

Review

# Analog Gaussian Function Circuit: Architectures, Operating Principles and Applications

Vassilis Alimisis \*, Marios Gourdouparis, Georgios Gennis, Christos Dimas  and Paul P. Sotiriadis

Department of Electrical and Computer Engineering, National Technical University of Athens, 15780 Athens, Greece; mariosgourd97@gmail.com (M.G.); giorgosyennis@gmail.com (G.G.); chdim@central.ntua.gr (C.D.); pps@ieee.org (P.P.S.)

\* Correspondence: alimisisv@gmail.com

**Abstract:** This review paper explores existing architectures, operating principles, performance metrics and applications of analog Gaussian function circuits. Architectures based on the translinear principle, the bulk-controlled approach, the floating gate approach, the use of multiple differential pairs, compositions of different fundamental blocks and others are considered. Applications involving analog implementations of Machine Learning algorithms, neuromorphic circuits, smart sensor systems and fuzzy/neuro-fuzzy systems are discussed, focusing on the role of the Gaussian function circuit. Finally, a general discussion and concluding remarks are provided.

**Keywords:** Gaussian function circuit; integrated circuits; bulk-controlled; translinear principle; floating-gate transistors; extra components; machine learning applications



check for updates

**Citation:** Alimisis, V.; Gourdouparis, M.; Gennis, G.; Dimas, C.; Sotiriadis, P.P. Analog Gaussian Function Circuit: Architectures, Operating Principles and Applications. *Electronics* **2021**, *10*, 2530. <https://doi.org/10.3390/electronics10202530>

Academic Editor: Inhee Lee

Received: 18 September 2021

Accepted: 11 October 2021

Published: 17 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Since the first Gaussian function circuit (also called the Bump circuit) was introduced by Delbruck (Delbruck's Simple Bump) in 1991 [1,2] (shown in Figure 1), many research groups have focused on improving this architecture and/or incorporating it in various fields [3–14]. Some of the implementations are used for Machine Learning (ML) applications [15,16], neuromorphic systems [17,18], smart sensors [19,20] and fuzzy or neuro-fuzzy systems [21,22]. Gaussian function circuits are specific analog circuits that provide an output current, which is a typical Gaussian function [23] or a bell-shaped curve [24]. The main three characteristics of a Gaussian function curve, shown in Figure 2, are the height (amplitude), the mean value (center) and the variance (width) [1,2]. Depending on the type of the architecture or the application of the Gaussian function circuit can operate in the sub-threshold [25] or the strong inversion region [26].

The first Gaussian function circuit (Simple Bump) is used for computing the similarity of analog voltages (or generally the distance between input signals) [1,2]. The Simple Bump circuit consists of two sub-circuits, a current correlator and a simple differential pair, as shown in Figure 3. The (non-symmetric) current correlator is a compact circuit which consists of four PMOS transistors ( $M_{p1} - M_{p4}$ ), shown in Figure 1. When the current correlator operates only in the sub-threshold region, it has the ability to measure the similarity between two input signals (two input currents  $I_1$  and  $I_2$ ). If one of the transistors operates above the threshold, the correlation is not commutative. This leads to a difficulty in providing the appropriate mathematical model (more complicated expression). Normally, the output current  $I_{out}$  is a self-normalized correlation of the two input currents that resembles the Gaussian function, based on Equation (1), shown in Figure 4. The output current of Delbruck's Simple Bump is given by:

$$I_{out} = \frac{I_{bias} S}{4 \cosh^2 \left( \frac{\kappa V_{in} - V_{mean}}{2} \right)}, \quad (1)$$

where  $I_{bias}$  and  $V_{mean}$  are the bias current and the voltage parameter, controlling the height and the mean value of the Gaussian function curve, respectively. The quantity  $S$  is an important circuit parameter for the simple current correlator, given by Equation (2),  $\kappa$  is the slope factor and  $V_{in}$  is the input voltage.

$$S = \frac{(W/L)_{3,4}}{(W/L)_{1,2}} \tag{2}$$

In the case of the Simple Bump circuit, the height and the mean value are independently tuned via the circuit's parameters, while the deviation is altered by the effective  $W/L$  ratio (via transistors' dimensions). We note that a primary aim in the design flow of most Gaussian function circuits is the independent and electronic tunability of the Gaussian curve's characteristics (height, mean value and variance) [27–29]. By designing a fully-tunable architecture, the Gaussian function circuit can be used as a general purpose circuit in multiple applications [22]. On the other hand, a design lacking electronic tunability is focused on a specific application [9].

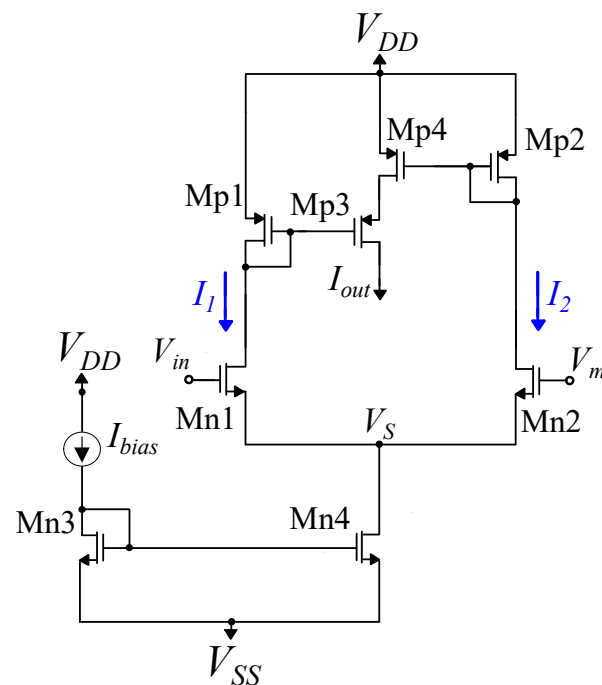


Figure 1. Delbruck's Simple Bump transistor level implementation.

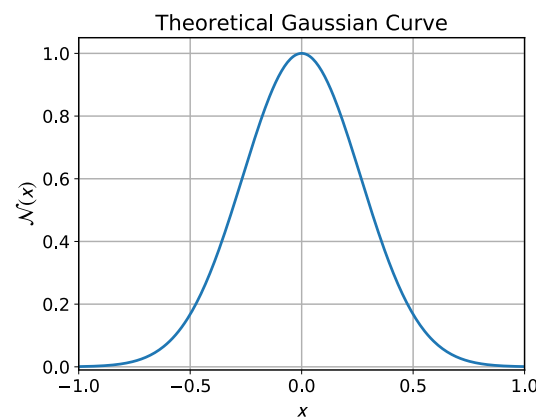
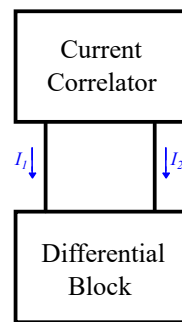
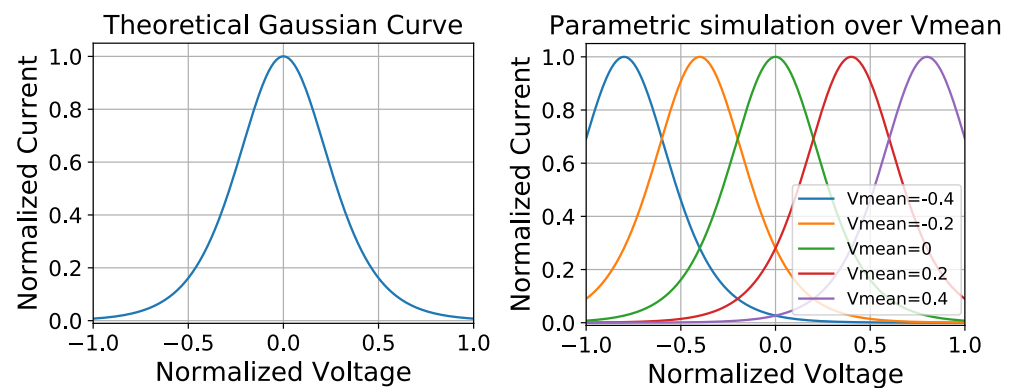


Figure 2. Theoretical Gaussian Function Curve.



**Figure 3.** Block diagram of a generic Gaussian function circuit (inspired by Delbruck's Simple Bump).



**Figure 4.** Theoretical output current of Delbruck's Simple Bump (left) Simulation for  $V_{mean} = 0$  (right) Parametric simulation over  $V_{mean}$ .

In order to properly demonstrate the existing Gaussian function implementations we divide them in five main categories based on their operating principles:

- Architectures based on the translinear principle which use absoluteurs, squarers, current to voltage (I-V) converters, exponentiators and compensators as building blocks;
- Bulk-controlled circuits based on modifications of Delbruck's Simple Bump, that use the body effect in order to tune the variance;
- Circuits including floating-gate transistors that use floating nodes in direct current and capacitively connected inputs in order to achieve tunability in the characteristics;
- Designs using exclusively differential pairs and current mirrors;
- Implementations that add extra components, for example Operational Transconductance Amplifiers (OTAs), Digital to Analog Converters (DACs), mixed-mode circuits, and so forth, which provide the appropriate tunability in the variance.

These categories are not mutually exclusive, since there are a few implementations that may belong in more than one. Each category consists of the circuits with the most possible similarities since an absolute criterion for categorization is difficult. Moreover, there are other implementations, which use different design methodologies for the realization of the Gaussian function. Therefore, we add an extra category (other implementations), which consists of the designs that are not part of the previous categories.

Gaussian function circuits' applications range from low-power and area efficient to high speed computation and high accuracy due to the fact that each application has its own limitations [30,31]. For example, in the case of a wearable classification application, the design of a low-power and area efficient Gaussian function circuit is necessary, because its realization consists of many cells which have to operate in parallel fashion [30]. On the other hand, for object recognition or fuzzy systems, high accuracy, high-speed and real-time computation are needed [31]. The most popular Gaussian function circuits' domains and applications are the following:

- Analog-hardware implementations of ML algorithms, for example Radial Basis Function Neural Networks (RBF NNs), Support Vector Machines (SVMs), the K-means clustering algorithm and so forth;
- Neuromorphic systems, architectures which use physical artificial neurons for computations or design artificial neural systems;
- Smart sensor systems, devices that take input from the physical environment and use built-in computing resources;
- Fuzzy or neuro-fuzzy systems with main applications in controllers and object recognition.

This paper presents an overview of circuits and systems for Gaussian function circuits focusing on integrated implementations. All related architectures, operating principles, design methods and applications, are provided, to the best of the authors' knowledge. The rest of the paper is organized as follows: Architectures and operating principles are reviewed in Section 2. System level implementations and applications are summarized in Section 3. Section 4 discusses and summarizes the performance of different Gaussian function circuits. Concluding remarks are drawn in Section 5.

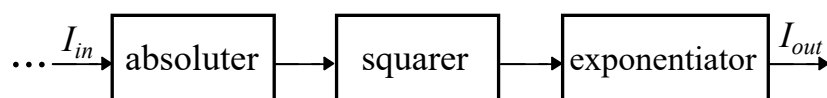
## 2. Architectures and Operating Principles

Many Gaussian function circuits have been proposed in a effort to achieve low power consumption, area efficiency and high accuracy, for example [5,9,32,33]. This Section presents and analyzes the existing Gaussian function circuits, and categorizes them based on their operating principle. The five main categories include the architectures based on the translinear principle, the bulk-controlled approach, circuits built with floating-gate transistors, circuits built exclusively with differential pairs and designs combining different fundamental blocks. Additionally, a sixth category is formed of all other types of Gaussian function circuits.

### 2.1. Architectures Based on the Translinear Principle

The translinear principle, introduced by Barry Gilbert in 1975 [34], results in a direct and elegant methodology to analyze and synthesize circuits realizing certain nonlinear mathematical functions, like multiplication, power-law, etc., using exclusively analog circuits [35,36]. The core of such circuits is a closed translinear loop, containing a number of translinear elements (e.g., bipolar and sub-threshold MOS transistors exhibiting an exponential current–voltage relationship). A typical translinear loop contains an even number of only one type of transistors (p-type or n-type).

Typically, this realization of a Gaussian function circuit is done by implementing and combining the absolute value, the square and exponential functions circuits using the translinear principle [35,36]. The design flow (Figure 5) involves three basic building blocks, an absoluter (for example Figures 6 and 7), a squarer (for example Figures 8 and 9) and an exponentiator (for example Figure 10) [25,37–39]. Some of the implementations combine the absoluter with the squarer or use a squarer directly [3,4,40–44]. The exponentiator can be excluded for the generation of the Gaussian function's logarithm [45]. Some implementations use additional components to improve the accuracy of the Gaussian function. Specifically, an I-V converter (Figure 11) [4,40–42] or a transimpedance amplifier [37] are used between the squarer and the exponentiator. Such components are used because typical translinear squarers have a current output while typical exponentiators have a voltage input. Alternatively, in [43] a compensation circuit is added targeting the reduction of the offset caused by the Body Effect.



**Figure 5.** Design flow diagram for the implementation of the Gaussian function based on the translinear principle.

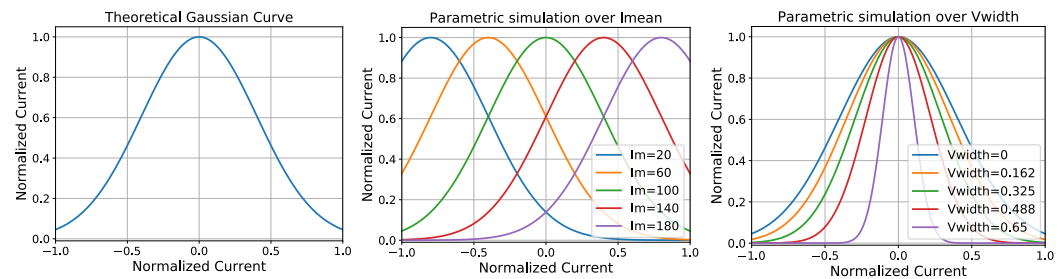




The expression of the output current of a Gaussian function circuit, can be a good approximation of the Gaussian function, or an exact realization (at least with the standard simple models). For example in [25],

$$I_{out} = I_{pre} \cdot \exp \left[ - \frac{\kappa \epsilon (I_x - I_\mu)^2}{I_{bias} k_{tri} (V_{DD} - V_{width}) V_T} \right], \quad (3)$$

where  $I_{pre}$  is the transistor's pre-exponential current,  $\kappa$  is the slope factor,  $\epsilon$  is the generic error term,  $V_T$  is thermal voltage and  $I_{bias}$ ,  $I_m$  and  $V_{width}$  are the bias current and current and voltage parameters, controlling the height, the mean value and the variance of the Gaussian function curve, respectively. The value  $k_{tri} = \mu_p C_{ox} (W/L)_{tri}$ , where  $\mu_p$  is the hole mobility,  $C_{ox}$  is the oxide capacitance per unit area and  $(W/L)_{tri}$  is the effective  $(W/L)$  ratio of the exponentiator input transistor. The theoretical output current of the Gaussian circuit, according to (3), is presented in Figure 12.



