An Event-Based, Digital Time Difference Encoder Model Implementation for Neuromorphic Systems

Daniel Gutierrez-Galan, *Member, IEEE*, Thorben Schoepe, *Member, IEEE*, Juan P. Dominguez-Morales, Angel Jimenez-Fernandez, Elisabetta Chicca, *Senior Member, IEEE*, and Alejandro Linares-Barranco, *Senior Member, IEEE*

Abstract—Neuromorphic systems are a viable alternative to conventional systems for real-time tasks with constrained resources. Their low power consumption, compact hardware realization, and low-latency response characteristics are the key ingredients of such systems. Furthermore, the event-based signal processing approach can be exploited for reducing the computational load and avoiding data loss, thanks to its inherently sparse representation of sensed data and adaptive sampling time. In event-based systems, the information is commonly coded by the number of spikes within a specific temporal window. However, event-based signals may contain temporal information which is complex to extract when using rate coding. In this work, we present a novel digital implementation of the model, called Time Difference Encoder, for temporal encoding on event-based signals, which translates the time difference between two consecutive input events into a burst of output events. The number of output events along with the time between them encodes the temporal information. The proposed model has been implemented as a digital circuit with a configurable time constant, allowing it to be used in a wide range of sensing tasks which require the encoding of the time difference between events, such as optical flow based obstacle avoidance, sound source localization and gas source localization. This proposed bio-inspired model offers an alternative to the Jeffress model for the Interaural Time Difference estimation, validated with a sound source lateralization proof-of-concept. The model has been simulated and implemented on an FPGA, requiring 122 slice registers of hardware resources and less than 1 mW of power consumption.

Index Terms—Digital design, event-based processing, neuromorphic systems, spiking neuron, time difference encoder

I. INTRODUCTION

Manuscript received October 31, 2020; revised October 31, 2020. This research was supported by the Spanish grant (with support from the European Regional Development Fund) COFNET (TEC2016-77785-P) and by the Cluster of Excellence Cognitive Interaction Technology CITEC (EXC 277) at Bielefeld University, which is funded by the German Research Foundation (DFG). The work of D. G.-G. was supported by a "Formación de Personal Investigador" Scholarship from the Spanish Ministry of Education, Culture and Sport. Part of this work was carried out during research internships of D. G.-G., J. P. D.-M. and A. L.-B. ("Salvador de Madariaga" program) in the Faculty of Technology (Bielefeld University). The authors would like to acknowledge the financial support of the CogniGron research center and the Ubbo Emmius Funds (Univ. of Groningen).

D. G.-G., J. P. D.-M., A. J.-F. and A. L.-B. are with the Robotics and Technology of Computers Lab, Universidad de Sevilla, Spain; and also with the SCORE Lab, IBUS, Universidad de Sevilla, Spain (email: dgutierrez@atc.us.es).

T. S. is with the Faculty of Technology and Cognitive Interaction Technology Center of Excellence, Bielefeld University, Germany

T. S. and E. C. are with the Bio-Inspired Circuits and Systems Lab, Zernike Institute for Advanced Materials, University of Groningen, Netherlands; and also with the Groningen Cognitive Systems and Materials Center, University of Groningen, Nijenborgh 4, NL-9747 AG Groningen, Netherlands.

MODERN computers are based on the von Neumann architecture. Over the years, developments in traditional computing technologies have been focused on increasing the speed of computation. Unfortunately, the achieved Central Processing Unit (CPU) speed is not matched by the speed of the memory access (leading to the so-called von Neumann bottleneck [1]) and the gap has been increasing exponentially over the years [2]. Furthermore, Moore's law might come to an end in the next years due to physical constraints such as increasing thermal noise [3]. These developments have triggered increasing research efforts for the design of alternative computational architectures which may complement and augment traditional von Neumann machines.

One interesting architecture is the human brain, which can compute complex correlations in real-time with approximately 100 billion neurons and more than 100 trillion synapses while consuming only 20 watts [4], [5], [6]. Given the ability of the brain and the fundamentally different substrate (asynchronous operation, co-localization of memory and computation, full parallelism, etc.), neural computation is a promising source of inspiration. The capability to take complex decisions in real time by means of limited sensory data poses the basis for the development of a new generation of edge computing devices [7]. To this end, researchers have aimed at modelling specific parts of the brain with Artificial Neural Networks (ANNs). For doing so, neural connectomics and neurophysiological data acquired from biology have been studied intensely to extract the most important characteristics of neural computation. These findings led to the creation of various artificial synapse and neuron models at different levels of abstraction. In this context, the spiking approach observed in the nervous system can offer considerable advantages in terms of latencies, power consumption and compactness provided by a digital implementation optimized for the target task [8]. Some spiking neuron models are the Leaky Integrate-And-Fire (LIF) and the Izhikevich models, which have already been implemented in Field Programmable Gate Arrays (FPGAs)[9][10], other digital circuits [11] [7] and also in analog circuits [12].

The communication between neurons is carried out by means of current pulses, so-called spikes, which is the dominant mode of information transference in the vertebrate nervous system. The nervous system is excited with external stimuli through the senses, as the vision and the hearing. However, that sensory information is in the analog domain, and needs to be encoded into the spike domain. For this task, two main coding schemes can be found in the literature. The

first one, called rate coding [13], [14], encodes the information in the number of spikes fired over a time period. This value is also known as firing rate, and can be used for simple signal classification [15]. The second one, where the information is encoded not only in the number of spikes but also in the time between two consecutive spikes, is called temporal coding [16]. This approach is being mostly used in reservoir computing systems [17] and in the encoding of fast visual stimuli through the latency to first spike (LFS) coding [18]. The temporal responses of different neuron populations to a dynamic input stimulus contains spatio-temporal information which is self-contained in their output spike trains. In addition, time difference, or signal temporal correlation between spike trains of consecutive neurons within a network, contains relative information about the nature of the stimulus that can be useful for complex post-processing tasks such as object tracking or autonomous navigation [19].

A recently proposed model which computes temporal dependencies in Spiking Neural Networks (SNNs) is the spiking Elementary Motion Detector (sEMD) proposed by Milde et al. [19]. In that work, the sEMD model consisted of two parts: an event-based vision sensor as input and the Time Difference Encoder (TDE) as sensory preprocessing unit. The TDE unit translates the time difference between two events into a burst of output spikes. Both the number of output spikes and the duration of the burst produced by the model directly reflect the temporal correlation of two input signals, and it is inversely proportional to the time difference. Milde et al. developed an analog Complementary Metal-Oxide-Semiconductor (CMOS) implementation of the TDE, characterized its performances on silicon and applied it to the encoding of Optical Flow (OF). The TDE model has already been used for processing visual [20], auditory and olfactory information. Its universal applicability has high potential for inspiring innovative pre-processing for SNNs, especially supporting close-loop neuromorphic systems with low latency requirements.

This wide range of possible applications presents a challenge in terms of time resolution and scalability. Time resolution in analog circuits is constrained by the size of the capacitors. Therefore, for high time constants applications, big capacitors would be needed. Furthermore, mismatch problems and parameter setting difficulties may appear due to the analog nature of the implementation. In this work, a generic, event-based, digital implementation of the Time Difference Encoder (TDE) model is presented. Its time resolution is configurable by means of a clock divider, covering a time range from nanoseconds to seconds. Moreover, the model can be deployed on FPGA-based platforms. This computational platform suits very well the integration of SNNs due to its highly-parallel, low-latency nature. This TDE implementation facilitates the development of complex and reconfigurable neuromorphic networks receiving input from event-based sensors such as bio-inspired retinas [21] and cochleas [22], [23], [24], [25]. Finally, the TDE's performance was evaluated in simulation by characterizing its response to synthetic input stimuli and also to real world recordings from a Neuromorphic Auditory Sensor (NAS).

