Neuromorphic Signal Filter for Robot Sensoring

Luis M. García-Sebastián*, Victor H. Ponce-Ponce*, Humberto Sossa, Elsa Rubio-Espino and José A. Martinez-Navarro

Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico City, México

Noise management associated with input signals in sensor devices arises as one of the main problems limiting robot control performance. This article introduces a novel neuromorphic filter model based on a leaky integrate and fire (LIF) neural model cell, which encodes the primary information from a noisy input signal and delivers an output signal with a significant noise reduction in practically real-time with energy-efficient consumption. A new approach for neural decoding based on the neuron-cell spiking frequency is introduced to recover the primary signal information. The simulations conducted on the neuromorphic filter demonstrate an outstanding performance of white noise rejecting while preserving the original noiseless signal with a low information loss. The proposed filter model is compatible with the CMOS technology design methodologies for implementing low consumption smart sensors with applications in various fields such as robotics and the automotive industry demanded by Industry 4.0.

Keywords: neuromorphic, filter, CMOS, low-frequency, sensing

1. INTRODUCTION

The term neuromorphic, coined by Mead (1990), refers to Very Large Scale of Integration (VLSI) systems aiming to reproduce biological neuron behaviors. Neuromorphic computing platforms are relatively simple regarding the number of active elements (transistors) compared to complex traditional digital units (microprocessors) to replicate brain-like responses. Today, the convergence of electronics, computing science, and neuroscience offers bountiful inspiration to explore novel hardware structures, algorithms, and innovative ways to process information more efficiently, maintaining low levels of energy waste and material use (Schuman et al., 2017). One of the most remarkable contributions of this inter-discipline convergence is the conception of spiking neurons (SN), also called the third generation of artificial neurons (Maass, 1997). The main difference concerning previous generations is the inclusion of temporal information in the computing process, and this feature offers the possibility to process signals efficiently with variations across time. Unfortunately, the large-scale modeling of SN units is limited due to the high computational cost involved in solving numerically the whole set of differential equations representing each SN unit. Therefore, the design and implementation of these units are more convenient at the silicon plane and in the analog domain to overpass this vast amount of numerical computation effort.

Traditional analog filters are designed based on scaling specific frequency domain signal components and attenuating the rest. This approach has been proven effective with noise that is primarily out of signal frequency range. However, linear filters cannot clear noisy signals when disturbance affection is in the same frequency range as primary signal. In this context, digital filters, especially average filter techniques, take precedence at the cost of resources expense. For this reason, some filter proposals based on the use of several SNs have been made...
(Orchard et al., 2021; Sharifshazileh et al., 2021). Generally, integrated circuits hosting neuromorphic implementations possess an inherent capacity to extract primary features of given entries since integrals tied to the SN model can be interpreted as the average value operator on a time window. Thus, neuromorphic systems allow average filtering while retaining the benefits of analog circuits.

2. METHODS
2.1. Neural Circuit
There are several proposals reported on analog implementations of neural model circuits (Abbott, 1999; Wijekoon and Dudek, 2008; Zamarreno-Ramos et al., 2011; Wu et al., 2015; Zare et al., 2021). Throughout the development of this study, the leaky integrate and fire neuron (IFN) circuit, proposed in Wu et al. (2015) was used, as seen in Figure 1. However, the proposed methodology could be easily adapted to work with other neural circuit models. This proposal is divided into two main parts, a leaky, current integrator circuit (LI) to emulate the behavior of a neuron during the period of depolarization and a reset engine that returns the output voltage of the operational amplifier (V\text{mem}) to a reference voltage level (V\text{ref}). It also assumes the generation of a convenient spike shape, compliant with memristors technology to allow weight adjustment during the learning phase.