**Figure 12.** Theoretical output current of an example of a circuit based on the Translinear Principle (left) Simulation for  $I_m = 100$  and  $V_{width} = 0.4$  (center) Parametric simulation over  $I_m$  for  $V_{width} = 0.4$  (right) Parametric simulation over  $V_{width}$  for  $I_m = 100$ .

In Table 1, we summarize the technology used, the minimum operational characteristics (power consumption, power supply, bias current), the operation region and the number of transistors for each implementation. The power consumption for the Gaussian function circuit ranges from 350 nW to 1.534  $\mu$ W, the power supply ranges from 0.7 V to 3.3 V with the minimum operational bias current being less than 0.8  $\mu$ A (except from [44]). All of the architectures are designed to operate in the sub-threshold region, except from [3,44]. The number of the transistors is usually high, as the Gaussian function is implemented with multiple stages. Though, it should be noted that the Gaussian function's dimensionality can be increased by adding extra squarers and using the same exponentiator circuit, therefore decreasing the overall number of transistors compared to a fully cascaded implementation. In the case of [4,39], the number of transistors and the presented power consumption refers to the realized Support Vector Regression Algorithm [4] or Self Organized Map (SOM) [39], respectively.

**Table 1.** Gaussian Function Circuits based on the translinear principle. \* For the entire system.

Ref.	Technology	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[3]	-	-	3 V	325 nA	above threshold	14
[4]	180 nm	-	1.8 V	50 nA	sub-threshold	* 600
[25]	180 nm	485 nW	0.7 V	10 nA	sub-threshold	14
[37]	1.2 $\mu$ m	200 $\mu$ W	1.2 V	4 nA	sub-threshold	22
[38]	180 nm	350 nW	0.7 V	50 nA	sub-threshold	31
[39]	180 nm	* 1.1 mW	1.5 V	0.8 $\mu$ A	above and sub-threshold	55
[40]	0.35 $\mu$ m	650 nW	1.3 V	90 nA	sub-threshold	17
[41]	180 nm	-	1.8 V	50 nA	sub-threshold	14
[42]	180 nm	-	1.8 V	50 nA	sub-threshold	14
[43]	0.6 $\mu$ m	843 nW	1.5 V	10 nA	sub-threshold	22
[43]	0.6 $\mu$ m	1.534 $\mu$ W	1.5 V	40 nA	sub-threshold	26
[44]	0.35 $\mu$ m	-	3.3 V	10 $\mu$ A	above threshold	45
[45]	-	-	-	-	sub-threshold	-

## 2.2. Bulk-Controlled Implementations

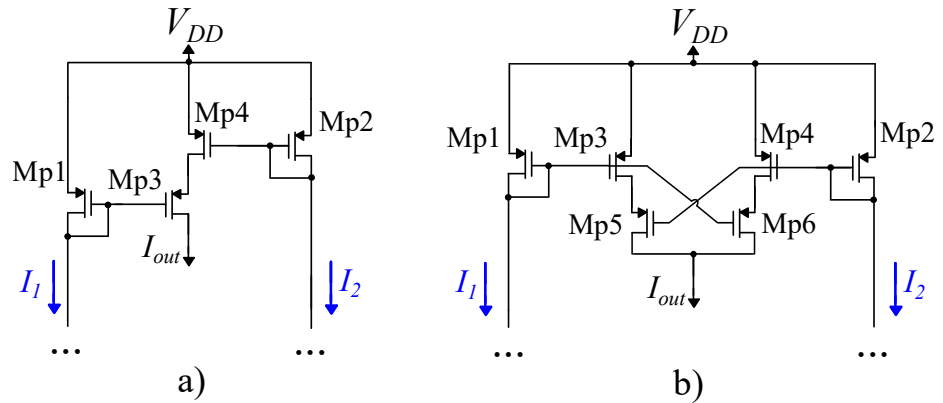
MOS transistors are four-terminal devices (Gate, Drain, Source, Bulk), in which traditionally the Gate terminal is used as a signal input. Depending on the type of the CMOS technology (for example P-well, N-well, twin-tub), the Bulk terminal is usually connected to either the negative (for NMOS) or the positive (for PMOS) supply voltage or even the related Source terminal (isolating the Bulk from the P-substrate) [46–51]. However, there are cases (PMOS transistors, triple N-well technologies) in which voltage signals are applied to the Bulk terminal directly. By using bulk-driven or bulk-controlled transistors the voltage threshold limitation is reduced from the signal path [46–51]. Therefore, lower power supply voltages and bias currents are available and hence, using mainly sub-threshold region techniques, the power consumption is decreased. Additionally, the control voltage connected to the Bulk offers a wide range tunable parameter, directly affecting the transistor's Drain current.

The aforementioned benefits motivated researchers to implement new Gaussian function circuits built with bulk-controlled transistors, which achieve electronic tunability in the Gaussian output curve's variance. The variance tunability is also enhanced by altering Delbruck's Simple Bump [1,2], which consists of a non-symmetric current correlator, shown in Figure 13a and a simple differential pair. Some of the proposed modifications include a symmetric current correlator [24,32], shown in Figure 13b, a differential difference pair [5,6,17,18,24,52–54], shown in Figure 14 and/or adding additional transistors to the standard differential pair [24,27,32,55]; an example is shown in Figure 15. Any combination of current correlators and differential blocks can implement a Gaussian function curve, for example, Figure 16. Moreover, there are researchers who made significant modifications to Delbruck's Simple Bump [17,18,24,52–54]. Specifically, the variable width (VW) Bump, shown in Figure 17, used in [17,18,52–54], adds multiple current mirrors along with the (non symmetric) current correlator and the [24] combines the symmetric current correlator with current substructures.

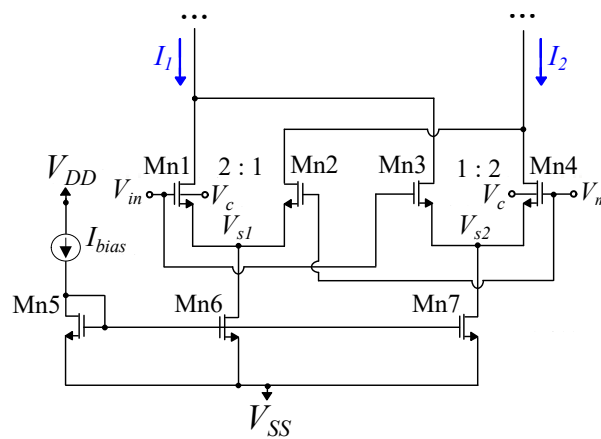
In general, the desired tunability in the Gaussian function curve's variance is achieved by connecting a control voltage to the Body node of some of the differential block's transistors. By incorporating a differential difference pair [5,6,17,18,24,52–54] an increased linearity [56], along with additional output currents are provided. In either case, the control voltage provides sigmoidal shaped curves  $I_1$ ,  $I_2$  with adjustable slopes or symmetric displacement of currents. The correlation of these currents results in a tunable Gaussian function circuit. The standard current correlator is, by design, not an ideal circuit, having inherent asymmetries in the output current. The output of a symmetric current correlator



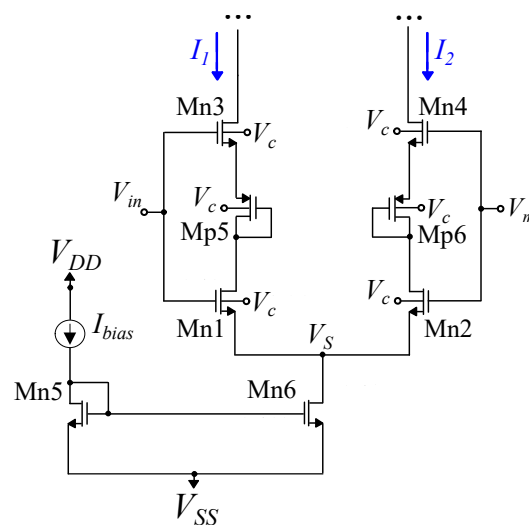
is the sum of two non symmetric current correlator cells and therefore reduces such asymmetries. By using this correlator, the Gaussian function is realized more accurately in the cost of increased power consumption and circuit's complexity.



**Figure 13.** Transistor level implementation of (a) a non symmetric current correlator and (b) a symmetric current correlator.



**Figure 14.** Transistor level implementation of a differential difference block.



**Figure 15.** Transistor level implementation of a versatile differential block.



The analysis of the bulk-controlled designs is based on the MOS model described in [57] and since all transistors operate in the sub-threshold region (domain), the currents for the PMOS and NMOS devices are, respectively:

$$I_{pmos} = I_{op} e^{\kappa_p (V_w - V_G) / V_T} \left( e^{(V_S - V_w) / V_T} - e^{(V_D - V_w) / V_T} \right), \tag{4}$$

$$I_{nmos} = I_{on} e^{\kappa_n (V_G - V_w) / V_T} \left( e^{(V_w - V_S) / V_T} - e^{(V_w - V_D) / V_T} \right). \tag{5}$$

Here,  $\kappa_p$  and  $\kappa_n$  are the slope factors for PMOS and NMOS transistors, respectively,  $V_G$ ,  $V_S$ ,  $V_D$  and  $V_w$  are the gate voltage, source voltage, drain voltage and bulk voltage, respectively,  $V_T$  is the thermal voltage and  $I_{op}$  and  $I_{on}$  are the characteristic currents (pre-exponential current) for PMOS and NMOS transistors, respectively [57]. Specifically, by using (4) and (5) the output current of [5] is expressed as:

$$I_{out} = \frac{3I_{bias}}{2} \frac{12 + 3M^2 + 12M \cosh x}{(2 \cosh x + M)(6e^x + 4e^{-x} + 5M)}, \tag{6}$$

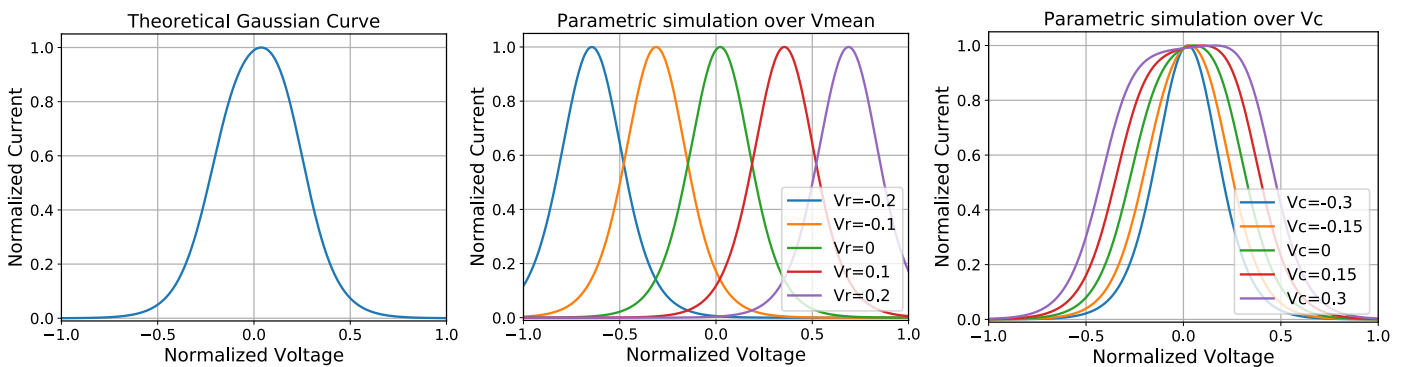
where the variable  $M$  is defined as:

$$M = 2e^{-(\kappa_n - 1)(V_c - V_{ss}) / V_T} + \frac{e^{(\kappa_n - 1)(V_c - V_{ss}) / V_T}}{2}, \tag{7}$$

and parameter  $x$  is given by:

$$x = \frac{\kappa_n (V_r - V_{in})}{V_T}. \tag{8}$$

Here,  $V_{ss}$  is the lower supply voltage,  $V_{in}$  is the input voltage and  $I_{bias}$ ,  $V_r$  and  $V_c$  are the bias current and the voltage parameters, controlling the height, the mean value and the variance of the Gaussian function curve, respectively. This circuit [5] consists of a non-symmetric current correlator and a differential difference pair. The theoretical output current of the Gaussian circuit, according to (6), is presented in Figure 18.



**Figure 18.** Theoretical output function of an example of a bulk-controlled circuit (**left**) Simulation for  $V_r = 0$  and  $V_c = -0.1$  (**center**) Parametric simulation over  $V_r$  for  $V_c = -0.3$  (**right**) Parametric simulation over  $V_c$  for  $V_r = 0$ .

A summary of each implementation’s characteristics is presented in Table 2. This Table includes information regarding the technology used, the minimum operational characteristics (power consumption, power supply, bias current), the operation region and the number of transistors. Regarding the technology, most implementations are in CMOS process, while two of them are tested with discrete components. In the case of [6,24] PTM, transistor models and quad MOS transistor arrays (Advanced Linear Devices) are used, respectively. In general, the power consumption ranges from a few nW to a couple  $\mu$ W, with the power supply being lower than 1.8 V (except from [6]). Moreover, for [54], the power consumption of the entire system instead of a single bump cell is provided. The consumption of 50  $\mu$ W is relatively low for the realized application (neuromorphic Spiking NN for Electromyography (EMG) signals). The transistors of all the implementations

operate in the sub-threshold region, and therefore the minimum operational bias currents are less than 50 nA (except from [24]). All of the architectures add additional transistors to achieve independent tunability of the Gaussian function's characteristics. As a result, their number of the transistors is higher than that of Delbruck's Simple Bump.

**Table 2.** Bulk-controlled Gaussian Function Circuits. \* For the entire System.

Ref.	Technology	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[5]	90 nm	3.9 nW	0.6 V	1 nA	Sub-threshold	11
[6]	Discrete	-	5 V	2 nA	Sub-threshold	10
[17]	180 nm	-	-	5 nA	Sub-threshold	20
[18]	180 nm	-	-	5 nA	Sub-threshold	20
[24]	Discrete	4.1 $\mu$ W	0.7 V	1 $\mu$ A	Sub-threshold	22
[27]	90 nm	6 nW	0.6 V	5 nA	Sub-threshold	10
[32]	90 nm	3.3 nW	0.6 V	3 nA	Sub-threshold	14
[52]	180 nm	-	1.8 V	50 nA	Sub-threshold	15
[53]	180 nm	-	1.8 V	50 nA	Sub-threshold	15
[54]	180 nm	* 50 $\mu$ W	1.8 V	50 nA	Sub-threshold	15
[55]	90 nm	4 nW	0.6 V	3 nA	Sub-threshold	10

### 2.3. Circuits Built with Floating-Gate Transistors

The floating-gate transistor or, as it is also called, Floating-gate MOSFET (FGMOS), is a type of complementary metal-oxide semiconductor transistor or metal-oxide-semiconductor field-effect transistor, which has the ability to hold an electrical charge in a memory device that is used to store data [58–60]. In comparison with a typical MOS, it has an additional terminal (electrode) between the gate and the semiconductor. The name floating is derived from the floating gate terminal, which is not connected to a voltage source. As in a typical MOS all the other terminals (Gate, Drain, Source, Bulk) can be connected with a voltage source. In the case of a high current, electrons get stuck in the floating terminal. Due to the fact that the floating gate is not connected to anything, it can maintain the stored data.