The main contributions of this work include the following:

- The digital TDE model implementation as an alternative of the analog version for event-based, real-time neuromorphic applications with different time constants.
- The simulation and full characterization of the proposed model, verifying the basic cases and analyzing its response for complex input stimuli.
- The deployment into an FPGA-based board, thus having flexibility for further designs, with a power consumption of less than 1 mW for each TDE unit and allowing up to 400 units in basic FPGA chips.
- A proof-of-concept of a sound source lateralization task using the proposed model, where the events were received in real time from a neuromorphic auditory sensor, providing a new alternative to the state-of-the-art of the sound source localization systems.

The paper is organized as follows: section II details the operating principles of the TDE model based on the original model, as well as its design constraints. Section III shows the digital architecture of the model, and describes how it works. Then, section IV presents the full characterization results of the proposed model. Finally, in section V the results are compared with the original model and its usability is also discussed.

II. TDE MODEL OPERATING PRINCIPLE

The TDE model [19] translates the temporal difference between two input events into a short burst of output digital pulses. It comprises two inputs: the facilitatory pulse (faci) and the trigger synapse (trig), as well as one spiking output shown in Fig. 1 a). When an event arrives at the facilitatory input, an exponentially decaying facilitatory factor variable is generated, called gain. If an event enters the trigger synapse shortly after (i.e., small time difference Δt), as in Fig. 1 b), an Excitatory Post-Synaptic Current (EPSC) is produced. In this process, the amplitude of the EPSC depends proportionally on the facilitatory variable value i.e. on the gain factor. Hence, the EPSC's amplitude decreases with increasing time difference. The trigger synapse projects onto a LIF neuron which integrates the postsynaptic currents in its Membrane Potential (V_{mem}). Every time V_{mem} reaches the spiking threshold τ_{spike} , a digital output pulse is released. The number of spikes generated is antiproportional to the time difference between the two input spikes (See Fig. 1 e). When the time difference between the facilitatory and the trigger pulses is long, the gain value at the time of arrival of the trigger signal is not high enough to generate an EPSC. Thus, no spikes are generated. In case of a negative time difference (an event arrives first at the trigger synapse and then at the facilitatory input, as in Fig. 1 d), no output spikes are produced. Therefore, the TDE is direction-selective.

As introduced in section I, both the number of spikes within the burst and the burst duration depend on the time difference between the input pulses. The detection time range of the analog CMOS TDE implementation ranges from 10 nanoseconds up to hundreds of milliseconds, according to [19]. This range can be tuned by adjusting the LIF neuron's parameters in order to detect the timing differences more

Fig. 1. Theoretical behavior representation of theTDE model based on the model proposed by Milde et al. [19]. a) TDE schematic with facilitatory (fac) and trigger (trig) input and spiking output. b) Case one: A small positive time difference between facilitatory and trigger spike leads to a high number of output spikes (out). c) Case two: large positive time difference leads to no output spikes. d) Case three: A negative time difference leads to no output spikes. e) Number of TDE output spikes in dependency of time difference Δt between two input events (gain: gain factor, epsc: exponential postsynaptic current, mem: membrane potential).

precisely in an accurate spectrum, thus obtaining different TDE response profiles. Those profiles are known as tuning curves, which represent the neuron responses against the time difference between the facilitatory and the trigger input pulses. Furthermore, a nonlinear behavior of the tuning curves was expected to be obtained from the analog CMOS TDE implementation due to the transistors but, instead, a linear profile was observed. Milde et al. highlighted in [19] that the nonlinear response was manifested at the population response level, as well as in the temporal evolution of the Interspike Interval (ISI) distribution within a burst. This feature was taken into account for the digital model design proposed in this work, since it determines the way in which the temporal modules are implemented.

III. ARCHITECTURE AND IMPLEMENTATION

The proposed architecture is shown in Fig. 2. There are two event-based inputs: the facilitatory input ("facilitatory") and the trigger synapse ("trigger"). In addition, four configuration signals are available to set the model's parameters: the facilitatory weight, called "detection_time", which defines the maximum time difference that the model is able to encode, i.e. the time during which the gain value is non zero; the gain factor that influences the trigger synaptic weight ("tau"); the gain factor that influences the spike generation process ("weight"); and the decay time factor (decay) of the EPSC signal value. As output, there is a single event-based signal ("spike") which is the spike fired by the encoder. Beyond those signals, the system is governed by the system clock signal ("clock"),

Both control and event-based signals have 1-bit width. Internal data lines, as well as the "detection_time" signal,

have n -bit width, being n a generic parameter of the model denoted by "NBITS". On the opposite, the rest of data lines are m -bit width, being m also a generic parameter of the model denoted by "LOG2NBITS", which represents the result of the $\log_2$ NBITS. By default, "NBITS" value is set to 16, hence "LOG2NBITS" is set to 4. Data width plays a key role on the model behavior, since it defines the timing resolution and affects the output response due to the implementation details of the spike generator module.

By following the schematic presented by Moritz et al. [19], the proposed architecture was divided in three computational blocks: the gain generation, the EPSC generation, and the spike generation, shown in Fig. 2 in red, blue, and green respectively. A phenomenological design strategy, instead of biophysical emulation approach, was followed to implement the digital TDE model in order to avoid the computation of differential equations. Therefore, no floating point operations were employed. Instead, integer values were used. This approach was successfully adopted by Frenkel et al. [10] where linear operations were performed, thus reducing both the hardware cost and the model complexity.

A. Gain generator block

Upon an event is received at the facilitatory synapse, an exponentially decaying signal is generated, so-called gain. The decay time constant, as well as the input synaptic weight, determine the maximum time in which the facilitatory synapse current is non zero, i.e. the maximum time difference that the TDE is able to detect. Additionally, if more than one event arrive at the facilitatory input consecutively while the gain is higher than zero, the resultant gain value is the sum of the remaining gain value and the new gain value generated due to the input event. Therefore, a feedback mechanism is needed. In order to prevent the overflow effect, the gain block saturation level is controlled by the *GAIN_GEN_SAT* parameter.

The decaying signal has been implemented as a decreasing linear function by means of a countdown timer with pre-load value (represented by the *timer_0* module in Fig. 2). The pre-load value establishes the initial configuration of the timer, i.e. the amount of time that the timer is activated. Thus, this temporal window restricts the maximum time difference that the model is able to encode. The input signal "detection_time" sets that value, and it can be updated in real-time.

The feedback feature is achieved by internally appending an adder to the timer where its inputs are the timer's output and the aforementioned "detection_time" signal value, and the output is the timer's load value. Therefore, for each rising edge of the time reference signal "tr_tick", the *timer_0* module is updated according to equation 1.

$$tmr_0[k] = \begin{cases} tmr_0[k-1] + d_t & \text{if } faci == 1 \\ GAIN_GEN_SAT & \text{if } satu == 1 \\ tmr_0[k-1] - 1 & \text{if } tmr_0[k-1] > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Where $tmr_0[k]$ is the timer's output value at the time reference tick k , $k-1$ is the previous time reference tick, d_t

Fig. 2. Detailed block diagram of the TDE digital architecture. It is composed of three main blocks: gain generator block (red), EPSC generator block (blue) and spike generator block (green). Synchronous modules are indicated by squared corner blocks with a small triangle, while asynchronous modules are indicated with rounded corner blocks. The spike arrow is a 1-bit width signal, where events are either received as input or sent as output. Control arrows are also 1-bit width, and they act as flags. Finally, data arrows can be either n-bit or m-bit width, being used for the internal communication between blocks and also for loading configuration values.

is the unsigned integer constant value defined by the "detection_time" signal, $faci$ corresponds with the "facilitatory" input signal, and sat_u is a flag which is activated when the condition $(tmr_0[k-1] + d_t) \geq GAIN_GEN_SAT$ is true.