Wu’s neural circuit shown in Figure 1 operates in the integration and firing mode. At the integration mode, the OPAMP works as a leaky integrator, over the current, \(I_{in}\), flowing at its negative input. At the integration mode, the voltage level at the output of the OPAMP decreases until it reaches a determined threshold voltage, \(V_{thr}\). The comparator circuit compares the membrane voltage, \(V_{mem}\), with \(v_{thr}\), to generate a signal activating the Phase Control block when the descending \(V_{ref}\) reaches \(V_{thr}\). At this moment, the Phase Control block commands the Spike Generator block to initiate a spike event with a predefined waveform and it changes the control signals, \(\alpha_{fire}\) to ON state, while \(\alpha_{out}\) to OFF state. These control signals are complementary. The neural circuit is reconfigured by the current states of \(\alpha_{fire}\) and \(\alpha_{out}\). If \(\alpha_{int}\) is ON, the OPAMP works in the integration mode, if \(\alpha_{fire}\) is ON the neuron is in fire mode. During integration mode the neuron output \(V_{out}\) is set in \(V_{ref}\), at the same time, the spike generator block must hold a \(V_{ref}\) at the positive OPAMP input, which is buffered at the negative OPAMP input. In the firing mode, \(V_{out}\) is connected to \(V_{mem}\); generating a spike event, feedbacking, \(V_{mem}\) to the negative OPAMP input. At the end of the firing mode, \(C_{mem}\) is reset to a \(V_{ref}\), potential.

The equivalent model of the LI section is presented in Equation (1).

\[
\frac{dV_{mem}}{dt} = \frac{V_{p} - V_{mem}(t)}{R_{leak}C_{mem}} - \frac{I_{in}}{C_{mem}} \tag{1}
\]

Where \(V_{p}\) is the voltage objective, while the circuit is in integration mode \(V_{p}\) corresponds to \(V_{thr}\).

Wu’s circuit functioning could still be simplified to implement the proposed methodology, performing a noise signal filtering process. The simplification consists of establishing constant delays before switching states and restarting the integration phase. It imposes a period of neuron inactivity corresponding to the refractory period, seen in biological brains. This behavior is modeled as shown below:

\[
\begin{align*}
\text{if } V_{mem} &\approx V_{thr} \\
\text{then } V_{p} &\leftarrow V_{ref}, R_{leak} \leftarrow 1, I_{in} \leftarrow 0
\end{align*}
\]

Once the neuron is in the refractory period, it maintains its state for a predefined period, after which it returns to the previous (integration) state.

2.2. Tuning Curves
Since spike trains convey information through their timing and any spike-wave produced by neural circuits models are supposed to be identical (Gerstner et al., 2016), the membrane’s potential in neuromorphic circuits can be characterized simply by a list of events: \(t^{0}, t^{1}, ..., t^{n}\), where \(0 \leq t^{i} \leq T\), with \(i = 0, 1, 2, ..., n\) is the \(i\)-th spike time in an observed period \(T\) (Dayan and Abbott, 2001). Figure 2 shows a representation of this list.

A simple way to characterize the response of a neuromorphic circuit is by counting the number of peak voltages fired during the presentation of a stimulus (input current). By repeating this operation for a certain number of different stimuli, it is possible to estimate a function, \(f\), that describes the relationship between an input current, \(I_{in}\), and a frequency of spikes \(fr\) (Dayan and Abbott, 2001; Eliasmith, 2013). In this study, an alternative way to estimate the neuron frequency is proposed. Since neuromorphic circuits have no stochastic behavior, it is possible to prove that the same neuron frequency response will always be obtained for a given \(I_{in}\). Therefore, by measuring the time elapsed between the event of two spikes, \(p = t^{n} - t^{n-1}\), the frequency is obtained by using \(\frac{1}{p}\).

2.3. Mean Value Theorem for Integrals
The time between spikes in the circuit presented in Figure 1 corresponds to the mean value of the input current.

Rearranging elements from Equation (1), we find the next expression.

\[
I_{in} = \frac{V_{p} - V_{mem}(t)}{R_{leak}} - \frac{C_{mem}}{dt}dV_{mem}, \tag{2}
\]

Equation (2) corresponds to Current Kirchhoff’s Law, producing a summation of all the currents at input node, \(I_{in} = I_{R_{leak}} + I_{C_{mem}}\), where \(I_{R_{leak}}(t)\) is the current across \(R_{leak}\) which behavior is unknown in advance, thus:

\[
I_{in} = I_{R_{leak}}(t) - \frac{C_{mem}}{dt}dV_{mem}. \tag{3}
\]