In all the presented implementations, a classic bump circuit architecture is modified by replacing existing transistors with floating gate ones or by simply adding extra floating gate transistors. There are implementations directly inspired by Delbruck's Simple Bump [1,2], which add an inverse generation stage [61,62], shown in Figure 19 or a folded differential pair [63]. Moreover, a compact design based on Delbruck's non-symmetric current correlator with an integrated differential pair, using floating gate input transistors, is proposed in [64], shown in Figure 20. Similarly, there are architectures inspired by [65], replacing the input transistors with floating gate ones [66–68], depicted in Figure 21. Some designs also incorporate floating gate transistors in architectures following the mathematical approach of the translinear principle. Specifically, [26,69] modify the exponentiator; [26] is shown in Figure 22, [7] creates a squaring circuit using floating gate transistors and [8] enhances a Gilbert multiplier [70] with a floating gate memory cell.

In general, the Gaussian function curve's characteristics are controlled via the floating gate transistor's parameters. In particular, the voltage stored in the floating terminal can be used as an additional parameter. In the case of [61–63], an inverse generation block composed of floating gate transistors is used, shown in Figure 19. This block provides the appropriate input voltages to the variable gain amplifier (VGA). By altering the gain of the VGA, the width tunability is achieved. An architecture which replaces the input transistors of Delbruck's Simple Bump with FGMOS ones is provided in [64]. It uses FGMOS in order to subtract the stored mean value from the gate input and achieves the width tunability by setting the value of the input capacitors. A simple exponential based design is presented in [66–68]. It consists of three MOS and two FGMOS transistors, shown

in Figure 21. The height of the Gaussian output curve is controlled via the parameter  $V_{GG}$  and References [67,68] achieve the width tunability using extra digital components and signals. References [7,8,26,69] are based on the translinear principle and use FGMOS to reduce the complexity (number of transistors) of typical translinear architectures.

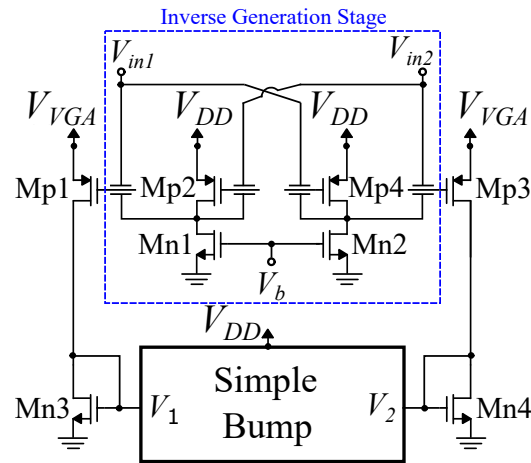


Figure 19. Schematic of a Gaussian function circuit with a floating gate transistor based inverse generation block.

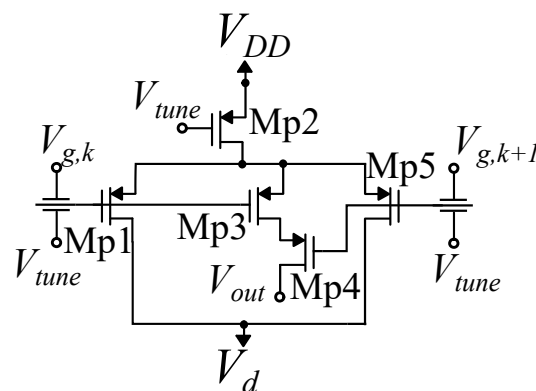


Figure 20. A floating gate transistor based modification of a bump–antibump circuit.

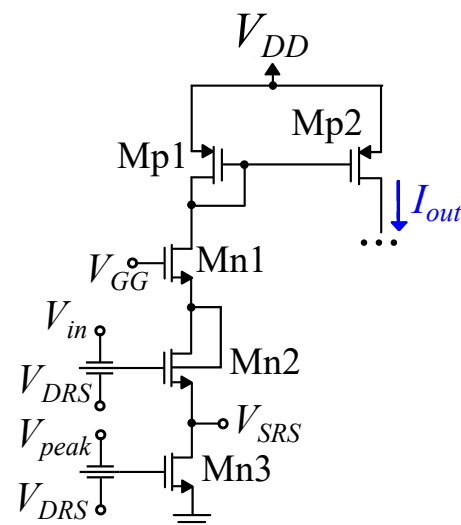
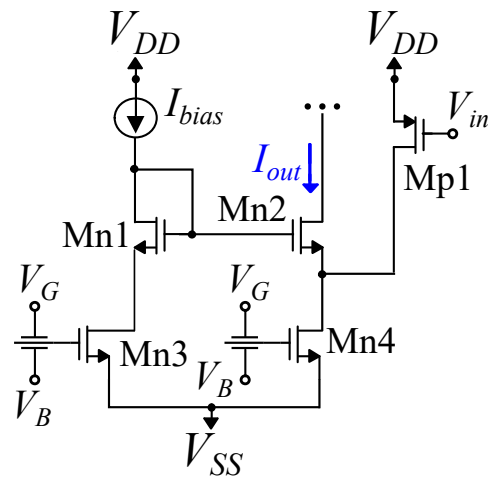


Figure 21. A modification of an exponential-based Gaussian function circuit, using floating gate transistors.



**Figure 22.** Floating gate transistor level architecture based on an exponentiator circuit for the implementation of the Gaussian function.

The output current of a circuit incorporating floating gate transistors depends on the original (without floating gate transistors) architecture. For a design inspired by the Simple Bump (specifically for [63]) the output is given by:

$$I_{out} = \frac{2I_{bias}}{2 + e^{\kappa\gamma(V_{in}-V_{mean})/V_T} + e^{-\kappa\gamma(V_{in}-V_{mean})/V_T}}, \tag{9}$$

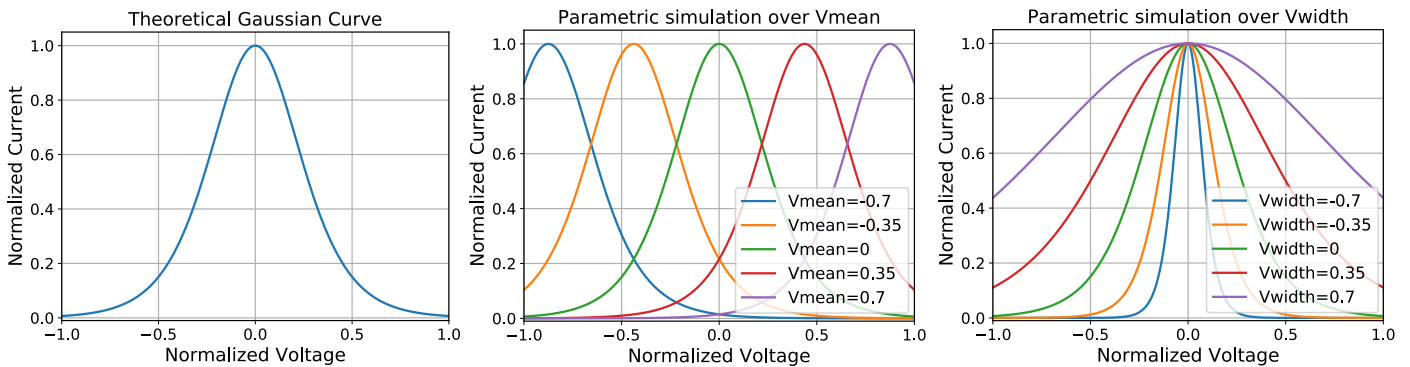
where variable  $\gamma$  is defined as:

$$\gamma = -2V_T \frac{I_0}{I_b} e^{-\beta_1 V_{width}}, \tag{10}$$

where  $\beta_1$  is set as:

$$\beta_1 = \frac{C_c}{V_T(C_c + C_d)}. \tag{11}$$

Here,  $\kappa$  is the slope factor,  $V_T$  is the thermal voltage,  $I_{bias}$ ,  $V_{mean}$  and  $V_{width}$  are the bias current and the voltage parameters, controlling the height, the mean value and the variance of the Gaussian function curve, respectively, and  $V_{in}$  is the input voltage.  $I_b$  is a tail current of the core differential pair,  $I_0$  is the pre-exponential current.  $C_c$  and  $C_d$  are floating gate input capacitances. Gaussian function curves based on Equation (9) are shown in Figure 23.



**Figure 23.** Theoretical output function of an example of a floating gate based circuit (**left**) Simulation for  $V_{mean} = 0$  and  $V_{width} = 0$  (**center**) Parametric simulation over  $V_{mean}$  for  $V_{width} = 0$  (**right**) Parametric simulation over  $V_{width}$  for  $V_{mean} = 0$ .

The characteristics of each implementation are summarized in Table 3. The power consumption for the presented circuits is higher than 90  $\mu$ W (except from [7]), with a power

supply ranging from 0.75 V to 10 V and a minimum operational bias current varying from a couple nA to a couple  $\mu\text{A}$ . The power consumption of [66] refers to the realized handwritten digit recognition system. The number of the transistors depends on the original architecture and the methodology used to achieve the electronic tuning (replacing existing transistors or adding new ones). Regarding the operation regime, there are implementations in both the above and the sub-threshold regions.

**Table 3.** Gaussian Function Circuits using Floating Gate Transistors. \* For the entire system.

Ref.	Technology	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[7]	180 nm	160 nW	0.75 V	35 nA	sub-threshold	8
[8]	-	-	10 V	10 $\mu\text{A}$	above threshold	-
[26]	0.25 $\mu\text{m}$	214 $\mu\text{W}$	3.3 V	-	above threshold	5
[61]	0.5 $\mu\text{m}$	90 $\mu\text{W}$	3.3 V	-	above and sub-threshold	15
[62]	0.5 $\mu\text{m}$	-	3.3 V	-	sub-threshold	15
[63]	0.5 $\mu\text{m}$	-	1.6 V	200 nA	sub-threshold	16
[64]	0.5 $\mu\text{m}$	-	-	5 $\mu\text{A}$	above threshold	5
[66]	0.6 $\mu\text{m}$	* 6 mW	5 V	100 nA	above or sub-threshold	6
[67]	0.6 $\mu\text{m}$	-	5 V	1 $\mu\text{A}$	above or sub-threshold	6
[68]	0.6 $\mu\text{m}$	-	5 V	100 nA	above threshold	6
[69]	180 nm	100 $\mu\text{W}$	1.8 V	10 $\mu\text{A}$	above threshold	14

#### 2.4. Circuits Built Exclusively with Differential Pairs

Some researchers follow a simpler approach to realize Gaussian function circuits and base their designs on multiple differential pairs. The Gaussian function is formed by adding and subtracting currents using mainly the differential pairs and current mirrors, unlike the Delbruck inspired architectures that combine differential pairs with current or voltage correlators. Most of the architectures have the same operating principles. Some of the implementations produce a Gaussian function curve by adding two currents (generated from a different input voltage) from two differential pairs [10,21,23,71–75]. A characteristic example of such circuits is shown in Figure 24. In [75] an extra resistor is used to determine the height of the Gaussian function curve. Some architectures follow the same principle but use folded cascode differential pairs [76], include multiple mirrors [77,78] or produce more than one Gaussian curve [79]. Furthermore, some designs [9,80] are inspired by Gilbert's Gaussian circuit, shown in Figure 25, which is not fundamentally different from the previous implementations but is based on the Gilbert multiplier [70], an example is shown in Figure 26.

The output current of a typical circuit built exclusively with differential pairs (for example [23]) is given by:

$$I_{out} = \frac{\beta_2}{2} \left( V_{mean} - V_{in} + \sqrt{\frac{2I_{bias}}{\beta_1}} \right)^2, \quad (12)$$

where  $\beta_i = K \cdot W_i/L_i$ ,  $i = 1, 2$  representing the input transistors,  $K$  is a process related constant,  $V_{mean}$  and  $I_{bias}$  are the voltage and current parameters controlling the mean value and height of the Gaussian curve, respectively, and  $V_{in}$  is the input voltage. The variance is controlled via the parameter  $\beta_1$ . The theoretical output current of the Gaussian circuit, according to (12), is presented in Figure 27.

A summary that includes the technology used, the minimum operational characteristics (power consumption, power supply, bias current), the operation region and the number of transistors for each implementation is presented in Table 4. Regarding the technology used, all the implementations are in the CMOS process, with the exception of [10,76], which

are tested with discrete components. All the provided power consumptions, mentioned in Table 4 refer to the realized system (except from [23]). The power supply is generally higher than 2 V with most of the implementations operating mainly in the above threshold region and the minimum operational bias current being around a few  $\mu\text{A}$ . Regarding the number of transistors, most implementations are generally compact. The implementation with the minimum number of transistors (only four) is categorized here. Moreover, the implementations that are marked with a single star, have a simplified schematic where the bias transistors are replaced with current sources and therefore the actual number of transistors is higher.

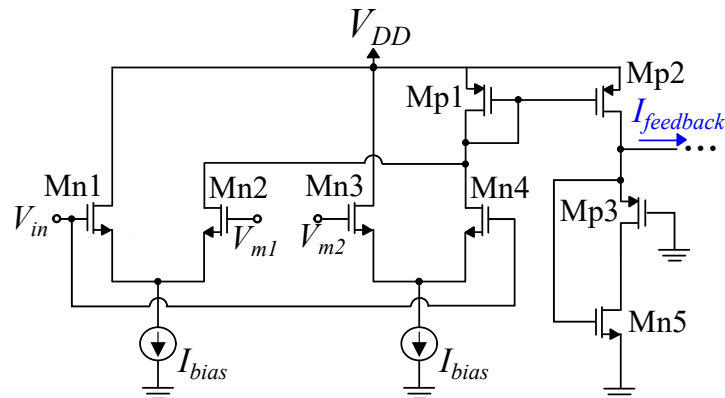


Figure 24. An example of a Gaussian function circuit using only differential pairs.

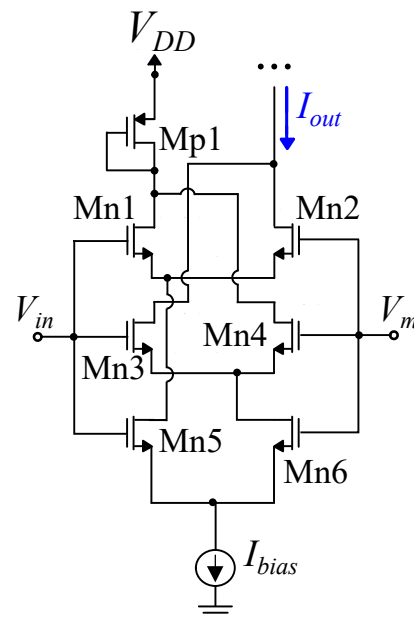


Figure 25. Gilbert's Gaussian circuit.



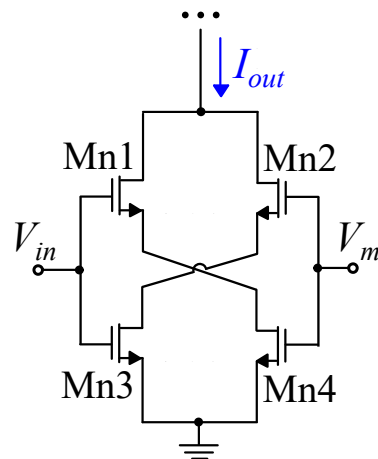


Figure 26. An area-efficient version of Gilbert’s Gaussian circuit.