Two clock domains have been used to implement the digital TDE model. The main clock signal, defined in Fig. 2 as "clock", governs the control processes of the sequential blocks, as well as the input events detection and the output events generation. Furthermore, a second clock signal, called "tr_tick", is provided as time reference tick in order to allow the model to operate with different time scales, thus achieving an operational time range between nanoseconds and seconds. With this, the model acquires enough flexibility to be used along with a wide set of neuromorphic sensors, which can operate at different time resolutions. No internal clock generator has been implemented. Instead, an external configurable clock frequency divider is needed when a TDE module is instantiated. When multiple instances of a TDE unit are present, a shared clock frequency divider can be used rather than a single one per unit, hence reducing the overall hardware resources consumption and increasing the number of units that can fit into a design.

While in the standard LIF neuron model [26] each synapse outputs a postsynaptic current which integrates onto the membrane potential, the TDE facilitatory block generates a gain factor that regulates the trigger synapse weight. This way, to cover this feature on the proposed architecture, two mechanisms were implemented. First, the $timer_0$ output is weighted by the input signal "tau" in such a way the timer value is either right or left shifted by "tau" positions. The shift operation was implemented according to the Barrel shifter

[27], represented by equation 2.

$$d_out = \begin{cases} d_in * 2^{n_pos} & \text{if } l_r == 0 \\ d_in / 2^{n_pos} & \text{if } l_r == 1 \end{cases} \quad (2)$$

where d_out is the output value, d_in is the input data, n_pos is the number of positions to shift the input data, and l_r is for selecting whether the signal has to be shifted either to the left or to the right. Due to this being a combinational circuit, the output result is available at the same clock cycle, therefore avoiding the need for sequential blocks for the synchronization. The computed value is then fed as input of the $timer_1$, which generates the EPSC signal, acting as the trigger synapse weight.

Second, $timer_0$'s output value is weighted by the input signal "weight" also through a Barrel shifter module. In this case, the result has influence on the spike generation process in such a way that it controls the amount of spikes to be generated and, therefore, the precision of the encoding.

The shifted value is read by the spike generator block when a trigger pulse is detected. Based on the operating principles of the TDE model, detailed in section II, it can be deduced that the spikes-to-burst duration ratio is lower when the time difference between the facilitatory input and the trigger input is short compared to a longer time difference. That is, the former will produce many output spikes over a longer time bin, and the latter will produce less spikes but in a shorter time bin.

A register was included to store the last value used as input for the spike generator. This way, it can be used as feedback value to be added to the gain value, thus increasing the final gain value and, therefore, increasing the output spike rate.

Fig. 3. Gain generator block output example. First, the block receives a single facilitatory event. Immediately after, *timer_0* is loaded with the "detection_time" signal's value. The output value of *timer_0* decreases in one unit for each time reference tick, which was set to microseconds. Since the "weight"'s value was set to 1 (meaning that the timer's value is left shifted by one position), the *shift_1*'s output value is two times the value of *timer_0*. Second, the block receives multiple facilitatory events in order to show the accumulative behavior. *add_0* plot shows the value that is used as input for the spike generator block. In this case, its value matches with the *shift_1*'s value since no trigger event was received.

Equation 3 describes the gain feedback register, identified in Fig. 2 as *reg_0*.

$$reg_0 = \begin{cases} d_in & \text{if } trigg == 1 \\ 0 & \text{if } tmr_0[k] == 0 \\ reg_0 & \text{otherwise.} \end{cases} \quad (3)$$

Where *reg_0* is the value stored by the register, *d_in* is the last value loaded on the spike generator block, *trigg* corresponds with the input signal "trigger", and *tmr_0[k]* is the timer's output value at the time reference tick *k*. This register is reset to zero when the *timer_0* reaches zero, and it can also be disabled if needed.

Conceptually, it can be affirmed that the gain value has influence in both the temporal aspect (through the *tau* factor) and the amplitude aspect (through the *weight* factor) of the TDE response.

B. EPSC generator block

Similarly to the gain factor generation, when an incoming event is detected at the trigger's input, an exponentially decaying signal is generated, so-called Excitatory Post-Synaptic Current (EPSC). This signal is proportional to the gain signal due to the influence of the facilitatory synapse over the trigger synaptic weight [19]. Thus, the smaller the arriving time difference (Δt) between the facilitatory and trigger events is, the higher the gain factor is and, therefore, the higher the amplitude of the trigger synaptic current is.

Consequently, if a trigger pulse is detected without any previous facilitatory pulse, no EPSC current is generated since the gain factor is zero, as it is shown in Fig. 1 c). Nevertheless, if a trigger pulse is detected shortly after a facilitatory pulse (i.e. low Δt), an EPSC current proportional to the gain signal value at that time is generated. The generated EPSC current is high enough to generate spikes when it is integrated onto the membrane potential, as it is shown in 1 b). Equivalently,

if a trigger pulse is detected long time after the facilitatory pulse (i.e. large Δt), the resultant EPSC current might not be enough to produce output spikes, as in Fig. 1 c).

Multiple events can arrive to the trigger synapse while the gain factor is higher than zero, thus producing an accumulated EPSC current signal. The resultant signal is the sum of the left over EPSC current value and the left over weighted gain current value. Therefore, a feedback circuit is needed to limit the output current. The feedback value tends to decrease due to the decaying gain factor. However, a high input spike rate may saturate the EPSC current generation. This saturation level is set by the TDE generic parameter *EPSC_GEN_SAT*.

Following the same implementation principle of the gain generator block, the EPSC decaying signal was implemented as a decreasing linear function also by means of a countdown timer with pre-load value, identified as *timer_1* in Fig. 2. In this case, we can affirm that the pre-load value is the remaining time to zero of the gain generator block timer (*timer_0*), i.e. the gain current signal is zero. In order to maintain the synchronization with the gain generator block, the *timer_1* module is updated at every rising edge of the time reference signal "tr_tick" according to equation 4.

$$tmr_1[k] = \begin{cases} tmr_1[k-1] + (tmr_0[k-1] / 2^{tau}) & \text{if } trigg == 1 \\ EPSC_GEN_SAT & \text{if } satu == 1 \\ tmr_1[k-1] - 1 & \text{if } tmr_1[k-1] > 0 \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Where *tmr_1[k]* is the timer's output value at the time reference tick *k*, *k-1* is the previous time reference tick, *tau* is a factor to weight *timer_0*'s output value, *tmr_0[k-1]* is *timer_0*'s output value at the previous time reference tick, *trigg* corresponds with the input signal "trigger", and *satu* is a flag which is activated when the condition $(tmr_1[k-1] + (tmr_0[k-1] * 2^{tau})) \geq EPSC_GEN_SAT$ is true.

As it was aforementioned, the trigger timer determines the output spike burst duration as the same way the EPSC synaptic current decay in [19] is set by a voltage parameter. This synaptic current is injected into the neuron which integrates the current until it generates a spike as soon as the membrane potential rises above its threshold. Therefore, the neuron is able to produce spikes while the EPSC signal is higher than zero. In addition, the number of generated spikes is directly proportional to the EPSC signal duration, thus inversely proportional to the time difference between the facilitatory and trigger input spikes, in that order.

In the proposed design, the value which is loaded in the *timer_1* (i.e. in the EPSC generator) is called the remaining time to zero. This value has influence over the spike generator block in two similar aspects. Firstly, the spike generator block is active, i.e. producing spikes while the EPSC value is higher than zero. Thus, it acts as an enable signal. Moreover, the remaining time to zero is used to handle the temporal evolution of the spike generation process in order to mimic the ISI increment of the original analog TDE model [19] by exploding

a feature of the Exhaustive Synthetic Spikes Generator models proposed in [13] and implemented in [28] and [29].