Now, integrating both sides of Equation (3) with the defined time intervals limits between neural events (spike occurrences) results in Equation (4). Internal values of the neuromorphic units are reset at the end of each neural event,

\[
\int_{t_{n-1}}^{t_{n}} I_{in}dt = \int_{t_{n-1}}^{t_{n}} I_{R_{leak}}(t)dt - \int_{V_{ref}}^{V_{thr}} C_{mem}dV_{mem} \tag{4}
\]
Solving integrals on both sides:

\[ I_{in}(t^n - t^{n-1}) = \int_{t^{n-1}}^{t^n} I_{R_{leak}}(t) \, dt - C_{mem} V_{mem} \tag{5} \]

Where the value \( V_{mem} = V_{thr} - V_{ref} \) results at the end of the integration period. Equation (5) corresponds to the Mean Value Theorem for integrals (Stewart, 2018). Thus, a constant value of \( I_{in} \) exists such that applied for the time interval, \( t^n - t^{n-1} \), equals the value of the current \( I_{R_{leak}}(t) \) on the same period. Particularly, \( I_{in} \) can be seen as the mean value of current on the period \( t^n - t^{n-1} \) plus a constant value (\( C_{mem} V_{mem} \)).

3. PROPOSED METHODOLOGY FOR A NEURAL FILTER DESIGN

Our proposal consists of using the tuning curve function of the neuromorphic circuit to estimate the \( I_{in} \) value on Equation (5). The methodology proposed to use a neural circuit as a signal filter is depicted in Figure 3.

The tuning curve for the circuit introduced in Figure 4, is obtained by sweeping the current \( I_{in} \) of Equation (1) between a current interval \( \in [0, 300] \mu A \), considering the following electrical and timing parameters: \( C_{mem} = 1 \mu F \), \( R_{leak} = 10k \Omega \), and a refractory period of 10 \( \mu s \), we proceed to measure the time elapsed between potential membrane spikes. The below equation is proposed as a prototype to estimate function \( f \).

\[ f(I_{in}) = \ln(I_{in} + a) b - c \tag{6} \]

Parameters \( a = 1724.8761 \), \( b = 21.6051 \), \( c = -161.1285 \), are determined using nonlinear least squares curve fitting (Virtanen et al., 2020).

Because Equation (6) is invertible, we can take two produced spikes and calculate the current in the elapsed period between spikes. That is to say, \( f^{-1} \) computes the equivalent input current value in the system (\( I_{in} \)).

\[ f^{-1}(fr) = I_{in}(fr) = \beta \left[ \frac{b - c}{e^{\frac{b - c}{\beta}} - a} \right] \tag{7} \]

Where:

\[ fr = \frac{1}{\alpha \beta} \]
FIGURE 3 | Proposed Methodology. First, define the parameters of the neural circuit. Second, characterize the response to obtain the circuit tuning curve function, $f$, and its inverse function. Finally, stimulate the neural circuit with the noisy input signal and use the time elapsed since the last spike to evaluate the inverse function $f^{-1}$. The value computed using $f^{-1}$ is held until the next event occurs.

4. EXPERIMENTS AND RESULTS

To demonstrate the performance of our proposal, the following experiment was conducted. First, synthetic white noise is simulated to ensure a critical noise condition affectation over the clean signal with a uniform frequency distribution (Grinsted, 2022) (Figure 6). Second, the white noise is added to an arbitrary signal, refer to Figure 7. Finally, the noisy signal is used as the input for Equation (1), and the equivalent current output is computed using Equation (7). The results are shown in Figure 8.

It is possible to appreciate a significant noise reduction after the rebuilding operation. Figure 9 shows the Power Spectral Density of both original and noisy signals, and preservation of fundamental frequency is observed, thus we could conclude that the recovered signal is a good approximation of the original one. Notice that fundamental frequencies of the original signal are within the frequency range of noise.

In order to compare this proposal with other approaches, the same noisy signal was filtered using linear (Chebyshev, Butterworth, Elliptic) and digital (median Tukey, 1977) filters (Virtanen et al., 2020). Figure 10 shows the output responses obtained from these standard filters. Figure 11 shows the error measure of each filter, computed as the difference between the original and the output of the corresponding filter. Additional experiments were conducted using Gaussian Multiplicative Noise and Impulsive Random Noise, results are shown in Figures 12, 13.