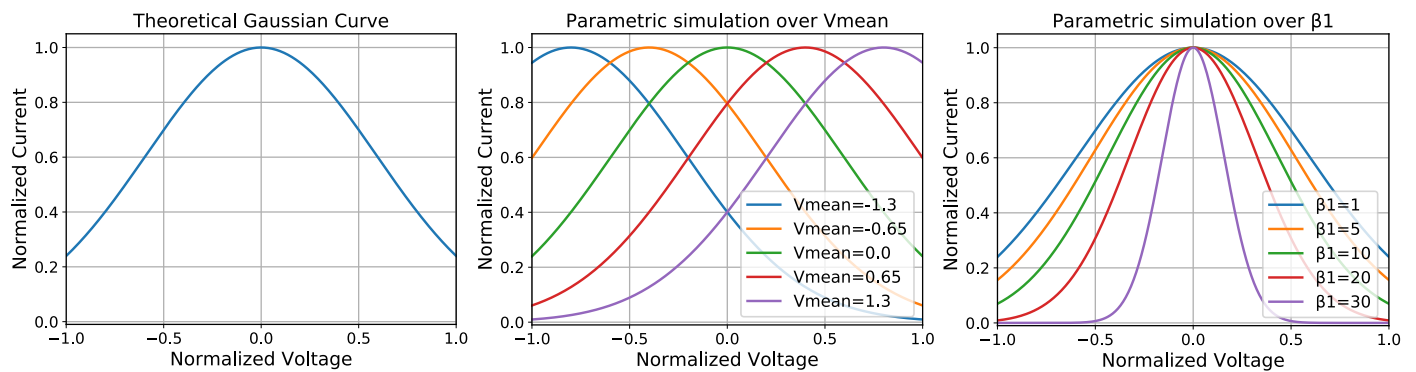


Figure 27. Theoretical output function of an example of a circuit built with differential pairs (left) Simulation for  $V_{mean} = 0$  and  $\beta_1 = 1$  (center) Parametric simulation over  $V_{mean}$  for  $\beta_1 = 1$  (right) Parametric simulation over  $\beta_1$  for  $V_{mean} = 0$ .

Table 4. Gaussian function circuits built with differential pairs. \* Additional current sources or resistors. \*\* Power consumption for the entire System.

Ref.	Technology	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[9]	180 nm	-	1.3 V	-	above and sub-threshold	4
[10]	Discrete	-	3 V	-	above threshold	8
[21]	-	-	10 V	10 $\mu$ A	above threshold	* 8
[23]	0.35 $\mu$ m	105 $\mu$ W	3.3 V	10 $\mu$ A	above threshold	15
[71]	0.7 $\mu$ m	** 45 mW	5 V	100 nA	above threshold	-
[72]	180 nm	** 20 mW	3.3 V	100 $\mu$ A	above threshold	9
[73]	180 nm	** 2.64 mW	3.3 V	100 $\mu$ A	above threshold	9
[74]	1.6 $\mu$ m	-	5 V	15 $\mu$ A	above and sub-threshold	* 11
[75]	180 nm	** 1 mW	3.3 V	-	above threshold	* 8
[76]	Discrete	-	3 V	1.4 $\mu$ A	above threshold	* 8
[77]	2 $\mu$ m	-	3 V	10 $\mu$ A	above threshold	15
[78]	1.2 $\mu$ m	-	10 V	5 $\mu$ A	above threshold	15
[79]	0.35 $\mu$ m	** 2.54 mW	3.3 V	10 $\mu$ A	above threshold	* 19
[80]	2 $\mu$ m	-	10 V	50 $\mu$ A	above threshold	* 10

### 2.5. Designs Incorporating Extra Components

There are implementations that use extra components in order to realize a tunable Gaussian function circuit. In most cases, digital or other analog circuits are attached

around a partially tunable Gaussian or Gaussian-like function circuit (Figure 28) to improve the tunability of the Gaussian function curve’s three characteristics (height, mean value and variance). The extra components include multiplexers (MUX) [11,81] and/or switches [15,82–85] and other digital circuitry (mixed-mode architectures [86]) [82–84], series of resistors [19,87,88], DACs [22,28,87] or Analog to Digital converters (ADC) [22], multipliers [28,31,33,87] or tunable current mirrors [89,90], OTAs [91–94] or other amplifiers [12,87], common mode feedback circuits (CMFB) [22,95], squarers [33], exponentiators [87], current-controlled current-conveyor second generation (CCII) circuits [90], minimum value circuits [96] and additional current correlators [97]. Four representative examples are provided in Figures 29–32. By adding such components, the power consumption and the area of the system are increased but more versatile circuits are created.

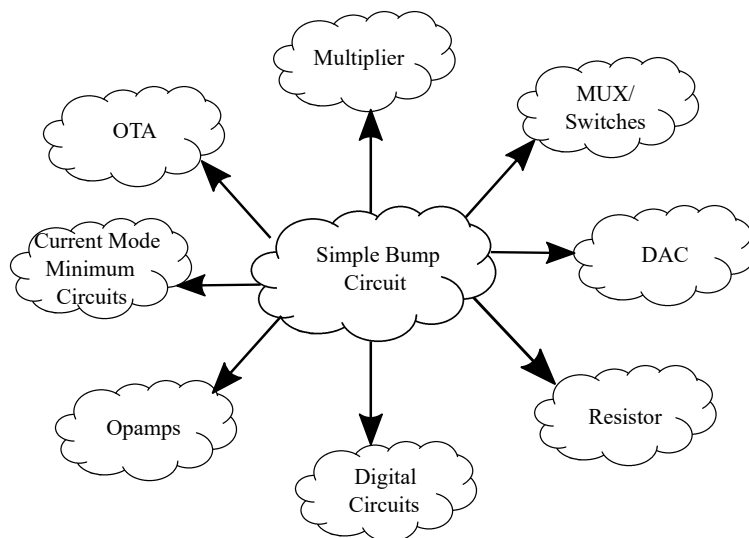


Figure 28. A general flow chart, presenting potential extra components of fully tunable Gaussian function circuit architectures.

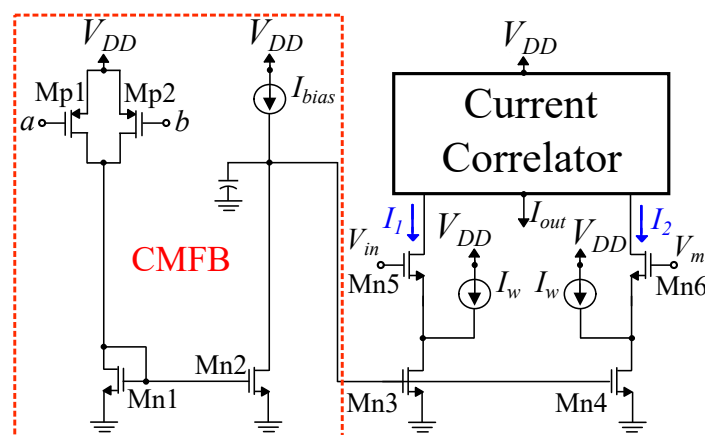


Figure 29. A Gaussian function circuit with a CMFB stage.

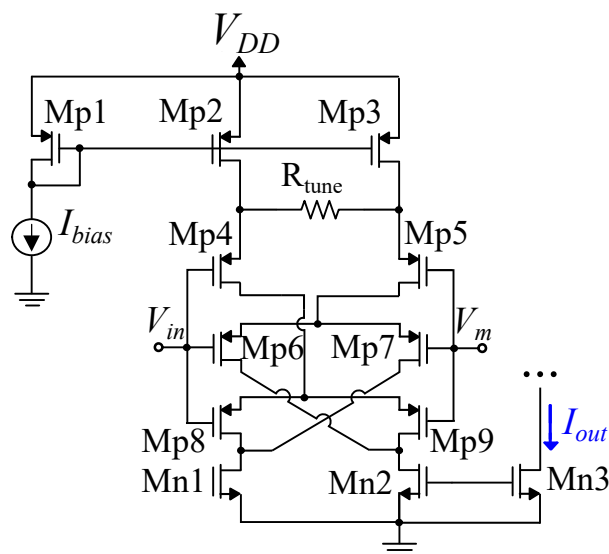


Figure 30. A modified Gilbert's Gaussian circuit, using a series of resistors controlled via switches.

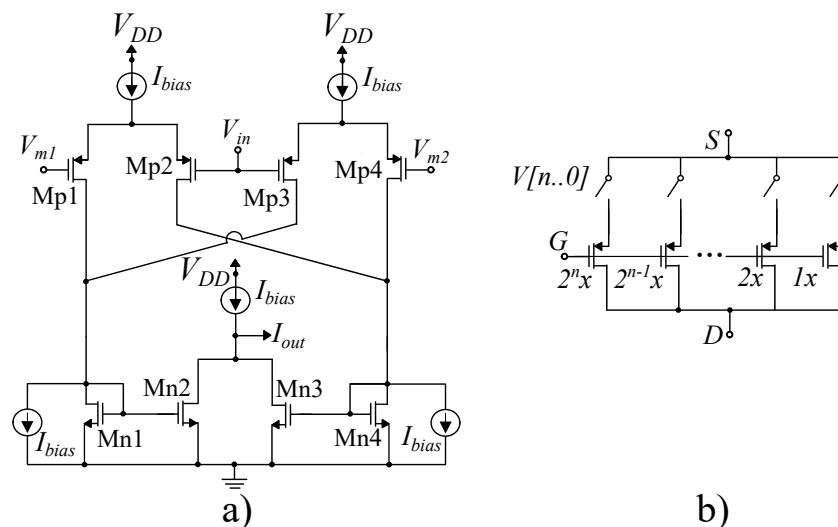


Figure 31. A Gaussian function circuit using a multiplexer to adjust the multiplicity of the transistors in the differential pairs. (a) schematic (b) block for selecting transistors with different dimensions via the multiplexer.

Each implementation uses the extra components differently, but the general concept regarding the added extra components is to enhance the operation of a simple Gaussian function core, for example, to provide variance tunability to the circuit. In particular, multiplexers and switches are used to select the appropriate value from multiple parallel outputs in order to achieve the tunability in the variance [11,15,81–85]. In a similar manner, the series of resistors alter the Gaussian function output by changing the total resistance value [19,87,88]. Moreover, DACs, multipliers, squarers or tunable current mirrors usually directly affect the height of the Gaussian function [22,28,31,33,87,89,90]. There are implementations that use OTAs as current to voltage converters [91] or deploy three OTAs along with multiple resistors as basic building blocks to design tunable Gaussian function circuits [92–94]. Similarly, CCIs, exponentiators, additional current correlators or minimum value circuits are used as basic building blocks in [87,90,96,97]. The operational amplifier in [87] is used to bias a BJT transistor in the exponential region, while the sense amplifier in [12] operates as a CMFB, similar to the extra components in [22,95].

In Table 5, we summarize the characteristics of each implementation. The provided power consumption for most of the implementations refers to the realized system and

varies from a couple to many mW (except from [88]). For the rest, the power consumption ranges from 13.5 nW to 220  $\mu$ W. The increased power consumption is reasonable due to the scale of the extra components. The power supply is different for each application, with most implementations operating above the threshold (in the saturation region) and the minimum operational bias current ranges from nA to  $\mu$ A. There are implementations for which the provided power consumption or number of transistors, mentioned in Table 5, refer to the bump circuit core without including all or any of the added components. These implementations are marked accordingly.

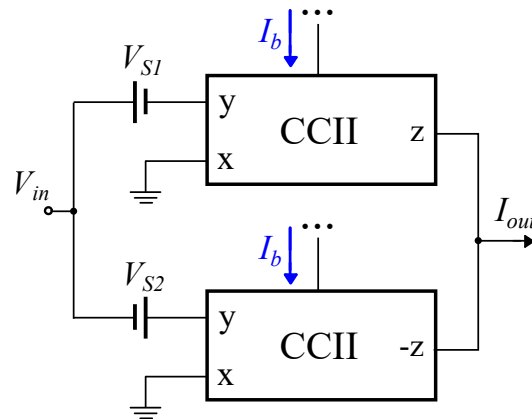


Figure 32. A current conveyor second generation based Gaussian function circuit.

Table 5. Gaussian Function Circuits using extra components. \* Does not include the extra components. \*\* For the entire system.

Ref.	Technology	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[11]	180 nm	* 13.5 nW	0.9 V	-	sub-threshold	* 9
[12]	-	** 2 mW	-	-	sub-threshold	* 6
[15]	130 nm	** 2.2 mW	1.2 V	1.2 $\mu$ A	above threshold	*14
[19]	45 nm	200 nW	-	-	sub-threshold	12
[22]	130 nm	** 13.92 mW	1 V	-	above threshold	* 19
[28]	180 nm	-	1.8 V	100 nA	sub-threshold	* 11
[31]	130 nm	** 496 mW	1.2 V	-	above threshold	12
[33]	180 nm	100 $\mu$ W	1 V	10 $\mu$ A	above threshold	30
[81]	130 nm	10.5 $\mu$ W	2 V	722 nA	above threshold	* 23
[82]	130 nm	** 1.2 mW	1.2 V	1.2 $\mu$ A	above threshold	* 8
[83]	130 nm	** 345 mW	1.2 V	1.2 $\mu$ A	above threshold	* 8
[84]	130 nm	** 1.2 mW	1.2 V	1.2 $\mu$ A	above threshold	* 8
[85]	0.35 $\mu$ m	** 13.4 mW	-	18 $\mu$ A	above threshold	* 23
[87]	2 $\mu$ m	-	5 V	-	above threshold	-
[88]	2.4 $\mu$ m	** 550 nW	10 V	-	above threshold	* 4
[89]	0.35 $\mu$ m	220 $\mu$ W	3.3 V	9 $\mu$ A	above threshold	* 14
[90]	180 nm	23.7 $\mu$ W	2 V	5 $\mu$ A	above threshold	32
[91]	-	-	5 V	-	sub-threshold	* 10
[92]	0.8 $\mu$ m	-	5 V	4 $\mu$ A	above threshold	-
[93]	0.8 $\mu$ m	-	5 V	4 $\mu$ A	above threshold	-
[94]	0.8 $\mu$ m	-	5 V	1 $\mu$ A	above threshold	* 36
[95]	130 nm	18.9 nW	3 V	1 nA	sub-threshold	* 14
[96]	180 nm	27 $\mu$ W	1.8 V	2 $\mu$ A	above threshold	15
[97]	3 $\mu$ m	-	5 V	1.2 nA	above threshold	* 9

## 2.6. Other Implementations

The following implementations do not belong to any of the previously described categories and do not fit to a different distinct design methodology. Nonetheless, they can be added into a general group. In particular, there are architectures [13,14,20,30] based on Delbruck's Simple Bump, designs [65,98] inspired by Anderson [12] or based on other function generation circuits, like a triangular [29,99], an exponential [16,100] or a Euclidean [101]. An example for each group is shown in Figures 33–36, respectively.