The temporal evolution can be adjusted by the factor "*decay*", which weights the *timer_1*'s output also by means of a Barrel shifter module identified in Fig. 2 as *shift_2*. The timer's value decreases one unit for each "*tr_tick*" rising edge. With this factor, we can scale the decreasing speed, allowing us to obtain different range of values but preserving the time bin, i.e. the activation time of the spike generator module. Further details about the effect of this parameter are discussed in the next section where the spike generator block is presented.

Due to the implementation details of the spike generator block, the generated EPSC signal needs to be inverted, thus obtaining an incremental signal instead of a decreasing signal. This transformation can be achieved by storing the reference value and periodically subtracting the original value every time it is updated. In this case, the reference value corresponds with the pre-load value of the *timer_1* when a pulse is detected at the "*trigger*" input, and the original value is the *timer_1*'s output value. A generic register, denoted by *reg_1* in Fig. 2, was added to the proposed architecture, and its behavior is described by equation 5.

$$reg_1 = \begin{cases} tmr_1[k] * 2^{decay} & \text{if } trigg == 1 \\ reg_1 & \text{otherwise.} \end{cases} \quad (5)$$

Where *reg_1* is the value stored by the register, *decay* is the factor which weight the *timer_1* value, and *trigg* corresponds to the "*trigger*" input signal.

Note that the "*trigger*" signal is latched to let the timer load the pre-load value and to output the correct value, which takes one clock cycle of the main clock signal ("*clock*"). This latched "*trigger*" signal is shared by the *reg_1*, the *reg_0*, and the *spikes_generator_0* modules to keep the synchronization and to operate with the precise values.

The output of the subtractor module, whose output ranges from zero to *reg_1*'s output value, is then used as input of the spikes generator module that generates spikes accordingly to both the *add_0*'s output value and the *sub_0*'s output value, i.e. the gain factor value and the EPSC factor value, respectively.

C. Spike generator block

In the presence of an input facilitatory spike and an input trigger spike, a burst of output spikes is produced by the spike generator block. As it was detailed in subsections III-A and III-B, both the amplitude and the duration of the burst depend on the generated gain factor and the generated EPSC factor respectively. In contrast to the LIF neuron, which integrates the pre-synaptic current into the membrane and produces a spike if the membrane potential reaches a threshold, an event-based unsigned integer-to-spikes converter was implemented based on the model implementation by Jimenez-Fernandez et al. [28].

This converter, called Exhaustive Unsigned Synthetic Spikes Generator (EU-SSG), takes an unsigned integer value as input and produces a burst of spikes where both the number of output spikes and their distribution over the time are proportional to the input value. Similarly to the synaptic current integration

Fig. 4. EPSC generator block output example. First, the model was stimulated with a single facilitatory event, which is not represented, before the first trigger event. Then, the current value of *shift_1* is loaded in *timer_1*. Second, the accumulative effect is also shown when multiple triggers are received. The *sub_0* module generates an increasing signal, which is used as the clock divider value for the spike generator block.

Fig. 5. Exhaustive Unsigned Synthetic Spikes Generator (EU-SSG) block diagram. A complete description of both the implementation and the behavior of this module is presented in [28].

process which polarizes the membrane, producing a membrane potential, the EU-SSG implements an up counter that increases its value every time a pulse is detected, shown in Fig. 5 as *up_counter*. Then, the output of the counter is processed by the *Exhaustive Synthetic Method (ESM)* block, which determines the integration method. Finally, analogous to the comparison between the firing threshold and the membrane potential, the counter's output value and the input data are compared, and a spike is fired only when both values are equal.

To generate the pulse according to the desired output firing rate (expressed in spikes per second), a clock frequency divider module was used, identified as *clk_freq_div* in Fig. 5. It has two inputs signals: the system clock "*clock*" signal, and the "*clk_div*" signal that is the clock frequency division factor; and one output signal, the clock enable "*ce*" signal, which generates a pulse when corresponding.

By combining both the input value and the clock frequency divider value of the spike generator block properly, the desired behavioral output response of the original TDE model can be achieved. Analytically, the EU-SSG's output firing rate for a given input value was obtained following equation 6.

$$f(d_in) = \frac{F_{clock} * d_in}{2^n (clk_div + 1)} \quad (6)$$

Where *f* is the spike generator's output firing rate (expressed

in spikes per second), d_in is the input value to be converted to spikes, the constant F_{clock} is the system clock frequency (in Hz), n is data width, and clk_div is the internal clock frequency divider value.

Although the equation 6 is almost identical to the one defined in [28], there is a difference in the component 2^n due to the fact that in the original model the sign is taken into account (2^{n-1} is used instead), while in equation 6, the unsigned version is used. However, the number of bits selected to represent the input value does not affect the maximum firing rate achievable by the EU-SSG block. Let d_in be the maximum value that can be represented with n bits, which is 2^n . In that case, if d_in is replaced in equation 6, the maximum output firing rate is expressed as in equation 7.

$$f_{max} = \frac{F_{clock}}{(clk_div + 1)} \quad (7)$$

Where f_{max} is the spike generator's output maximum firing rate, the constant F_{clock} is the system clock frequency (in Hz), and clk_div is the internal clock frequency divider value. Therefore, the output firing rate only depends on both the system clock frequency and the clock frequency divider value. Since the maximum generating frequency is independent of the number of bits used to represent the values, this parameter can be set up according to the desired time resolution, thus benefiting the resources saving and allowing the implementation of a larger number of TDE units.

Two particular differences can be highlighted from this spike generator block compared to the LIF neuron model. Firstly, the EU-SSG block does not implement any refractory period. Therefore, the spikes can be produced through consecutive clock cycles. Instead, the clock frequency divider value needs to be adjusted in order to achieve the desired output spikes distribution. Secondly, the spike generator does not stop producing spikes. If the input value is higher than zero, the spike generator continues generating spikes. To stop the spike generation process, a clear input signal "*clear*" was added to the EU-SSG block, which resets its internal registers to zero. This control line is activated by the comparator *cmp_1* when the EPSC's timer output value reaches zero, meaning that the EPSC synaptic current is zero and, therefore, there is no current to integrate.

According to the architecture shown in Fig. 2, the EU-SSG input data ("*d_in*") corresponds to the TDE gain value. This signal depends on the current value of the *timer_0*, which determines the detection time. It also depends on the multiplication factor to control the number of spikes to generate, as well as the last input value loaded into the generator. Thus, the gain block's output signal, and therefore the "*d_in*" signal, can be defined as in equation 8.

$$d_in = reg_0 + (2^{weight} * timer_0) \quad (8)$$

Where d_in is the spike generator input value, reg_0 is the value stored in the register defined by the equation 3, *weight* represents the *timer_0* factor, and *timer_0* is the timer value defined by the equation 1.

In the same way, the EU-SSG clock divider value ("*clk_div*") corresponds to the TDE EPSC value. It depends on the current value of *timer_1*, which in turn depends on *timer_0* and *shift_0*, according to equation 4. In addition, the *timer_1*'s value is weighted to control the output firing rate and the ISI variation. Therefore, the EPSC block's output value can be defined as in equation 9.

$$clk_div = reg_1 - (2^{decay} * timer_1) \quad (9)$$

Where clk_div is the spike generator input value, reg_1 is the value stored in the register defined by equation 5, *decay* represents the *timer_1* factor, and *timer_1* is the timer value defined by equation 4.

A behavioral example of the proposed model is shown in Fig. 6. The first row shows the input events differentiated by colors and following the color code used in Fig. 1: red color for the facilitatory event and blue for the trigger event. The second and the third rows represent the evolution of the gain and EPSC timers over time, respectively. The fourth row illustrates the "*d_in*" signal value loaded into the spike generator block, and the fifth one the clock divider value of that block. Finally, the sixth row shows the output spikes produced in the presence of the shown input stimuli.