To define a figure of merit for the proposed filter, we measure the distance between each filter response and the original signal (noiseless signal). As each filter introduces a different amount of time delay caused by the filtering process, Euclidean distance is not an appropriate choice since filtering techniques with a minimum delay will tend to render better results. Therefore, Fast Dynamic Time Wrapping (Salvador and Chan, 2007) (FastDTW) was used as a performance evaluation criterion. Comparisons of error results applying both Euclidean distance and FastDTW are shown in Table 1.

5. CONCLUSION

This study introduced the capacity and performance of simulated spiking neural network circuits to recognize primary signal...
FIGURE 4 | Tuning curve of the used circuit. Marks shows measurements frequencies obtained by sweeping the current $I_\ell$ of Equation (1) in interval $0 - 300 \mu A$, with electrical and timing parameters: $C_{\text{mem}} = 1 \mu F$, $R_{\text{leak}} = 10 k\Omega$, and a refractory period of 10 $\mu s$. The orange line shows approximation by 7. * refers to the measured frequency of spikes. - refers to measurements approximation made by Equation (6).

FIGURE 5 | Signal rebuild scheme. (A) Behavior of Equation (7) evaluated at each step of the simulation. (B) At each spike Equation (7) is evaluated with $\rho$ equal to the time elapsed from the last spike, and the value is preserved until the next event is reached. The dotted line marks the time of two spikes.

FIGURE 6 | Synthetic white noise generate for this study. (A) Noise signal added. (B) Fast Fourier Transform of noise. It is possible to observe that the noise has a frequency uniform distribution between 1Hz and $31.6228 \times 10^4$ Hz.
FIGURE 7 | Comparison between the original signal with the noisy signal. For this experiment, the signal was computed evaluating the function $\sin(1500t^2) \cos(50t) + 200 \times 10^{-6}$. It is important to observe that the signal must be positive at any moment.

FIGURE 8 | Comparison between signal rebuild from spike frequency on neuron output (blue line) and original signal (orange line).
FIGURE 9 | Power Spectral Density graph of the original signal and filter signal by our proposal. We can observe than primary frequencies are presented with low degradation.

FIGURE 10 | Comparison of results between different filters. (A) Median filter. (B) 3rd-order Chebyshev type 1 filter with cutoff frequency at 55kHz. (C) 3rd-order Chebyshev type 2 filter with cutoff frequency at 55kHz. (D) 3rd-order Butterworth filter with cutoff frequency at 55kHz. (E) 3rd-order Elliptic filter with cutoff frequency at 55kHz. (F) Neural filter proposed.
FIGURE 11 | Comparison of error between outputs of different filters (additive white noise) and the original signal. The ideal error signal must be 0 at any time. (A) Median filter. (B) 3rd-order Chebyshev type 1 filter. (C) 3rd-order Chebyshev type 2 filter. (D) 3rd-order Butterworth filter. (E) 3rd-order Elliptic filter. (F) Neural filter proposed.

FIGURE 12 | Comparison of results between different filters, using a signal affected by Gaussian Multiplicative Noise of 30%. (A) Median filter. (B) 3rd-order Chebyshev type 1 filter with cutoff frequency at 55kHz. (C) 3rd-order Chebyshev type 2 filter with cutoff frequency at 55kHz. (D) 3rd-order Butterworth filter with cutoff frequency at 55kHz. (E) 3rd-order Elliptic filter with cutoff frequency at 55kHz. (F) Neural filter proposed.
FIGURE 13 | Comparison of results between different filters. The signal is affected by impulsive random noise (random pulses of 5µs and 0.1mA were added). (A) Median filter. (B) 3rd-order Chebyshev type 1 filter with cutoff frequency at 55kHz. (C) 3rd-order Chebyshev type 2 filter with cutoff frequency at 55kHz. (D) 3rd-order Butterworth filter with cutoff frequency at 55kHz. (E) 3rd-order Elliptic filter with cutoff frequency at 55kHz. (F) Neural filter proposed.