The characteristics of each implementation are summarized in Table 6. Regarding the technology, most architectures are in the CMOS process, except for one [101]. In the case of the power consumption, only three designs provide the appropriate value and two of them refer to the entire system's consumption. The power supply ranges from 1.8 V to 5 V and the minimum operational bias current is less than 1  $\mu\text{A}$  (except from [29]). There are implementations operating in either the sub-threshold or the saturation region, with two of them having transistors that operate in the triode region [13,30]. Regarding the number of transistors, most architectures are generally compact. The implementations that are marked with a single star have simplified the schematic by replacing the bias transistors with current sources or have added resistors or capacitors to their designs.

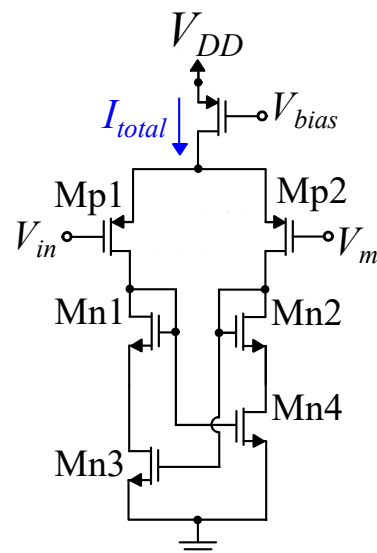


Figure 33. Gaussian function circuit with voltage correlator.

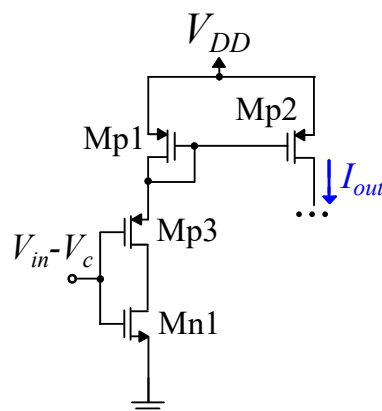


Figure 34. A design inspired by Anderson.

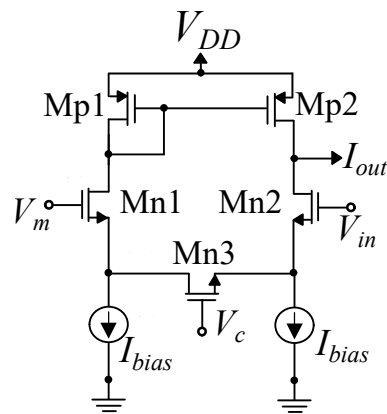


Figure 35. An example of a Gaussian function circuit based on an exponentiator circuit.

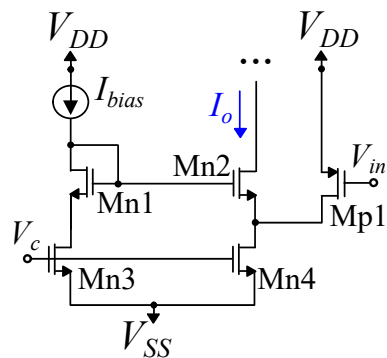


Figure 36. A different example of a Gaussian function circuit based on an exponentiator circuit.

**Table 6.** Gaussian Function Circuits with other implementations. \* Additional current sources, resistors or capacitors.  
\*\* Power consumption for the entire System.

Ref.	Technology	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[13]	0.35 $\mu\text{m}$	-	-	-	above threshold	10
[14]	-	-	-	-	sub-threshold	12
[16]	2.4 $\mu\text{m}$	-	5 V	0.5 $\mu\text{A}$	above threshold	* 5
[20]	180 nm	** 0.9 $\mu\text{W}$ per pixel	1.8 V	-	above threshold	8
[29]	180 nm	23.7 $\mu\text{W}$	2 V	3 $\mu\text{A}$	above threshold	* 22
[30]	0.35 $\mu\text{m}$	-	-	-	above threshold	10
[65]	2 $\mu\text{m}$	-	3 V	200 nA	above or sub-threshold	* 4
[98]	2 $\mu\text{m}$	-	3 V	200 nA	above or sub-threshold	* 4
[99]	2 $\mu\text{m}$	** 0.7 mW	5 V	0.5 $\mu\text{A}$	sub-threshold	14
[100]	3 $\mu\text{m}$	-	5 V	1 $\mu\text{A}$	above threshold	5
[101]	Discrete	-	5 V	-	above threshold	* 4

### 3. Gaussian Function Circuit Applications

Gaussian function circuits are used as building blocks in various applications and domains. This Section discusses the applications and describes the role of the Gaussian function circuits in system level implementations. Various realizations are presented and categorized in four main fields. These categories are the following: (a) Analog-hardware implementation of ML algorithms; (b) neuromorphic circuits/systems; (c) smart sensor systems; and (d) fuzzy/neuro-fuzzy systems. The use of Gaussian function circuits in (a)–(d) is extensively explained.

### 3.1. Analog-Hardware ML

The world is filled with a lot of data (words, pictures, videos, etc.) and it does not look like it is going to slow down anytime soon [102,103]. ML provides the promise of deriving meaning from all of that data. As an interdisciplinary field, ML shares common threads with the mathematical fields of statistics, information theory, game theory, and optimization [104,105]. ML is a combination of tools and technology that can be used in order to process all these data. Moreover, all these automated techniques (algorithms) may be able to figure out meaningful patterns (or hypotheses) that may have been missed by the human observer. Traditionally, all these algorithms are implemented in the software. However, there is a trend in which hardware-friendly implementations are used in order to realize these algorithms and models [57,106].

There are three different hardware design approaches with their own advantages and disadvantages. These three approaches are analog, digital, and mixed-mode implementations. In general, digital circuits for ML applications have the advantage that they can achieve high classification accuracy, flexibility, and programmability, but they consume huge power and area due to the large amount of data transaction and high operation speed. On the other hand, specific analog-hardware ML enables low-cost parallelism with low-power computation, but their inaccurate circuit parameters induced by noise and low precision degrade the accuracy. Several mixed-mode architectures took advantage of both analog and digital implementations obtaining low-power consumption within small areas, but it suffers from domain conversion overhead costs.

There are dedicated Analog-hardware architectures for ML algorithms and models that are based on Gaussian function circuits. In Table 7, we summarize some common characteristics of the system level implementations, presented along with the Gaussian function circuit. The proposed ML systems are RBF NNs [11,12,14,16,24,28,61,81,94,101], a general design flow is shown in Figure 37 or other NNs, like a Multi-layer Perceptron (MLP)/RBF network (RBFN) [15] or a Gaussian RBF NN (GRBF NN) [87], Support Vector Machine (SVM) [41,62], regression (SVR) [4] or domain description (SVDD) [42] algorithms, pattern-matching classifiers [66,68], vector quantizers [64,99], a Deep ML (DML) engine [45], a similarity evaluation circuit [67] and an SOM [39]. A typical example of an Analog-hardware implementation of the SVM algorithm is shown in Figure 38. Gaussian function circuits are used for the implementation of two functions that are useful for many ML algorithms: (a) kernel density (b) distance computation. Most of the applications are designed for an input dimensionality lower than 65 dimensions, with some not specifying an upper boundary [15,24,64,87], being able to categorize high definition images. Additionally, the simulation level as well as the circuit area for the layout and chip implementations (or for [39] an estimation), if provided, can be found in Table 7.

### 3.2. Neuromorphic Systems

Traditional computing systems based on the von Neumann architecture are facing many problems related to power efficiency (high power consumption) and memory limitations [102,103]. Indeed, the amount of data to be processed is ever increasing and it is necessary for new computing paradigms. To address these problems, an emerging approach which demonstrates promising results in computing hardware is that of neuromorphic systems [107,108]. This design methodology is inspired by synaptic plasticity in the brain, which is capable of in-memory computing and is suitable for multi-valued or analog arithmetic. The basic building blocks for the implementation of neuromorphic systems are analog spike-based circuits and memristors. There are also design flows which are based on Gaussian function circuits; an example is shown in Figure 39.

Neuromorphic computing represents a novel paradigm for non-Turing computation that aims to reproduce aspects of the ongoing dynamics and computational functionality found in biological brains. This endeavor entails an abstraction of the brain's neural architecture that retains an amount of biological fidelity sufficient to reproduce its functionality while disregarding unnecessary detail. Models of neurons, which are considered

the computational unit of the brain, can be emulated using electronic circuits or simulated using specialized digital systems. Analog designs offer power and area efficiency, which is necessary for large parallel neuron arrays. On the other hand, digital counterparts provide reconfigurability (FPGA), portable or scaled Hardware Description Language (HDL) designs (from one technology to another), invariance to process, variance and temperature (PVT), a simpler design of complex functions and fast design of high-level architectures.

All the applications, the simulation level of the designs and the use or not of memristive devices are summarized in Table 8. A spike-based circuit with a configurable stop-learning feature for always-on online learning applications is presented in [17]. The Bump circuit used (VW Bump) is a necessary building block for the implementation of the Delta-rule (a learning algorithm), which compares the rate of the neuron spikes to a target value. A high accuracy Spiking NN (SNN) based on an error-triggered learning rule with the aforementioned stop-learning capability is explained in [18]. The Bump circuit here (VW Bump) is used in order to indicate the stop-learning and/or the weight update mechanism. A stochastic learning rule based on the stochastic nature of memristors is proposed in [52]. The VW Bump is used similarly with [18]. An SNN architecture is realized in [53] based on synaptic elements and mixed-mode circuits. The VW Bump compares the rate of the neuron spikes to a target value and outputs the direction of the weight update. All the previous designs [17,18,52,53] are based on memristive devices and the simulation results are extracted from the circuit's schematic. A mixed-mode neuromorphic processor for the discrimination of EMG signals is presented in [54]. The EMG signals are converted into spikes using a delta encoding scheme. Here, the VW Bump is used for the weight update mechanism, similarly to the previous implementations [17,18,52,53]. This architecture does not include memristive devices and its performance is verified on a manufactured chip.

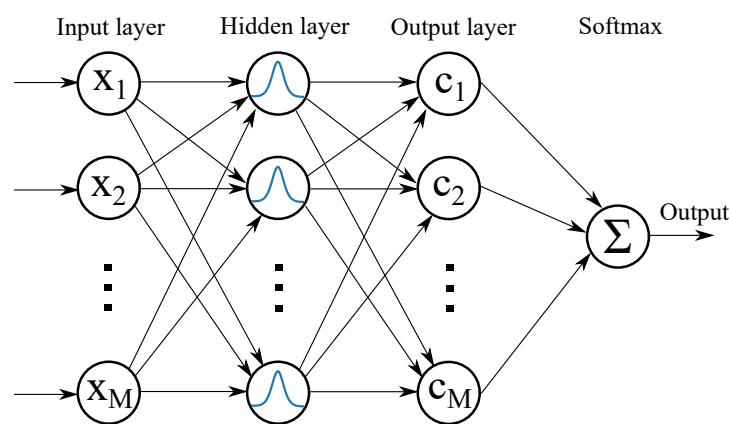
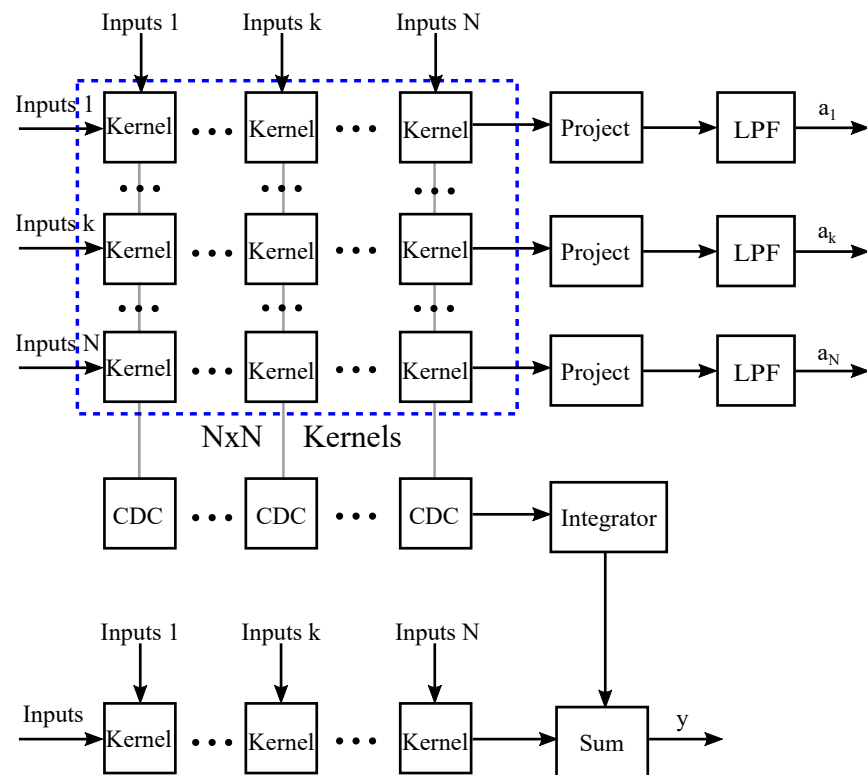


Figure 37. A generic Radial Basis Function Neural Network architecture.





**Figure 38.** A hardware friendly implementation of the Support Vector Machine algorithm (learning and classification).

**Table 7.** Analog-hardware ML algorithm summary. \* tested on LTSpice. \*\* Estimated.

	<b>Implementation</b>	<b>No. of Dimensions</b>	<b>Simulation Level</b>	<b>Area</b>
[4]	SVR algorithm	2	Schematic	-
[11]	RBF NN	1	Chip	0.013 mm <sup>2</sup>
[12]	RBF NN	8	Chip	21.12 mm <sup>2</sup>
[14]	RBF NN	-	Chip	-
[15]	MLP/RBFN	1280 × 720 pixels	Chip	0.140 mm <sup>2</sup>
[16]	LVQ or RBF NN	16	Chip	-
[24]	RBF NN	<i>N</i>	* Layout	10 μm <sup>2</sup> per bump
[28]	RBF NN	2	Chip	0.060 mm <sup>2</sup>
[39]	SOM	3	Schematic	** 0.24 mm <sup>2</sup>
[41]	SVM algorithm	64	Chip	-
[42]	SVDD algorithm	2	Schematic	-
[45]	Deep ML engine	8	Chip	0.36 mm <sup>2</sup>
[61]	RBF NN	2	Chip	2.250 mm <sup>2</sup>
[62]	SVM algorithm	2	Schematic	-
[64]	Vector Quantizer	<i>N</i>	Chip	-
[66]	Pattern-matching classifier	16	Chip	20.25 mm <sup>2</sup>
[67]	Similarity evaluation	4	Chip	-
[68]	Pattern-matching classifier	16	Chip	16,500 μm <sup>2</sup>

Table 7. Cont.

	Implementation	No. of Dimensions	Simulation Level	Area
[81]	RBF NN	-	Chip	68,400 $\mu\text{m}^2$
[87]	GRBF NN	$N$	Schematic	-
[94]	RBF NN	2	Chip	-
[99]	Vector Quantizer	16	Chip	4.95 $\text{mm}^2$
[101]	RBF NN	32	Chip	1 $\text{cm}^2$

Table 8. Neuromorphic Systems Summary.