When an event is detected at the facilitatory input, the gain timer is initialized according to its detection time value. For each time reference tick, the gain timer decreases its value. As soon as an event is detected, the current value of the gain timer is loaded into the EPSC timer. Concurrently, that value is weighted and loaded into the spike generator as input data. Therefore, the spikes begin to be generated.

At this moment, the clock divider value of the spike generator is set to zero, thus producing the spikes at the maximal firing rate. For each time reference tick, the EPSC timer decreases its value and the clock divider value of spike generator block is updated to a higher value. This causes a decrement in the output firing rate and consequently an increment of the ISI between two consecutive output events.

When another spike is detected at the trigger input, the current value of the facilitatory is added to the current value of the trigger timer. The input data of the spike generator block is updated, and the clock divider value is reset to zero. The TDE produces spikes until the EPSC timer reaches zero, when the internal stop signal is enabled. A more exhaustive behavioural analysis on how the TDE responds in the presence of both simple and complex stimuli is presented in the following sections.

IV. ANALYSIS AND RESULTS

Three test scenarios were considered in order to validate and analyse the proposed model. First, a behavioral simulation was performed for the most probable input stimulus combinations. The results were cross-validated with the results presented in the reference work [19]. Once validated, the model was analyzed and characterized by carrying out different experiments where the ISI distribution and the number of output spikes were measured. Second, a single TDE unit was synthesized for an FPGA platform. The output spikes obtained from the FPGA

Fig. 6. An operation example of the proposed model. The detection time was set to $700 \mu s$; the tau value was set to 0; the weight value was set to 4; the decay value was set to 2. The main clock was set to 50 MHz, time reference tick was set to 1 MHz, and the data width was set to 16. The Δt value between the facilitatory event and the first trigger event is $200 \mu s$, while the Δt value between the facilitatory event and the second trigger event is $400 \mu s$.

were measured using an oscilloscope and recorded using a computer. A quantitative evaluation was carried out between the simulation and the deployed version of the TDE unit. Third, a proof-of-concept of a sound source lateralization task was designed and tested using a population of TDE units.

A. Simulation

As the TDE model has two inputs, many different input event combinations can occur. It is important to study each of these scenarios since they will directly effect the behavior of the model and, thus, its response. With the aim to verify whether the behavior of the proposed model matches with the expected output, twelve cases were simulated. For this experiment, a single TDE unit, with $100 \mu s$ as detection time, a tau value of 0, a gain value of 5, a decay value of 1, and both the facilitatory and trigger saturation values of 256 was instantiated. In addition, the time resolution was set to microseconds. Fig. 7 presents the response of the TDE unit when being excited by twelve different sequences of input events.

Cases A and B depict simple examples where either a single facilitatory or trigger event are received by the TDE unit. No events were produced at the output. However, while the gain signal started being generated in case A as response to the facilitatory event, the trigger signal remains at zero in case B since no facilitatory event was received before. Indeed, this effect can also be seen in case C, where the trigger event is received just a few microseconds before the facilitatory event. The same response of the model is obtained when both events arrive at the TDE unit at the same time, as shown in D.

When a facilitatory event is presented at the TDE's input before the trigger event, the TDE's response is inversely proportional to the time difference (also called Δt) between both. Cases E to G show the output events generated by the

TDE unit for short, medium and long Δt values, respectively. As it can be seen, the number of output events decreases with higher Δt values while the ISI increases, matching the expected behavior. When Δt is higher than the detection time (case H), the resulting response is the same as first having case A and then case B, meaning that no events are generated at the output.

The proposed model was also simulated and evaluated in the presence of more realistic input patterns. In a real world application, the input events are not received one by one. Instead, a continuous rate of events can be injected to the input. Cases I and J show the TDE model response when multiple facilitatory or trigger events are received at the input for a single opposite event.

On the one hand, case I shows how the EPSC signal is incremented by a value proportional to the remaining time to zero of the gain signal with the arrival of the second trigger event, following equation 4. The first produced burst contains more events with lower ISI whereas the opposite happens in the second burst. On the other hand, case J shows how the gain signal is incremented by the detection time value when the second facilitatory event arrives, according to equation 1. This alters the response of the simple case shown in F, generating a burst with a higher number of events.

When many consecutive facilitatory-trigger pairs with Δt lower than the detection time parameter are received, the accumulated EPSC/gain signal value will increase. Thus, after some time, the signal will saturate according to the saturation parameter. At the saturation level, the TDE output firing rate is considered as the maximum, and its behavior can be estimated by using equation 7. We can affirm that the saturation parameter limits the TDE response, and its value would depend on the used application and its context.

To this regard, the behavioral validation of the proposed TDE model shown in Fig. 7 has been proven to act in accordance with the reference model [19] in terms of both performance and requirements. Moreover, cases which were not evaluated in the original model were also reported in order to fully characterize the proposed model.

After the behavioral validation, a more precise timing analysis of the TDE response was carried out. In [19], this study was performed by investigating the ISI distribution within a burst for six different Δt values. The ISI was calculated as $ISI_n = t_n - t_{n-1}$, where t_n is the timestamp of the n_{th} event. The authors reported that the obtained results matched with the expected nonlinear response in the temporal evolution of the ISI within a burst.

The same test was done in order to verify that the nonlinear ISI variation feature was also achieved by the proposed model. Fourteen facilitatory-trigger pairs of events with different Δt values were used as input stimuli. Two TDE units were configured to work at different time scales by setting the time reference tick to microseconds and milliseconds, respectively.

Fig. 8 presents the results obtained from the simulations with time reference tick in the scale of microseconds, where a TDE unit with $700 \mu s$ as detection time, 0 as tau, 4 as weight, and 2 as decay was instantiated. Similarly, Fig. 9 presents the results obtained from the simulations but setting the time

Fig. 7. RTL simulation for twelve basic cases of the TDE model with time resolution of microseconds. Following the color code, red is related to the facilitatory, blue is related to the trigger, and green is related to the spike generator.

Fig. 8. TDE ISI distribution for a facilitatory-trigger pair with different Δt values for a microseconds resolution configuration. Note that the smallest Δt value used was not zero (no output events would be produced) but 20 ns (one clock cycle).

reference tick to milliseconds, where a TDE unit with 70 ms as detection time, 0 as tau, 0 as weight, and 3 as decay was instantiated.

Nonlinear profiles can be clearly observed in all the cases. Those profiles were obtained by using exclusively linear operations and circuits, and thus avoiding explicit circuitry for generating exponential behaviors. This feature allow to reduce the needed resources and therefore to increase the total number of TDE units that can be instantiated into an FPGA or Application-Specific Integrated Circuit (ASIC).

When the Δt value is almost equal to the configured detection time (e.g. above 550 μs for microseconds and 55 ms for milliseconds), the produced output events are not enough to represent the characteristic curve that cases with lower Δt presented. On the other hand, the first output event pairs seem to have the same ISI value for most Δt values (specially for lower Δt values). These ISI values cannot be correctly appreciated in the plot, since, according to the global clock, the precision of the minimum time difference is in the order

Fig. 9. TDE ISI distribution for a facilitatory-trigger pair with different Δt values for a milliseconds resolution configuration.

of nanoseconds, and the Y-axis of the plot is represented in milliseconds.

The time scale set by the time reference tick affects not only the ISI curves, which posses a better and more uniform distribution for the millisecond time reference, but also the number of the produced output events. This effect is caused by the combined use of both clock domains in the spike generator module, where the time reference clock is used to manage the inputs and the global clock is used to produce the events.

Another simulation was carried out in order to prove the variation on the number of output events generated by the TDE unit using both different time references and detection times. Results are depicted in Fig. 10 and Fig. 11 for a microsecond and a millisecond time reference, respectively.