TABLE 1 | Comparison between filter performance with Additive White Noise, Multiplicative Gaussian Noise, and Impulsive Random Noise.

| Filter                     | Euclidean | FDTW | MSE     | PSNR*     |
|----------------------------|-----------|------|---------|-----------|
| **Additive white noise**   |           |      |         |           |
| Median filter              | 0.257062  | 2.66536 | 4.04539×10⁻⁰⁹ | 15.6014   |
| 3rd-order Chebyshev type-1 | 0.0300501 | 0.246951 | 6.02007×10⁻¹¹ | 34.2452   |
| type-1 filter              |           |      |         |           |
| 3rd-order Chebyshev type-2 | 0.0575698 | 0.576117 | 2.20652×10⁻¹⁰ | 28.5982   |
| type-2 filter              |           |      |         |           |
| 3rd-order Butterworth filter | 0.0378712 | 0.342879 | 1.96635×10⁻¹⁰ | 30.293    |
| Elliptic filter            | 0.047365  | 0.463987 | 1.49665×10⁻¹⁰ | 32.5455   |
| Proposed Neural filter     | 0.043605  | 0.187784 | 1.26760×10⁻¹⁰ | 31.0114   |
| **Multiplicative gaussian noise** |           |      |         |           |
| Median filter              | 0.0574893 | 0.543259 | 2.20335×10⁻¹⁰ | 28.6104   |
| 3rd-order Chebyshev type-1 | 0.0278647 | 0.153496 | 5.17626×10⁻¹¹ | 34.901    |
| type-1 filter              |           |      |         |           |
| 3rd-order Chebyshev type-2 | 0.0365451 | 0.299232 | 8.90364×10⁻¹¹ | 32.5545   |
| type-2 filter              |           |      |         |           |
| 3rd-order Butterworth filter | 0.0306838 | 0.206129 | 6.27664×10⁻¹¹ | 34.0639   |
| Elliptic filter            | 0.0338538 | 0.260648 | 7.64055×10⁻¹¹ | 33.21     |
| Proposed Neural filter     | 0.0453503 | 0.170839 | 1.37110×10⁻¹⁰ | 30.6705   |
| **Impulsive Random Noise** |           |      |         |           |
| Median filter              | 0.0653834 | 0.142627 | 2.85000×10⁻¹⁰ | 27.4928   |
| 3rd-order Chebyshev type-1 | 0.0964574 | 0.172277 | 8.86096×10⁻¹¹ | 32.5664   |
| type-1 filter              |           |      |         |           |
| 3rd-order Chebyshev type-2 | 0.0488858 | 0.209080 | 1.59321×10⁻¹⁰ | 30.0186   |
| type-2 filter              |           |      |         |           |
| 3rd-order Butterworth filter | 0.0468088 | 0.160894 | 1.46071×10⁻¹⁰ | 30.3956   |
| Elliptic filter            | 0.0488396 | 0.226337 | 1.57720×10⁻¹⁰ | 30.0623   |
| Proposed Neural filter     | 0.0438345 | 0.173065 | 1.28096×10⁻¹⁰ | 30.9658   |

Highlighted values represent the best performance according to the used criterion. A lower value implies better performance (Higher value is better). The results shown in this table are the average values obtained from 15 conducted simulations.

Information from signals corrupted deliberately with noise. Our proposal works as the analog mobile mean filter (refer to Mean Value Theorem for Integrals section) minimizing digital electronics, thus reducing the required number of transistors. Our frequency base decoding scheme has proven to have a good noise rejection, specially added white noise, but maintaining good performance with other types of noise, bringing artificial intelligence closer to circuit technology to deliver innovative solutions to filter white noise with the same frequency domain as the original signal, with minimal latency and low information loss. It is also a promising approach to, i.e., the conception of future innovative lab-on-chip implementations. Increasing the signal-to-noise ratio rejection ratio, cost efficiency, and sensitivity, is essential in these devices.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

LG-S proposed, developed, programmed the neural filter code, conducted the simulation runs, and wrote the first draft of the manuscript. VP-P and HS proposed modifications to the encoding and decoding strategy architectures. ER-E reviewed the test bench for the filter experiments. JM-N helped with the neural modeling in Python. All authors contributed to the conception and design of the study, manuscript revision, read, and approved the submitted version.
FUNDING

The authors would like to thank the economic support of the projects SIP 20210124, 20221780, 20211657, 20220268, 20212044, 20210788, 20220226, and COFAA and CONACYT FORDECYT-PRONACES 6005.