	[17]	[18]	[52]	[53]	[54]
Application	Stop Learning	Error-Triggered Learning Rule	Stochastic Learning	Neuromorphic Computing	EMG
Memristive devices	YES	YES	YES	YES	NO
Simulation Level	Schematic	Schematic	Schematic	Schematic	Chip

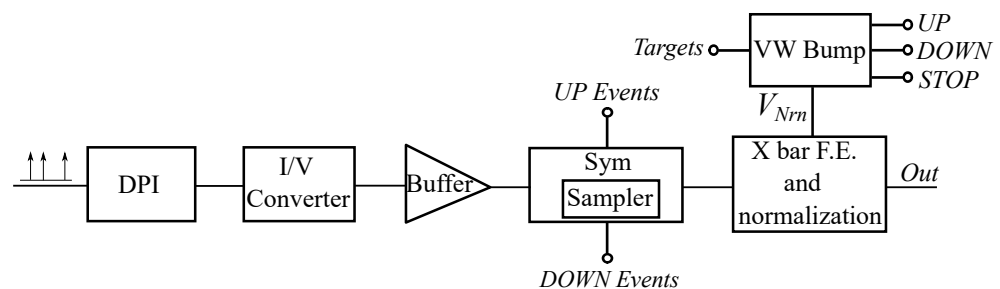


Figure 39. An example of a neuromorphic network high level architecture based on the VW Bump.

### 3.3. Smart Sensor Systems

A typical sensor is a device, sub-system or machine that can detect changes in an environment and it can also send this information to a related electronic system or a simple processor which will derive meaning from these data (signal detection, signal processing, data validation, etc.) [109]. In the case of a smart sensor system, multiple sensors are included [110,111]. Their operating properties can be set by an embedded microprocessor. All the smart sensors have four main functions—measurement, configuration, verification and communication. This means that, apart from a microprocessor, it is necessary to include a wireless communication system. There is a necessity for application-specific integrated circuits, which will be part of smart sensor systems. Except for the implementation of analog front-end, which consists of analog signal conditioning circuitry, there is a need for circuits which will be used for back-end acceleration [103].

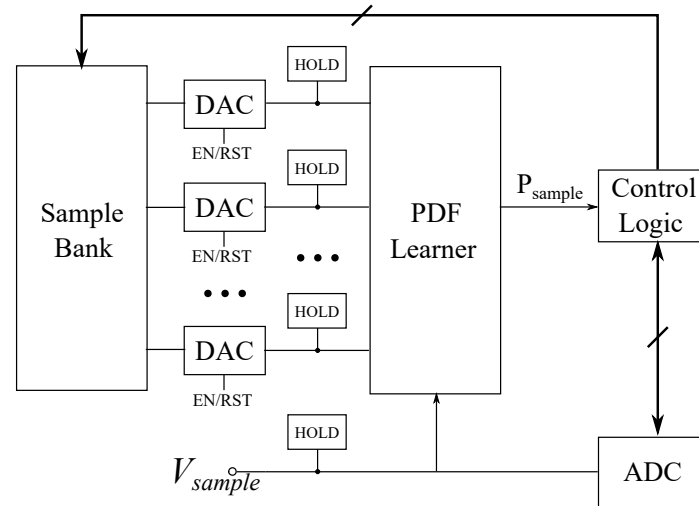
Smart sensor systems are based on analog signal processing because they collect real-time data from the environment. Analog signals are easier to process, best suited for audio and video transmission, have a higher density and can present more refined information. Additionally, they provide a more accurate representation of changes in physical phenomena, such as sound, light, temperature, position, or pressure. The drawbacks are the fact that they are prone to generation loss, are subjected to noise and distortion, as opposed to digital signals, which have much higher immunity and are generally lower quality signals than analog signals. In the case of digital circuits, there is a need for DAC/ADC converters and they suffer from round-off noise due to quantization.

Gaussian function circuits are used as a building block for the implementation of the detector in smart sensor systems. Specifically, two exemplary implementations exist in the literature [19,20], to the best of the author's knowledge, and their characteristics are summarized in Table 9. The first [19] is a mixed-mode real-time anomaly detection system for sensor stream statistics, shown in Figure 40. This system is operational in any type

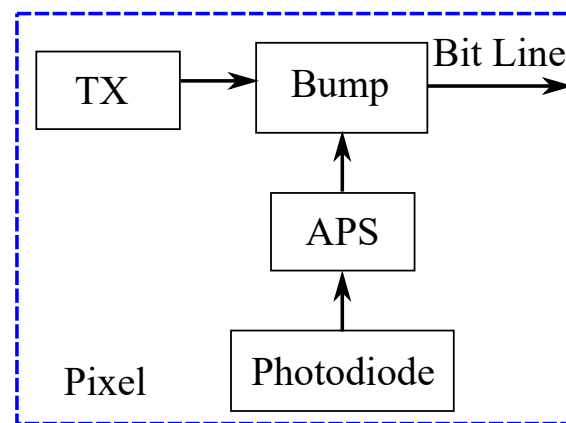
of sensor without the need for pre-training on the sensor's data. The probability density function (PDF) learner, which is a part of the implemented system, consists of parallel connected Gaussian function circuits which realise the kernel density based statistics estimation. The second [20] is a fully analog edge detection circuit directly integrated to a photodiode, shown in Figure 41. The edge detection is performed on the analog output of the photodiode greatly reducing the power consumption and the need for data transfer. Here, the output of the active pixel sensor (APS) is directed to the Gaussian function circuit, which provides a high output current if the pixel is an edge and a low otherwise. In comparison with [19], this implementation [20] is tested on a fabricated chip with an area per pixel of  $225 \mu\text{m}^2$ .

**Table 9.** Smart Sensor Systems Summary.

	[19]	[20]
<b>Application</b>	Anomaly detection	Edge detection
<b>Type of sensor</b>	General	Photodiode
<b>Fully Analog</b>	NO	YES
<b>Type of Gaussian function</b>	Extra components	Current-mode circuits
<b>Power Consumption</b>	$75 \mu\text{W}$	$0.9 \mu\text{W}$ per pixel
<b>Simulation Level</b>	Schematic	Chip
<b>Area</b>	-	$225 \mu\text{m}^2$ per pixel



**Figure 40.** A mixed-mode anomaly detection system.

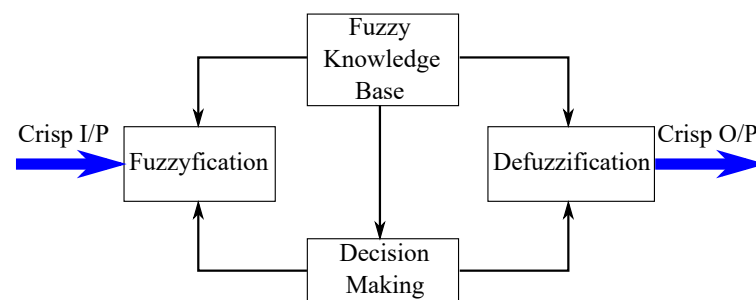


**Figure 41.** An edge detection circuit directly connected to a photodiode.

### 3.4. Fuzzy and Neuro-Fuzzy Systems

Fuzzy systems are based on fuzzy logic, which provides the theory of modeling real-world phenomena, which are inherently vague and ambiguous [112]. This theory provides all the tools (fuzzy techniques) for processing and mathematical representation. A neuro-fuzzy system is based on a fuzzy system, which is trained by a learning algorithm derived from NN theory [113,114]. Therefore a neuro-fuzzy system can be represented as a special multilayer feedforward NN and can be used as a universal approximator. Moreover, it can be interpreted as a system of fuzzy rules. It also uses fuzzy logic criteria for increasing the size of a NN. NNs are used to tune membership functions of fuzzy systems that are employed as decision-making systems for controlling equipment.

Except from software-based implementations, there are many realizations of membership functions, which are based on Gaussian function circuits. The fuzzy/neuro-fuzzy systems, which are realized based on these membership functions are categorized according to the application example. Most of them are mixed-mode implementations which took advantage of both analog and digital circuits. The existing categories include controllers [72,74,79,80,85,88], object recognition inference [82–84] or neural perception [31] engines or processors [22], function approximators [71,75] and a min–max network [21]. The presented controllers are hardware-friendly implementations based on classic fuzzy control theory, an example is shown in Figure 42. Specifically, References [74,79,80] design a Takagi–Sugeno based controller and [72] realize a Type-2 fuzzy controller. The object recognition applications [22,31,82–84] are based on neuro-fuzzy logic, an example is shown in Figure 43. In particular, a fuzzy system pre-processes the input data for the following perceptron (a single NN layer with an activation function). Both function approximators [71,75] combine different membership functions to produce more complex nonlinear functions. The min–max network [21] uses the Gaussian function circuits in the fuzzification block prior to the min–max operators.



**Figure 42.** A generic Fuzzy controller block diagram.

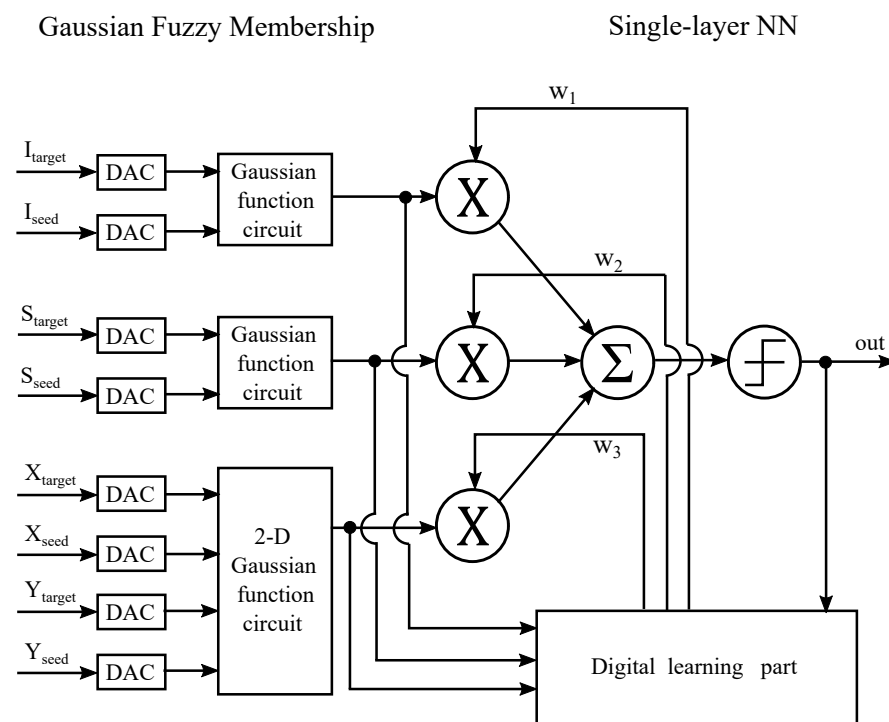


Figure 43. An example of a shallow Neuro-Fuzzy Radial Basis Function Neural Network.

In Table 10 we summarize the category, the complexity, the simulation level and the area of each implementation. The fuzzy rules complexity ranges from 4 to 50. A system with 50 rules is considered a high complexity system [113,114]. Almost all of the designs are tested on fabricated chips, except from [21] and the chip area varies from 0.08 mm<sup>2</sup> to 50 mm<sup>2</sup>.

Table 10. Neuro-fuzzy Systems Summary.

	Application	Complexity (Fuzzy Rules)	Simulation Level	Area
[21]	Min-Max Network	-	Schematic	-
[22]	Processor	50	Chip	13.5 mm <sup>2</sup>
[31]	Neural Perception Engine	-	Chip	49 mm <sup>2</sup>
[71]	Function Approximator	15	Chip	32 mm <sup>2</sup>
[72]	Controller	9	Chip	0.32 mm <sup>2</sup>
[74]	Controller	4	Chip	-
[75]	Function Approximator	25	Chip	-
[79]	Controller	25	Chip	0.08 mm <sup>2</sup>
[80]	Controller	-	Chip	-
[82]	Inference Engine	8	Chip	0.765 mm <sup>2</sup>
[83]	Inference Engine	-	Chip	50 mm <sup>2</sup>
[84]	Inference Engine	27	Chip	50 mm <sup>2</sup>
[85]	Controller	16	Chip	0.1 mm <sup>2</sup>
[88]	Controller	13	Chip	16.2 mm <sup>2</sup>

#### 4. Summary and Discussion

Throughout the years, there have been many different analog implementations of the Gaussian function, using various design techniques. These circuits are implemented targeting specific characteristics, for example low power consumption, area efficiency, high computation speed, better tunability or increased similarity with the theoretical response. Unfortunately, reliable documentation can be provided only for the power consumption, the number of transistors and the minimum operational bias current. The rest are not given by most of the research teams. Based on Section 2, a summary containing five architectures for each characteristic is presented in Tables 11–13.

**Table 11.** Gaussian Function Circuits performance summary and comparison. \*Additional current sources.

Ref.	Category	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[5]	Bulk-controlled	3.9 nW	0.6 V	1 nA	sub-threshold	11
[27]	Bulk-controlled	6 nW	0.6 V	5 nA	sub-threshold	10
[32]	Bulk-controlled	3.3 nW	0.6 V	3 nA	sub-threshold	14
[55]	Bulk-controlled	4 nW	0.6 V	3 nA	sub-threshold	10
[95]	Extra components	18.9 nW	3 V	1 nA	sub-threshold	* 14

**Table 12.** Gaussian Function Circuits performance summary and comparison. \* Power consumption for the entire System.

Ref.	Category	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[9]	Differential pair	-	1.3 V	-	above and sub-threshold	4
[20]	Other implementations	* 0.9 $\mu$ W per pixel	1.8 V	-	above threshold	8
[26]	Floating gate	214 $\mu$ W	3.3 V	-	above or sub threshold	5
[64]	Floating gate	-	-	5 $\mu$ A	sub-threshold	5
[100]	Other implementations	-	5 V	1 $\mu$ A	above threshold	5

**Table 13.** Gaussian Function Circuits performance summary and comparison. \*Additional current sources.

Ref.	Category	Power Consumption	Power Supply	Minimum $I_{bias}$	Operation Region	No of Transistors
[5]	Bulk-controlled	3.9 nW	0.6 V	1 nA	sub-threshold	11
[6]	Bulk-controlled	-	5 V	2 nA	sub-threshold	10
[55]	Bulk-controlled	4 nW	0.6 V	3 nA	sub-threshold	10
[95]	Extra components	18.9 nW	3 V	1 nA	sub-threshold	* 14
[97]	Extra components	-	5 V	2.6 nA	above or sub-threshold	* 9

Table 11 includes the implementations with the lowest power consumption. The power consumption ranges from 3.3 nW to 6 nW using bulk-controlled transistors [5,27,32,55], except from [95], which adds a CMFB circuit, which consumes 18.9 nW. They are all compact implementations, consisting of 10 to 14 transistors with a power supply at only 0.6 V [5,27,32,55] (except [95]) and their transistors operate in the sub-threshold region. All five designs are also fully electronically tunable.

Table 12 includes the implementations with the smallest number of transistors. The number of transistors ranges from 4 to 8 but without tunability in the Gaussian curve's characteristics, except from [64]. These compact implementations are designed using floating gate transistors [26,64], differential pairs [9] or other designs techniques [20,100] with a power supply ranging from 1.3 V to 5 V. The transistors of [20,100] operate in the above threshold region, while [64] operates in the subthreshold region. The design of [26] can operate in either the sub or the above threshold region and [9] has transistors operating in both regions.