Note that the peak located at $\Delta t = 300 \mu s$ (shown in Fig. 10) is caused by the implementation of the spike generator module, since the conversion from an integer value to a spike stream has an intrinsic error. This error is maximal at that time in this particular example, and it is deeply analyzed in [29]. Because of the timing resolution used in Fig. 11, even if the

Fig. 10. Number of TDE output spikes over Δt variation for microsecond time reference tick.

Fig. 11. Number of TDE output spikes over Δt variation for millisecond time reference tick.

error exists, the peak cannot be appreciated.

The characteristic curve that relates the number of output events produced by the TDE with respect to the Δt value between its facilitatory and its trigger inputs is known as tuning curve. By varying the parameters' values of the TDE unit, its tuning curve can be adjusted. Therefore, it is possible to have a set of TDE units with different tuning curves.

This feature allows to configure a TDE population with different tuning curves responding to different input patterns or using the population response as a global response. Differently to the tuning curve test carried out in [19], in which all the neurons shared the same parameters, we conducted a similar test but varied the TDEs configurations.

Table I summarizes the values used for each TDE unit within the population created for this test. The population size is four units. All the units share the saturation value, set to 3000 for both the gain and EPSC signals, as well as the τ value, which was set to zero.

TABLE I
PARAMETERS OF THE TDE POPULATION

Neuron ID	weight ($\mu\text{s/ms}$)	decay ($\mu\text{s/ms}$)	detection time ($\mu\text{s/ms}$)
TDE 0	9 / 4	2 / 1	100 / 10
TDE 1	6 / 2	1 / 2	300 / 30
TDE 2	5 / 1	2 / 3	500 / 50
TDE 3	3 / 0	2 / 12	700 / 60

The time difference of the two input events was varied from

Fig. 12. Individual TDE tuning curves and population tuning curve for microsecond time reference tick.

Fig. 13. Individual TDE tuning curves and population tuning curve for millisecond time reference tick.

20 nanoseconds to 750 microseconds for the microsecond time reference, with a 20 microseconds step size. Similarly, the relative timing was varied from 20 nanoseconds to 75 milliseconds for the millisecond time reference, with a 5 milliseconds step size. Obtained results from simulations are depicted in Fig. 12 and Fig. 13.

Both figures show the effect of the different tuning parameters in the output response of the TDE units. The TDE0's tuning curve presents a noticeable slope, meaning that it has a high output firing rate in a short time bin. On the contrary, the TDE1's tuning curve has almost a flat slope, which means it produces less spikes but during a longer time period.

Even though the individual TDE's tuning curves look practically flattened, the global behavior of the population, calculated as the sum of the output events for each TDE unit for each Δt value, fits with an exponential curve. For the microsecond time reference, the obtained exponential fitting curve had a $R^2 = 0.87$ with a $RMSE = 14.61$. For the milliseconds case, the R^2 value was 0.93 with a $RMSE = 42.24$.

The spike generator intrinsic error directly effects the expo-

Fig. 14. Block diagram of the setup used for real-time measurements acquisition.

nential approximation, but the fitting curve can be considered acceptable taking into account that a nonlinear profile was obtained by using exclusively linear modules.

B. FPGA

After simulating the proposed model, analyzing and validating its behavior, a TDE population was deployed into a FPGA-based device in order to verify the results obtained in simulation but running on a hardware platform. Fig. 14 depicts the setup used for this test.

The upper part describes in detail the implemented design deployed into the FPGA. Two timers with periodic interrupts were used to generate both the facilitatory and the trigger events. The time reference tick was set to microseconds and the Δt value was fixed to $100 \mu s$, having a wait time of 1 second between two consecutive stimulus generations. The population size was set to four in order to maintain the same architecture than in the simulation. Therefore, a Read Only Memory (ROM) module was added for storing the population parameters, which were the same as presented in table I for the microseconds case. Finally, an events monitor was connected to the population output to collect the events and to send them to the computer by using an Address-Event Representation (AER) protocol.

The lower part describes the two approaches used to measure the population response directly from the hardware. On the one hand, an oscilloscope was used to both measure and visualize the output events from the TDE3. Fig. 15 shows a screenshot with some of the captured events, where the increment of the ISI over time can be appreciated. On the other hand, a computer running the jAER software [30] was used to visualize and save the population output events in real time.

Besides the behavioral simulation, a post-synthesis and a post-implementation simulation were performed. Then, a wide comparison including all the simulation results, as well as the measurements from the oscilloscope and the events collected by the events monitor, was carried out. By taking the results from the behavioral simulation of the TDE3 as reference,

Fig. 15. Output spikes captured by using an oscilloscope.

Fig. 16. Comparison of the TDE response from different measurement sources.

the Pearson correlation value [31] was calculated for each measurement source.

Results were plotted in Fig. 16, showing a high correlation level (0.99 as the lowest value) and having the biggest differences in the later spike pairs. This could be caused by the inherent sampling error of the used devices. Nevertheless, the high correlation degree demonstrate that the TDE behavior does not change when it is deployed into an FPGA-based hardware in real time.

TABLE II
HARDWARE RESOURCES UTILIZATION FOR DIFFERENT FPGA DEVICES.

FPGA chip	Slice Registers Used / Available	LUT Slice Used / Available	Max. num. TDE units
XC6SLX150T (Spartan-6)	122/184304 (0.07%)	207/92152 (0.23%)	445
XC7A75T (Artix-7)	140/94400 (0.15%)	179/47200 (0.38%)	263
XC6SLX150T (Kintex-7)	140/597200 (0.02%)	179/298600 (0.06%)	1668

The resources needed by a single TDE unit were estimated for three different FPGA chips with different characteristics.

In addition, the maximum number of TDE units which can be instantiated on each was reported. Table II summarizes all estimations. Finally, a power consumption study was carried out for the XC6SLX150T chip, which was also used for all the measurements and real-time experiments in this work. A set of Switching Activity Interchange Format (SAIF) files were used for a realistic estimation, reporting a power consumption of less than 1 mW, with a static power consumption of 98% (intrinsic to the FPGA).

C. Real-time neuromorphic application

The applicability of the proposed TDE model was evaluated by means of a proof-of-concept application. In the work presented in Milde et al. [19], the proof-of-concept was focused on a neuromorphic application using visual information generated by event-based cameras. Therefore, in this work, a real-time sound source lateralization application for FPGA was implemented using the Neuromorphic Auditory Sensor (NAS).

Briefly defined, the sound source lateralization is considered as the capability to identify where the sound source is by just using binaural cues [32]. Simplifying the concept, we will consider the sound source lateralization as the ability to determine whether the sound is on the left, on the middle, or on the right.

For this task, the same binaural cues that are commonly used for sound source localization (Interaural Time Difference (ITD) and Interaural Level Difference (ILD)) can be used. A binaural sensor is needed in order to be able to capture those cues. The Neuromorphic Auditory Sensor (NAS), proposed by Jimenez-Fernandez et al. [23], is a neuromorphic sensor capable of decomposing the input sound from a pair of microphones into its frequency components, emulating the human cochlea. A general NAS architecture was depicted in 17 top. As event-based sensor, the output is encoded as events, thus the information is coded not only in the number of output spikes but also in the relative time between them. Only the ITD cue was used in the proof-of-concept application due to the timing nature given by the proposed TDE model.

The position of a sound source in space can be encoded by the temporal difference between the arrival of the sound waves at the ipsilateral side and the contralateral side. This time difference is known as the ITD. According to the specifications of the TDE model, an output response is exclusively produced if the incoming facilitatory event arrives before the incoming trigger event. Thus, two TDE populations were needed to perform the sound source lateralization task: one for detecting when the sound source is located at the left of the reference (the microphones pair) and one for detecting when the sound source is located at the right.

