadi, A., Alirezaee, S., Ahmadi, M. (2015) Hyperbolic Tangent Passive Resistive-Type Neuron. *IEEE Int. Symp. on Cir. and Sys. (ISCAS)*.
- [10] Namin, A.H., Leboeuf, K., Muscedere, R., Wu, H., Ahmad, M. (2009). Efficient hardware implementation of the hyperbolic tangent sigmoid function. *Int. Symp. on Cir. and Sys. (ISCAS)*.
- [11] Erdener, Ö., Ozoguz, S. (2016). A new neuron and synapse model suitable for low power VLSI implementation. *Analog Integr. Circ. Sig. Process*, 89, 749–770.
- [12] Lin K.J., Liu M.K. (2007). CMOS current-mode exponential circuit using Padé approximation. *Chung Hua Journal of Science and Engineering*, 5, 77-80.
- [13] McCabe, J. H. (2009). On the Padé table for $e(x)$ and the simple continued fractions for e and $e(L/M)$. *The Ramanujan Journal*, 19, 95-105.
- [14] Baker Jr., G.A., Graves-Morris, P.R., *Padé Approximants*, Cambridge Uni. Press, Cambridge, 1996.
- [15] Saatlo, A.N., Ozoguz, S., Minaei, S. (2015). Applications of a CMOS Current Squaring Circuit in Analog Signal Processing. *38th Int. Conference on Telecom. and Signal Processing (TSP)*, 339-343.
- [16] Alikhani, A., Ahmadi, A. (2012). A novel current-mode four-quadrant CMOS analog multiplier/divider. *Inter. Journal of Electronics and Communications (AEÜ)*, 66 (7), 581-586.
- [17] Naderi, A., Khoei, A., Hadidi, K., Ghasemzadeh, H. (2009). A New High Speed and Low Power Four-Quadrant CMOS Analog Multiplier in Current-Mode. *Int. Journal of Elec. and Com. (AEÜ)*, 63(9), 769-775.
- [18] Salama, M.K., Soliman, A.M. (2003). Low-Voltage Low-Power CMOS RF Four-Quadrant Multiplier. *International Journal of Electronics and Communications (AEÜ)*, 57(1), 74-78.
- [19] Han, G., Sanchez-Sinencio, E. (1998). CMOS transconductance multiplier a tutorial, *IEEE Trans. Circuits Syst. II* 45 (12), 1550–1563.
- [20] Leboeuf, K., Namin, A., Muscedere, R., Wu, H., Ahmadi, M. (2008). High speed VLSI implementation of the hyperbolic tangent sigmoid function. *Int. Conf. Conver. Hybrid Inf. Technol.*, 1070–1073.
- [21] Namin, A., Leboeuf, K., Muscedere, R., Wu, H., Ahmadi, M. (2009). Efficient hardware implementation of the hyperbolic tangent sig. function. *IEEE Int. Symp. Circuits Syst.*, 2117–2120.
- [22] Heidari, M., Shamsi, H. (2019). Analog programmable neuron and case study on VLSI implementation of Multi-Layer Perceptron (MLP). *Microelectronics Journal*, 84, 36–47.
- [23] Ghomi, A., Dolatshahi, M. (2018). Design of a new CMOS Low-Power Analogue Neuron. *IETE Journal of Research*, 64(1), 67-75.