Table 13 includes the implementations with the smallest minimum operational bias current. These bias currents vary from 1 nA to 3 nA while being fully electronically tunable. Moreover, small bias currents are directly related to the operation of transistors in the sub-threshold region. The number of transistors is relatively low (9 to 14). However, in the case of [95,97], the schematic is simplified by replacing current mirrors with current sources.

Architectures based on the translinear principle have many advantages and drawbacks. Translinear circuits have high frequency operation, high parameters' tunability, low supply voltage, lower-power consumption, low noise, low third order intermodulation distortion, low total harmonic distortion, the immunity to body effects, extended dynamic range, compactness, design modularity, and low circuit complexity. Despite the fact that translinear circuits have efficient realization of many analog nonlinear signal processing functions with small quantities of MOS transistors, in the case of the Gaussian function curve's implementation three separate components are required (absoluter, squarer and exponentiator circuits). Therefore, the number of transistors is higher compared to other architectures. The trade-off lies between the accuracy in the realization of the Gaussian function curve and the architecture's complexity. Circuits based on the translinear principles offer higher quality Gaussian function curves (compared to the theoretical Gaussian function) since they implement the exact mathematical equations, at the expense of using extra transistors.

Bulk-controlled designs reduce the need for extra transistors by using the fourth-terminal (Bulk terminal) to offer the desired tunability. The bulk-controlled transistor also deals with voltage threshold limitations and the whole topology is biased easier with lower power supply (based on sub-threshold region techniques). An additional advantage of connecting a parameter voltage to the bulk-terminal of differential pairs is that the bulks are no longer connected to the power supply rails, thus reducing any possible supply noise. Consequently, this circuit has better power supply rejection ratio. Possible drawbacks include higher leakage currents, lower computation speed, approximate behavior of the Gaussian function and necessity for triple-nwell technology.

Circuits built with floating-gate transistors use an extra terminal just like bulk-controlled architectures. This terminal, except from the desire tunability, also provides a non-volatile data storage capability. As a result, these implementations are relatively compact. However, FGMOS implementations require a high power supply voltage, which leads to higher power consumption. Moreover, the incorporation of FGMOS presents challenges in the aspect IC fabrication.

Gaussian function circuits based on differential pairs are compact and have design modularity. They operate at low speeds, limited parameters' tunability, high supply voltage and high-power consumption. These designs are used as a simple solution to realize a Gaussian function curve. Architectures using extra components achieve higher tunability at the cost of higher complexity (area) and power consumption.

Despite the numerous works in the literature about the implementations and the applications of the Gaussian function circuits, analog realizations have not yet been established in commercial or real-world applications compared to digital or software-based ones. Therefore, to further motivate new researchers to implement new analog realizations, development should be focused on the advantages of analog-hardware implementations. New computing paradigms should lead to a new domain of smart industry based on low-power consumption, area efficiency, high computation speed and parallelization. This way, analog accelerators should gain popularity and create a stable dipole between them and digital ones. In this case, hardware (both analog and digital) implementations will gain a new role in the artificial intelligence domain and new demanding applications will be developed in the future.

## 5. Conclusions

This paper has provided a review of Gaussian function circuits' architectures, operating principles and applications. Furthermore, a number of the commonly used design

architectures for current correlators, differential blocks, current-mode circuits, analog computational circuits and sub-threshold region methods have been discussed in detail with possible tradeoffs. In the context of current applications, state-of-the-art high-level implementations have been subsequently described to illustrate challenges in their realization together with different approaches and techniques. Collecting and providing all the referred architectures and applications, it is necessary to upgrade these implementations and design new, high-speed, ultra-low power, area efficient and accurate Gaussian function circuits, which can be used as building blocks in different wearable or portable applications.

**Author Contributions:** Investigation, V.A., M.G., G.G. and C.D.; Writing—original draft, V.A., M.G., G.G. and C.D.; Writing—review and editing, V.A., M.G., G.G., C.D. and P.P.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

ADC	Analog to Digital Converter
APS	Active Pixel Sensor
CCII	Current-controlled Current-conveyor Second Generation
CMFB	Common Mode Feedback
DAC	Digital to Analog Converter
DML	Deep Machine Learning
EMG	Electromyography
FGMOS	Floating-gate MOSFET
GRBF	Gaussian Radial Basis Function
HDL	Hardware Description Language
I-V	Current to Voltage
LPF	Low Pass Filter
LVQ	Learning Vector Quantizer
ML	Machine Learning
NN	Neural Network
OTA	Operational transconductance Amplifier
PDF	Probability Density Function
PVT	Process, Variation, Temperature
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
SNN	Spiking Neural Network
SOM	Self Organized MAP
SVDD	Support Vector Domain Description
SVM	Support Vector Machine
SVR	Support Vector Regression
VGA	Variable Gain Amplifier
VW	Variable Width

### References

1. Delbruck, T. "Bump" circuits for computing similarity and dissimilarity of analog voltage. In proceedings of the IEEE International Joint Conference Neural Networks, Seattle, WA, USA, 8–12 July 1991.
2. Delbruck, T.; Mead, C. Bump circuits. In Proceedings of International Joint Conference on Neural Networks, Nagoya, Japan, 25–29 October 1993.
3. Melendez-Rodriguez, M.; Silva-Martínez, J. A fully-programmable temperature-compensated analogue circuit for gaussian functions. In Proceedings of the Third International Workshop on Design of Mixed-Mode Integrated Circuits and Applications, Puerto Vallarta, Mexico, 28 July 1999.
4. Zhang, R.; Uetake, N.; Nakada, T.; Nakashima, Y. Design of programmable analog calculation unit by implementing support vector regression for approximate computing. *IEEE Micro* **2018**, *38*, 73–82. [[CrossRef](#)]



5. Gourdouparis, M.; Alimisis, V.; Dimas, C.; Sotiriadis, P.P. An ultra-low power,  $\pm 0.3$  V supply, fully-tunable Gaussian function circuit architecture for radial-basis functions analog hardware implementation. *Aeu-Int. J. Electron. Commun.* **2021**, *136*, 153755. [[CrossRef](#)]
6. Minch, B.A. A simple variable-width CMOS bump circuit. In Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 22–25 May 2016.
7. Srivastava, R.; Gupta, M.; Singh, U. FGMOS transistor based low voltage and low power fully programmable Gaussian function generator. *Analog. Integr. Circuits Signal Process.* **2014**, *78*, 245–252. [[CrossRef](#)]
8. Galbraith, J.; Holman, W.T. An analog Gaussian generator circuit for radial basis function neural networks with non-volatile center storage. In Proceedings of the 39th Midwest Symposium on Circuits and Systems, Ames, IA, USA, 21–21 August 1996.
9. Vrtaric, D.; Ceperic, V.; Baric, A. Area-efficient differential Gaussian circuit for dedicated hardware implementations of Gaussian function based machine learning algorithms. *Neurocomputing* **2013**, *118*, 329–333. [[CrossRef](#)]
10. Churcher, S.; Murray, A.F.; Reeckie, H.M. Programmable analogue VLSI for radial basis function networks. *Electron. Lett.* **1993**, *29*, 18, 1603–1605. [[CrossRef](#)]
11. Mohamed, A.R.; Qi, L.; Li, Y.; Wang, G. A generic nano-watt power fully tunable 1-d Gaussian kernel circuit for artificial neural network. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *67*, 9, 1529–1533. [[CrossRef](#)]
12. Anderson, J.; Platt, J.C.; Kirk, D.B. An analog VLSI chip for radial basis functions. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann: San Mateo, CA, USA, 1993, 765–772.
13. Azadmehr, M.; Marchetti, L.; Berg, Y. An Analog Voltage Similarity Circuit with a Bell-Shaped Power Consumption. *Electronics* **2021**, *10*, 1141. [[CrossRef](#)]
14. Watkins, S.S.; Chau, P.M.; Tawel, R. A radial basis function neurocomputer implemented with analog VLSI circuits. In Proceedings of the 1992IJCNN International Joint Conference on Neural Networks, Baltimore, MD, USA, 7–11 June 1992.
15. Lee, K.; Park, J.; Yoo, H.J. A low-power, mixed-mode neural network classifier for robust scene classification. *J. Semicond. Technol. Sci.* **2019**, *19*, 129–136. [[CrossRef](#)]
16. Verleysen, M.; Thissen, P.; Voz, J.L.; Madrenas, J. An analog processor architecture for a neural network classifier. *IEEE Micro* **1994**, *14*, 16–28. [[CrossRef](#)]
17. Payvand, M.; Indiveri, G. Spike-based plasticity circuits for always-on on-line learning in neuromorphic systems. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019.
18. Payvand, M.; Fouda, M.E.; Kurdahi, F.; Eltawil, A.; Neftci, E.O. Error-triggered three-factor learning dynamics for crossbar arrays. In Proceedings of the 2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Genova, Italy, 31 August–2 September 2020.
19. Shylendra, A.; Shukla, P.; Mukhopadhyay, S.; Bhunia, S.; Trivedi, A.R. Low power unsupervised anomaly detection by nonparametric modeling of sensor statistics. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2020**, *28*, 1833–1843. [[CrossRef](#)]
20. Nam, M.; Cho, K. Implementation of real-time image edge detector based on a bump circuit and active pixels in a CMOS image sensor. *Integration* **2018**, *60*, 56–62. [[CrossRef](#)]
21. Ota, Y.; Wilamowski, B.M. Current-mode CMOS implementation of a fuzzy min-max network. *World Congr. Neural Netw.* **1995**, *2*, 480–483.
22. Oh, J.; Kim, G.; Nam, B.G.; Yoo, H.J. A 57 mW 12.5  $\mu$ J/Epoch embedded mixed-mode neuro-fuzzy processor for mobile real-time object recognition. *IEEE J. Solid-State Circuits* **2013**, *48*, 2894–2907. [[CrossRef](#)]
23. Khaneshan, T.M.; Nematzadeh, M.; Khoei, A.; Hadidi, K. An analog reconfigurable Gaussian-shaped membership function generator using current-mode techniques. In Proceedings of the 20th Iranian Conference on Electrical Engineering (ICEE2012), Tehran, Iran, 15–17 May 2012.
24. Dorzhigulov, A.; James, A.P. Generalized bell-shaped membership function generation circuit for memristive neural networks. In Proceedings of the 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 26–29 May 2019.
25. Li, F.; Chang, C.H.; Siek, L. A very low power 0.7 V subthreshold fully programmable Gaussian function generator. In Proceedings of the 2010 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics (PrimeAsia), Shanghai, China, 22–24 September 2010.
26. Srivastava, R.; Singh, U.; Gupta, M. Analog circuits for Gaussian function with improved performance. In Proceedings of the 2011 World Congress on Information and Communication Technologies, Mumbai, India, 11–14 December 2011.
27. Alimisis, V.; Gourdouparis, M.; Dimas, C.; Sotiriadis, P.P. Ultra-Low Power, Low-Voltage, Fully-Tunable, Bulk-Controlled Bump Circuit. In Proceedings of the 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCASST), Thessaloniki, Greece, 5–7 July 2021.
28. Kang, K.; Shibata, T. An on-chip-trainable Gaussian-kernel analog support vector machine. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2009**, *57*, 1513–1524. [[CrossRef](#)]
29. Abuelma'Ati, M.T.; Shwehneh, A. A reconfigurable gaussian/triangular basis functions computation circuit. *Analog. Integr. Circuits Signal Process.* **2006**, *47*, 53–64. [[CrossRef](#)]
30. Azadmehr, M.; Marchetti, L.; Berg, Y. A low power analog voltage similarity circuit. In Proceedings of the 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, 28–31 May 2017.
31. Kim, J.Y.; Kim, M.; Lee, S.; Oh, J.; Kim, K.; Yoo, H.J. A 201.4 GOPS 496 mW real-time multi-object recognition processor with bio-inspired neural perception engine. *IEEE J. Solid-State Circuits* **2009**, *45*, 32–45. [[CrossRef](#)]

32. Alimisis, V.; Gourdouparis, M.; Dimas, C.; Sotiriadis, P.P. A 0.6 V, 3.3 nW, Adjustable Gaussian Circuit for Tunable Kernel Functions. In Proceedings of the 2021 34th SBC/SBMicro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI), Campinas, Brazil, 23–27 August 2021.
33. Popa, C. Low-voltage improved accuracy Gaussian function generator with fourth-order approximation. *Microelectron. J.* **2012**, *43*, 515–520. [[CrossRef](#)]
34. Gilbert, B. Translinear circuits: A proposed classification. *Electron. Lett.* **1975**, *11*, 14–16. [[CrossRef](#)]
35. Mulder, J.; Serdijn, W.A.; van der Woerd, A.C.; van Roermund, A. *Dynamic Translinear and Log-Domain Circuits*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999.
36. Wiegierink, R.J. *Analysis and Synthesis of MOS Translinear Circuits*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1993; Volume 246.
37. Masmoudi, D.S.; Dieng, A.T.; Masmoudi, M. A subthreshold mode programmable implementation of the Gaussian function for RBF neural networks applications. In Proceedings of the IEEE International Symposium on Intelligent Control, Vancouver, BC, Canada, 30–30 October 2002.
38. Li, F.; Chang, C.H.; Basu, A.; Siek, L. A 0.7 V low-power fully programmable Gaussian function generator for brain-inspired Gaussian correlation associative memory. *Neurocomputing* **2014**, *138*, 69–77. [[CrossRef](#)]
39. Li, F.; Chang, C.H.; Siek, L. A compact current mode neuron circuit with Gaussian taper learning capability. In Proceedings of the 2009 IEEE International Symposium on Circuits and Systems, Taipei, Taiwan, 24–27 May 2009; pp. 2129–2132.
40. Moshfe, S.; Khoei, A.; Hadidi, K.; Mashoufi, B. A fully programmable nano-watt analogue CMOS circuit for Gaussian functions. In Proceedings of the 2010 International conference on electronic devices, systems and applications, Kuala Lumpur, Malaysia, 11–14 April 2010.
41. Zhang, R.; Shibata, T. Fully Parallel Self-Learning Analog Support Vector Machine Employing Compact Gaussian Generation Circuits. *Jpn. J. Appl. Phys.* **2012**, *51*, 4–10. [[CrossRef](#)]
42. Zhang, R.; Shibata, T. A VLSI hardware implementation study of SVDD algorithm using analog Gaussian-cell array for on-chip learning. In Proceedings of the 2012 13th International Workshop on Cellular Nanoscale Networks and their Applications, Turin, Italy, 29–31 August 2012.
43. Carlos, S.L.; Alejo, D.S.; Esteban, T.C. Generating Gaussian functions using low-voltage MOS-translinear circuits. In Proceedings of the WSEAS International Conference on Instrumentation, Measurement, Control, Circuits and Systems, Cancun, Mexico, 12–16 May 2002.
44. Saatlo, A.N.; Ozoguz, S. On the realization of Gaussian membership function circuit operating in saturation region. In Proceedings of the 2015 38th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 9–11 July 2015.
45. Lu, J.; Young, S.; Arel, I.; Holleman, J. A 1 TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13  $\mu\text{m}$  CMOS. *IEEE J. Solid-State Circuits* **2014**, *50*, 270–281. [[CrossRef](#)]
46. Thompson, S. MOS scaling: Transistor challenges for the 21st century. *Intel Technol. J.* **1998**, *Q3*, 1–19.
47. Khateb, F.; Bielek, D.; Khatib, N.; Vávra, J. Utilizing the bulk-driven technique in analog circuit design. In Proceedings of the 13th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, Vienna, Austria, 14–16 April 2010.
48. Blalock, B.J.; Allen, P.E. A low-voltage, bulk-driven MOSFET current mirror for CMOS technology. In Proceedings of ISCAS'95-International Symposium on Circuits and Systems, Seattle, WA, USA, 30 April–3 May 1995.
49. Khateb, F.; Vlassis, S. Low-voltage bulk-driven rectifier for biomedical applications. *Microelectron. J.* **2013**, *44*, 642–648. [[CrossRef](#)]
50. He, R.; Zhang, L. Evaluation of modern MOSFET models for bulk-driven applications. In Proceedings of the 2008 51st Midwest Symposium on Circuits and Systems, Knoxville, TN, USA, 10–13 August 2008.
51. Omura, Y.; Mallik, A.; Matsuo, N. *MOS Devices for Low-Voltage and Low-Energy Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2017.
52. Payvand, M.; Muller, L.K.; Indiveri, G. Event-based circuits for controlling stochastic learning with memristive devices in neuromorphic architectures. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018.
53. Payvand, M.; Nair, M.V.; Müller, L.K.; Indiveri, G. A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation. *Faraday Discuss.* **2019**, *213*, 487–510. [[CrossRef](#)]
54. Donati, E.; Payvand, M.; Risi, N.; Krause, R.; Indiveri, G. Discrimination of EMG signals using a neuromorphic implementation of a spiking neural network. *IEEE Trans. Biomed. Circuits Syst.* **2019**, *13*, 795–803. [[CrossRef](#)]
55. Gourdouparis, M.; Alimisis, V.; Dimas, C.; Sotiriadis, P.P. Ultra-Low Power (4nW), 0.6 V Fully-Tunable Bump Circuit operating in Sub-threshold regime. In Proceedings of the 2021 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems (DTS), Sfax, Tunisia, 7–10 June 2021.
56. Alimisis, V.; Dimas, C.; GPappas, S.; Sotiriadis, P.P. Analog Realization of Fractional-Order Skin-Electrode Model for Tetrapolar Bio-Impedance Measurements. *Technologies* **2020**, *8*, 61. [[CrossRef](#)]
57. Liu, S.C.; Kramer, J.; Indiveri, G.; Delbrück, T.; Douglas, R. *Analog VLSI: Circuits and Principles*; MIT Press: Cambridge, MA, USA, 2002.
58. Ozalevli, E. *Tunable and Reconfigurable Circuits Using Floating-Gate Transistors: Programming and Tuning, Design Method, Applications*; VDM Verlag Dr. Müller: Riga, Latvia, 2009.