Although the auditory information used was the same, thus containing the same temporal information, it projects onto the two TDE populations in an opposite way. The network architecture is shown in Fig. 17 bottom, and the parameters values used for the TDE units' configuration were the same than the ones presented in table I for the microseconds time reference. Therefore, the individual tuning curves, as well as

the population tuning curve, correspond to the plot shown in 12.

The test scenario was designed as follows: first, a virtual room of $10 \times 10 \times 2$ meters was created by using the Room Impulse Response (RIR) generator [33] software tool. A pair of directional microphones were placed in the center of the room, imitating the human's ears disposition in a regular head, and at one-meter height over the floor. Then, three sound sources were placed at -90 , 0 , and $+90$ degrees with respect to the microphones pair, corresponding to the left, front, and right positions, respectively, with a separation of two meters. The sound sources generated a pure tone beep of 500 Hz, with a duration of 0.5 seconds, every one second.

Regarding the auditory sensor, a 64-frequency-channels, binaural NAS, with a frequency range from 22 Hz to 22 KHz was generated using OpenNAS tool¹. The output events from the frequency channel number 33 were used as input of the TDE population since the center frequency of its associated event-based band pass filter was set to 502.38 Hz.

The sound was sent from the computer to the FPGA in real time, and the TDE population's output events were collected in a computer by using jAER. Obtained results from the real-time experiments are shown in Fig. 18. Top and bottom plots show the response of the TDE populations when the sound source was placed on the very left and very right positions (high ITD values).

As expected, most of the activity was produced by the left and right TDE populations, respectively. The maximum overall activity was found in those TDE units with higher detection time (TDE 3 for the left side and TDE 7 for the right side). On the contrary, TDE 0 and 4 barely presented any activity due to their low value for the detection time.

The center plot shows the response of both populations when the sound source was placed in front of the microphones pair. In this case, all the instantiated TDE units produced output events as a response to the low ITD value inherent to the input stimuli. Note how the shape of the normalized overall activity resembles to the tuning curve of the TDE population presented in 12. This activity could be post-processed in order to extract more precise spatial information of the sound source.

V. DISCUSSION AND CONCLUSION

In this work, we have presented a digital implementation of the Time Difference Encoder (TDE) model. The full project code is open-source and publicly available on GitHub (https://github.com/dgutierrezATC/TDE_vhdl). This event-based design encodes temporal differences into a burst of events. A phenomenological design strategy was followed in order to implement the TDE behavior with reduced design complexity. After being simulated, no differences were found compared with the expected behavior reported in [19], where a biophysical emulation approach was followed.

Once simulated, the model was deployed into an FPGA-based device in order to characterize its response in a real-time platform. Although the proposed model is also suitable to be implemented in an ASIC, an FPGA-based platform was

¹<https://github.com/RTC-research-group/OpenNAS>

Fig. 17. Detailed block diagram of the FPGA top module for the proof-of-concept, containing both the NAS and the TDE populations.

Fig. 18. Raster plot of the output events from the TDE population and normalized overall activity. Plots were generated using the python version of the NAVIS tool [34]. TDEs 0 to 3 (left population) correspond to indexes 0 to 3 in I, and TDEs 4 to 7 (right population) also correspond to indexes 0 to 3 in the same table. Therefore, TDE 0 and TDE 4 use the same configuration and so on.

considered due to its reconfigurability and fast prototyping workflow. This also allows to create custom TDE populations with different parameters and to scale up the population's size if needed, which is not possible in an either analog or digital ASIC.

Another advantage of the FPGA TDE implementation is the wide temporal resolution range that it offers. It can be adjusted for building specific applications in different temporal domains, such as sound source localization (microseconds), vision (milliseconds), and odour localization (seconds) just by setting a few integer values. In this case, the mismatch problem will not affect the behavior in any of the mentioned cases of

use.

However, it is also important to mention that, unlike the analog implementation, the model is not fully asynchronous. This feature forces the model to have clock signals, thus increasing the static power consumption due to the switching activity. Therefore, the reported power consumption of the proposed model is significantly higher (less than 1 mW) compared to the few nanoWatts of power consumption for low output spike frequencies (approximated from [35] and [36]) given by the subthreshold operation level in which the analog version works. In addition, the main drawback of the proposed TDE model resides on the tuning process, since it depends on the context of the application, and, consequently, some of its parameters need to be set by trial and error.

In this work, a proof-of-concept of a sound source lateralization application was presented to evaluate the usability and performance of the proposed TDE model with more realistic input stimuli. The use of the TDE model for sound source lateralization represents an alternative to the biologically-inspired Jeffress model [37]. Latter consists of an array of coincidence detector neurons with variable delay lines. The coincidence detector with a delay most similar to the signal's ITD indicates the sound source location with its dominant response.

As demonstrated in section IV-C, we can extract the sound source location from the TDE's spiking frequency only using four units for each side. The combination of TDEs with different facilitatory time constants leads to an exponential population response which further increases the model's accuracy. Also, the output of the TDE population could be sent to other neuromorphic processors (as Loihi [7] or SpiNNaker[11]) to further improve the localization accuracy by using SNNs.

ACKNOWLEDGMENT

The authors would like to thank Moritz Milde and Philipp Klein for their support. The authors would like to acknowledge the 2019 Capocaccia Neuromorphic Workshop and all their participants for the fruitful discussions. In addition, the authors would like to acknowledge NEUROTECH for the Capocaccia scholarships awarded to Daniel Gutierrez-Galan and Thorben Schoepe.