ACKNOWLEDGMENTS

The authors would like to thank the support provided by Instituto Politécnico Nacional, Secretaría de Investigación y Posgrado, Comisión de Operación y Fomento de Actividades Académicas and CONACYT-México for the support to carry out this research.

REFERENCES

Abbott, L. (1999). Lapicque’s introduction of the integrate-and-fire model neuron (1907). Brain Res. Bull. 50, 303–304. doi: 10.1016/S0361-9230(99)00161-6

Butterworth, S. (1930). On the theory of filter amplifier. Exp. Wireless Wireless Eng. 7, 536–541.

Dayan, P., and Abbott, L. K. (2001). Theoretical Neuroscience. Computational and Mathematical Modeling of Neural Systems. Cambridge, MA: MIT Press.

Dupeyroux, J., Stroobants, S., and de Croon, G. (2021). A toolbox for neuromorphic sensing in robotics.

Elliasmith, C. (2013). How to Build a Brain: A Neural Architecture for Cognition, 1st Edn. Oxford: Oxford University Press.

Gerstner, W., Kistler, W. M., Naud, R., and Paninski, L. (2016). Neuronal Dynamics. Cambridge: Cambridge University Press.

Grinsted, A. (2022). fftnoise - generate noise with a specified power spectrum.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. Neural Netw. 10, 1659–1671. doi: 10.1016/S0893-6080(97)00011-7

Mead, C. (1990). Neuromorphic electronic systems. Proc. IEEE 78, 1629–1636. doi: 10.1109/5.58356

Orchard, G., Frady, E. P., Dayan Rubin, D., Sossa, F., et al. (2021). “Efficient neuromorphic signal processing with loihi 2,” in 2021 IEEE Workshop on Signal Processing Systems (SiPS) (Coimbra: IEEE).

Salvador, S., and Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. Intell. Data Anal. 11, 561–580. doi: 10.3233/IDA-2007-11508

Schuman, C. D., Potok, T. E., Patton, R. M., Birdwell, J. D., Dean, M. E., Rose, G. S., et al. (2017). A survey of neuromorphic computing and neural networks in hardware. CoRR, abs/1705.06963. doi: 10.48550/arXiv.1705.06963

Sharifzadeh, M., Burelo, K., Sarnthein, J., and Indiveri, G. (2021). An electronic neuromorphic system for real-time detection of high frequency oscillations (hfo) in intracranial EEG. Nat. Commun. 12, 3095. doi: 10.1038/s41467-021-23342-2

Stewart, J. (2018). Calculus Early Transcendentals, 8th Edn. Belmont, CA: Cengage.

Tukey, J. W. (1977). Exploratory Data Analysis. Princeton, NJ: Addison-Wesley Publishing Company.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., et al. (2020). SciPy 1.0: fundamental algorithms for scientific computing in python. Nat. Methods 17, 261–272. doi: 10.1038/s41592-020-0772-5

Wijekoon, J. H., and Dudek, P. (2008). Compact silicon neuron circuit with spiking and bursting behaviour. Neural Netw. 21, 524–534. doi: 10.1016/j.neunet.2007.12.037

Wu, X., Saxena, V., Zhu, K., and Balagopal, S. (2015). A cmos spiking neuron for brain-inspired neural networks with resistive synapses and in-situ learning. IEEE Trans. Circ. Syst. II 62, 1088–1092. doi: 10.1109/TCSII.2015.2456372

Zamarreño-Ramos, C., Camuñas-Mesa, L. A., Pérez-Carrasco, J. A., Masquelier, T., Serrano-Gotarredona, T., and Linares-Barranco, B. (2011). On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex. Front. Neurosci. 5, 26. doi: 10.3389/fnins.2011.00026

Zare, M., Zafarkhah, E., and Anzahi-Nezhad, N. S. (2021). An area and energy efficient lif neuron model with spike frequency adaptation mechanism. Neurocomputing 465, 350–358. doi: 10.1016/j.neucom.2021.09.004

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer AZ declared a shared affiliation with the authors to the handling editor at the time of review.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 García-Sebastián, Ponce-Ponce, Sosa, Rubio-Espino and Martínez-Navarro. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.