59. Sudhanshu, S.J.; Susheel, S.; Mangotra, L.K. *Analog Applications of Floating-Gate MOS Transistor: Floating-Gate MOS Transistor Analog Applications in Current Mirrors and Current Conveyors*; LAP LAMBERT Academic Publishing: Saarbrücken, Germany, 2010.
60. Ozalevli, E. *Floating-Gate Transistors in Analog and Mixed-Signal Circuit Design: Programming, Design Methodology, and Applications*; VDM Verlag Dr. Müller: Riga, Latvia, 2009.
61. Peng, S.Y.; Hasler, P.E.; Anderson, D.V. An analog programmable multidimensional radial basis function based classifier. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2007**, *54*, 2148–2158. [[CrossRef](#)]
62. Peng, S.Y.; Minch, B.A.; Hasler, P. Analog VLSI implementation of support vector machine learning and classification. In Proceedings of the 2008 IEEE International Symposium on Circuits and Systems, Seattle, WA, USA, 18–21 May 2008.
63. Peng, S.Y.; Minch, B.A.; Hasler, P. A programmable floating-gate bump circuit with variable width. In Proceedings of the 2005 IEEE International Symposium on Circuits and Systems, Kobe, Japan, 23–26 May 2005.
64. Hasler, P.; Smith, P.; Duffy, C.; Gordon, C.; Dugger, J.; Anderson, D. A floating-gate vector-quantizer. In Proceedings of the 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002., Tulsa, OK, USA, 4–7 August 2002.
65. Theogarajan, L.; Akers, L.A. A multi-dimensional analog Gaussian radial basis circuit. In Proceedings of the 1996 IEEE International Symposium on Circuits and Systems. Circuits and Systems Connecting the World. ISCAS 96, Atlanta, GA, USA, 15 May 1996.
66. Yamasaki, T.; Shibata, T. Analog soft-pattern-matching classifier using floating-gate MOS technology. *IEEE Trans. Neural Networks* **2003**, *14*, 1257–1265. [[CrossRef](#)]
67. Yamasaki, T.; Shibata, T. An analog similarity evaluation circuit featuring variable functional forms. In Proceedings of the ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196), Sydney, NSW, Australia, 6–9 May 2001.
68. Yamasaki, T.; Yamamoto, K.; Shibata, T. Analog pattern classifier with flexible matching circuitry based on principal-axis-projection vector representation. In Proceedings of the 27th European Solid-State Circuits Conference, Villach, Austria, 18–20 September 2001.
69. Srivastava, R.; Gupta, M.; Singh, U. Fully programmable Gaussian function generator using floating gate MOS transistor. *ISRN Electron.* **2012**, *2012*, 1–5. [[CrossRef](#)]
70. Gilbert, B. A precise four-quadrant multiplier with subnanosecond response. *IEEE J. Solid-State Circuits* **1968**, *3*, 365–373. [[CrossRef](#)]
71. Franchi, E.; Manaresi, N.; Rovatti, R.; Bellini, A.; Baccarani, G. Analog synthesis of nonlinear functions based on fuzzy logic. *IEEE J. Solid-State Circuits* **1998**, *33*, 885–895. [[CrossRef](#)]
72. Khosla, M.; Sarin, R.K.; Uddin, M. Design of an analog CMOS based interval type-2 fuzzy logic controller chip. *Int. J. Artif. Intell. Expert Syst.* **1995**, *2*, 167–183.
73. Khosla, M.; Sarin, R.K.; Uddin, M.; Sharma, A. Analog realization of fuzzifier for IT2 fuzzy processor. In Proceedings of the 2011 3rd International Conference on Electronics Computer Technology, Kanyakumari, India, 8–10 April 2011.
74. Vidal-Verdu, F.; Delgado-Restituto, M.; Navas, R.; Rodriguez-Vazquez, A. A design approach for analog neuro/fuzzy systems in CMOS digital technologies. *Comput. Electr. Eng.* **25**, 309–337. [[CrossRef](#)]
75. Wang, W.Z.; Jin, D.M. Neuro-fuzzy system with high-speed low-power analog blocks. *Fuzzy Sets Syst.* **2006**, *157*, 2974–2982. [[CrossRef](#)]
76. Lucks, M.B.; Oki, N. Radial Basis Function circuits using folded cascode differential pairs. In Proceedings of the 2010 53rd IEEE International Midwest Symposium on Circuits and Systems, Seattle, WA, USA, 1–4 August 2010.
77. Choi, J.; Sheu, B.J.; Chang, J.F. A Gaussian synapse circuit for analog VLSI neural networks. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **1994**, *2*, 129–133. [[CrossRef](#)]
78. Lau, K.T.; Lee, S.T. A programmable CMOS gaussian synapse for analogue VLSI neural networks. *Int. J. Electron.* **1997**, *83*, 91–98. [[CrossRef](#)]
79. Dinavari, V.F.; Khoei, A.; Hadidi, K.H.; Soleimani, M.; Mojarad, H. Design of a current-mode analog CMOS fuzzy logic controller. In Proceedings of the IEEE EUROCON 2009, St. Petersburg, Russia, 18–23 May 2009.
80. Wilamowski, B.M.; Jaeger, R.C.; Kaynak, M.O. Neuro-fuzzy architecture for CMOS implementation. *IEEE Trans. Ind. Electron.* **1999**, *46*, 1132–1136. [[CrossRef](#)]
81. Lee, K.; Park, J.; Kim, G.; Hong, I.; Yoo, H.J. A multi-modal and tunable Radial-Basis-Function circuit with supply and temperature compensation. In Proceedings of the 2013 IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, 19–23 May 2013.
82. Oh, J.; Lee, S.; Yoo, H.J. 1.2-mw online learning mixed-mode intelligent inference engine for low-power real-time object recognition processor. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2012**, *21*, 921–933. [[CrossRef](#)]
83. Lee, S.; Oh, J.; Park, J.; Kwon, J.; Kim, M.; Yoo, H.J. A 345 mW heterogeneous many-core processor with an intelligent inference engine for robust object recognition. *IEEE J. Solid-State Circuits* **2010**, *46*, 42–51.
84. Oh, J.; Lee, S.; Kim, M.; Kwon, J.; Park, J.; Kim, J.Y.; Yoo, H.J. A 1.2 mW on-line learning mixed mode intelligent inference engine for robust object recognition. In Proceedings of the 2010 Symposium on VLSI Circuits, Honolulu, HI, USA, 16–18 June 2010.
85. Yosefi, G.; Khoei, A.; Hadidi, K. Design of a new CMOS controllable mixed-signal current mode fuzzy logic controller (FLC) chip. In Proceedings of the 2007 14th IEEE International Conference on Electronics, Circuits and Systems, Marrakech, Morocco, 11–14 December 2007.

86. Farag, E.N.; Elmasry, M.I. *Mixed Signal VLSI Wireless Design: Circuits and Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2000.
87. Cevikhas, I.C.; Ogrenci, A.S.; Dundar, G.; Balkur, S. VLSI implementation of GRBF (Gaussian radial basis function) networks. In Proceedings of the 2000 IEEE International Symposium on Circuits and Systems (ISCAS), Geneva, Switzerland, 28–31 May 2000.
88. Guo, S.; Peters, L.; Surmann, H. Design and application of an analog fuzzy logic controller. *IEEE Trans. Fuzzy Syst.* **1996**, *4*, 429–438. [[CrossRef](#)]
89. Pour, M.E.; Mashoufi, B. A low power consumption and compact mixed-signal Gaussian membership function circuit for neural/fuzzy hardware. In Proceedings of the 2011 International Conference on Electronic Devices, Systems and Applications (ICEDSA), Kuala Lumpur, Malaysia, 25–27 April 2011.
90. Abuelma'atti, M.T.; Al-Abbas, S.R. A new analog implementation for the Gaussian function. In Proceedings of the 2016 IEEE Industrial Electronics and Applications Conference (IEACon), Kota Kinabalu, Malaysia, 20–22 November 2016.
91. Bragg, J.A.; Brown, E.A.; DeWeerth, S.P. A tunable voltage correlator. *Analog. Integr. Circuits Signal Process.* **2004**, *39*, 89–94. [[CrossRef](#)]
92. Lucks, M.B.; Oki, N. A radial basis function network (RBFN) for function approximation. In Proceedings of the 42nd Midwest Symposium on Circuits and Systems (Cat. No. 99CH36356), Las Cruces, NM, USA, 8–11 August 1999.
93. De Oliveira, J.P.; Oki, N. A Programmable Analog Gaussian Function Synthesizer. In Proceedings of SBMicro, Porto Alegre, Brazil, 2002.
94. De Oliveira, J.P.; Oki, N. An analog implementation of radial basis neural networks (RBNN) using BiCMOS technology. In Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems. MWSCAS 2001 (Cat. No. 01CH37257), Dayton, OH, USA, 14–17 August 2001.
95. Lu, J.; Yang, T.; Jahan, M.S.; Holleman, J. Nano-power tunable bump circuit using wide-input-range pseudo-differential transistor. *Electron. Lett.* **2014**, *50*, 921–923. [[CrossRef](#)]
96. Azimi, S.M.; Miar-Naimi, H. Designing programmable current-mode Gaussian and bell-shaped membership function. *Analog. Integr. Circuits Signal Process.* **2020**, *102*, 323–330. [[CrossRef](#)]
97. Lin, S.Y.; Huang, R.J.; Chiueh, T.D. A tunable Gaussian/square function computation circuit for analog neural networks. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process.* **1998**, *45*, 441–446. [[CrossRef](#)]
98. Theogarajan, L.; Akers, L.A. A scalable low voltage analog Gaussian radial basis circuit. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process.* **1997**, *44*, 977–979. [[CrossRef](#)]
99. Cauwenberghs, G.; Pedroni, V. A charge-based CMOS parallel analog vector quantizer. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 1995; pp. 779–786.
100. Madrenas, J.; Verleysen, M.; Thissen, P.; Voz, J.L. A CMOS analog circuit for Gaussian functions. *IEEE Trans. Circuits Syst. II Analog. Digit. Signal Process.* **1996**, *43*, 70–74. [[CrossRef](#)]
101. Collins, S.; Marshall, G.F.; Brown, D.R. An analogue Radial Basis Function circuit using a compact Euclidean Distance calculator. In Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94, London, UK, 30 May–2 June 1994.
102. Chi, P.; Li, S.; Xu, C.; Zhang, T.; Zhao, J.; Liu, Y.; Wang, Y.; Xie, Y. Prime: A novel processing-in-memory architecture for neural network computation in rram-based main memory. *ACM SIGARCH Comput. Archit. News* **2016**, *44*, 27–39. [[CrossRef](#)]
103. Shawahna, A.; Sait, S.M.; El-Maleh, A. FPGA-based accelerators of deep learning networks for learning and classification: A review. *IEEE Access* **2018**, *7*, 7823–7859. [[CrossRef](#)]
104. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006.
105. Mohri, M.; Rostamizadeh, A.; Talwalkar, A. *Foundations of Machine Learning*; MIT Press: Cambridge, MA, USA, 2018.
106. Jabri, M.; Coggins, R.J.; Flower, B.G. *Adaptive Analog VLSI Neural Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012.
107. Liu, S.C.; Delbruck, T.; Indiveri, G.; Whatley, A.; Douglas, R. (Ed.) *Event-Based Neuromorphic Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
108. Lande, T.S. (Ed.) *Neuromorphic Systems Engineering: Neural Networks in Silicon*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1998; Volume 447.
109. De Silva, C.W. *Sensor Systems: Fundamentals and Applications*; CRC Press: Boca Raton, FL, USA, 2016.
110. Meijer, G.; Makinwa, K.; Pertijs, M. (Eds.) *Smart Sensor Systems: Emerging Technologies and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2014.
111. Lin, Y.L.; Kyung, C.M.; Yasuura, H.; Liu, Y. (Eds.) *Smart Sensors and Systems*; Springer International Publishing: Cham, Switzerland, 2015.
112. Chen, G.; Pham, T.T. *Introduction to Fuzzy Systems*; CRC Press: Boca Raton, FL, USA, 2005.
113. Fullér, R. *Introduction to Neuro-Fuzzy Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2000; Volume 2.
114. Nauck, D.; Klawonn, F.; Kruse, R. *Foundations of Neuro-Fuzzy Systems*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 1997.