REFERENCES

- [1] L. Alex, "The Von Neumann architecture topic paper #3," *Computer Science*, vol. 319, pp. 360–8771, 2009.
- [2] S. Wild, "Why is dual-pivot quicksort fast?" *CoRR*, vol. abs/1511.01138, 2015. [Online]. Available: <http://arxiv.org/abs/1511.01138>
- [3] L. B. Kish, "End of Moore's law: thermal (noise) death of integration in microand nano electronics," *Physics Letters A*, vol. 305, pp. 144–149, 2002.
- [4] J. Rigden, "Body, physics of," 1996.
- [5] D. Drubach, *The brain explained*. Prentice Hall, 2000.
- [6] H. Moravec, "When will computer hardware match the human brain?" *Journal of Evolution and Technology*, vol. 1, 1998.
- [7] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.
- [8] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in neuroscience*, vol. 5, p. 73, 2011.
- [9] R. Tapiador-Morales *et al.*, "Neuromorphic LIF row-by-row multiconvolution processor for FPGA," *IEEE transactions on biomedical circuits and systems*, vol. 13, no. 1, pp. 159–169, 2018.
- [10] C. Frenkel *et al.*, "A compact phenomenological digital neuron implementing the 20 Izhikevich behaviors," in *2017 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2017, pp. 1–4.
- [11] S. B. Furber *et al.*, "Overview of the spiNNaker system architecture," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, Dec 2013.
- [12] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, pp. 1–23, 2011.
- [13] A. Linares-Barranco *et al.*, "On algorithmic rate-coded AER generation," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 771–788, 2006.
- [14] E. T. Rolls *et al.*, "The neuronal encoding of information in the brain," *Progress in neurobiology*, vol. 95, no. 3, pp. 448–490, 2011.
- [15] J. P. Dominguez-Morales *et al.*, "Multilayer spiking neural network for audio samples classification using SpiNNaker," in *International Conference on Artificial Neural Networks*. Springer, 2016, pp. 45–53.
- [16] F. Theunissen *et al.*, "Temporal encoding in nervous systems: a rigorous definition," *Journal of computational neuroscience*, vol. 2, no. 2, pp. 149–162, 1995.
- [17] Y. Yi *et al.*, "FPGA based spike-time dependent encoder and reservoir design in neuromorphic computing processors," *Microprocessors and Microsystems*, vol. 46, pp. 175–183, 2016.
- [18] D. S. Reich *et al.*, "Interspike intervals, receptive fields, and information encoding in primary visual cortex," *Journal of Neuroscience*, vol. 20, no. 5, pp. 1964–1974, 2000.
- [19] M. B. Milde *et al.*, "Spiking elementary motion detector in neuromorphic systems," *Neural computation*, vol. 30, no. 9, pp. 2384–2417, 2018.
- [20] T. Schoepe *et al.*, "Neuromorphic sensory integration for combining sound source localization and collision avoidance," in *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*. IEEE, 2019, pp. 1–4.
- [21] P. Lichtsteiner *et al.*, "A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor," *IEEE journal of solid-state circuits*, vol. 43, no. 2, pp. 566–576, 2008.
- [22] M. Yang *et al.*, "A 0.5v 55 μw 64x2-channel binaural silicon cochlea for event-driven stereo-audio sensing," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*, Jan 2016, pp. 388–389.
- [23] A. Jiménez-Fernández *et al.*, "A Binaural Neuromorphic Auditory Sensor for FPGA: A Spike Signal Processing Approach," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 28, no. 4, pp. 804–818, 2017.
- [24] S.-C. Liu *et al.*, "Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. IEEE, 2010, pp. 2027–2030.
- [25] V. Chan *et al.*, "AER EAR: A matched silicon cochlea pair with address event representation interface," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 1, pp. 48–59, 2007.
- [26] W. Gerstner *et al.*, *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
- [27] A. Ito, "Barrel shifter," May 9 1989, uS Patent 4,829,460.
- [28] A. Jimenez-Fernandez *et al.*, "Building blocks for spikes signals processing," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–8.
- [29] F. Gomez-Rodriguez *et al.*, "Two hardware implementations of the exhaustive synthetic AER generation method," in *Computational Intelligence and Bioinspired Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 534–540.
- [30] T. Delbruck, "jAER open source project," *Internet: <http://jaer.wiki.sourceforge.net>*, 2007.
- [31] J. Benesty *et al.*, "Pearson correlation coefficient," in *Noise reduction in speech processing*. Springer, 2009, pp. 1–4.
- [32] G. Plenge, "On the differences between localization and lateralization," *The Journal of the Acoustical Society of America*, vol. 56, no. 3, pp. 944–951, 1974.
- [33] E. A. Habets, "Room impulse response generator," *Technische Universiteit Eindhoven, Tech. Rep.*, vol. 2, no. 2.4, p. 1, 2006.
- [34] J. P. Dominguez-Morales *et al.*, "NAVIS: Neuromorphic Auditory Visualizer tool," *Neurocomputing*, vol. 237, pp. 418–422, 2017.
- [35] C. Bartolozzi *et al.*, "An ultra low power current-mode filter for neuromorphic systems and biomedical signal processing," in *2006 IEEE Biomedical Circuits and Systems Conference*, 2006, pp. 130–133.
- [36] P. Livi *et al.*, "A current-mode conductance-based silicon neuron for address-event neuromorphic systems," in *2009 IEEE International Symposium on Circuits and Systems*, 2009, pp. 2898–2901.
- [37] L. A. Jeffress, "A place theory of sound localization," *Journal of comparative and physiological psychology*, vol. 41, no. 1, p. 35, 1948.

Daniel Gutierrez-Galan received the B.S. degree in computer engineering in 2014 and the M.S. degree in computer engineering and networks in 2016, both from the University of Seville, Sevilla, Spain. Since September 2017, he has been a Ph.D. student in the Department of Computer Architecture and Technology, at University of Seville, with a research grant from the Spanish Ministry of Education and Science. He is member of the Robotics and Technology of Computers Lab since 2016. His research interests include embedded systems programming, digital design, FPGA, spiking neural networks in embedded systems for audio processing, neuromorphic auditory sensors and neuromorphic robots.

Thorben Schoepe received the B.Sc. degree in Biomimetics in 2015 from the Westphalian University of Applied Science and the M. Sc. degree in BioMechatronics in 2018 from the Bielefeld University and the University of Applied Sciences Bielefeld. He has been a PhD student in the Neuromorphic Behaving Systems group at the Faculty of Technology and Cognitive Interaction Technology Center of Excellence (CITEC) at Bielefeld University since September 2018. In November 2020 he moved to the Bio-Inspired Circuits and Systems (BICS) Lab. Zernike Institute for Advanced Materials & Groningen Cognitive Systems and Materials Center at the University of Groningen to continue his PhD. His research interests include insect-inspired neurorobotics, navigation, analog design, neuromorphic sensors and sensory pre-processing.

Juan Pedro Dominguez-Morales was born in Sevilla (Sevilla, Spain) in 1992. He received the B.S. degree in computer engineering, the M.S. degree in computer engineering and networks, and the Ph.D. degree in computer engineering (specializing in neuromorphic audio processing and spiking neural networks) from the University of Seville, in 2014, 2015 and 2018, respectively. His Ph.D. was granted with a research grant from the Spanish Ministry of Education and Science. Since January 2019, he has been working as Assistant Professor in the same university. He is member of the Robotics and Technology of Computers Lab since 2015. His research interests include neuromorphic engineering, spiking neural networks, audio processing and deep learning. He has been an IEEE member for four years.

Alejandro Linares-Barranco (SM'17-M'04) received the B.S. degree in computer engineering, the M.S. degree in industrial computer science, and the Ph.D. degree in computer science (specializing in computer interfaces for neuromorphic systems) from the Universidad de Sevilla, Sevilla, Spain, in 1998, 2002, and 2003, respectively. From 1998 to 2000, he was a Member of the Technical Staff at the Sevilla Microelectronics Institute (IMSE-CNM-CSIC). From 2001 to 2006, he was an Assistant Professor at the Architecture and Technology of Computers Department of the University of Sevilla. He was promoted to Associate Professor in 2006, and in 2020 he was accredited as full professor. He is member of the Robotics and Computer Technology Lab since 2001, where his research interest includes AER-tools for debugging and communicating AER systems, VLSI digital design, neuromorphic computing for sensors, robotics, and deep-learning. He was officer of the Technical Committee on Neural Systems and Applications (NSATC) of the IEEE Circuits and Systems Society (2010-2014).

Angel Jimenez-Fernandez received the B.S. Degree in Computer Engineering in 2005, the M.S. Degree in Industrial Computer Science in 2007 and the Ph.D. in Neuromorphic Engineering in 2010 from the University of Seville, Sevilla, Spain. In October 2007 he became Assistant Professor at the department of Computer Architecture and Technology of the University of Seville. In April 2012, he was promoted to Associate Professor at the Architecture and Technology of Computers Department of the University of Seville. He is member of the Robotics and Technology of Computers Lab since 2004. His research interests include neuromorphic engineering applied to robotics, real-time spikes signal processing, neuromorphic sensors, field programmable gate array (FPGA) digital design, embedded intelligent systems development, and biomedical robotics.

Elisabetta Chicca (SM'16-M'06) is chair of the Bio-Inspired Circuits and Systems (BICS) lab in the Faculty of Science and Engineering at the University of Groningen (Netherlands). She studied physics at the University of Rome, where she graduated in 1999. In 2006 she received a PhD in Natural Sciences (ETHZ Zurich, Switzerland) and in Neuroscience (ZNZ Zurich). Immediately after the PhD, she started a PostDoc at the Institute of Neuroinformatics at the University of Zurich and ETH Zurich, where she continued working as Research Group Leader from May 2010 to July 2011. E. Chicca led the Neuromorphic Behaving Systems lab at Bielefeld University (Germany) from August 2011 until July 2020. She has a long-standing experience with developing event-based neuromorphic systems and their application to biologically inspired computational models. Her research activities cover models of cortical circuits for brain-inspired computation, learning in spiking neural networks, bio-inspired sensing and motor control. Photograph by Sylvia Germes.