



# Event-Based Computation for Touch Localization Based on Precise Spike Timing

Germain Haessig<sup>1\*</sup>, Moritz B. Milde<sup>2</sup>, Pau Vilimelis Aceituno<sup>3,4</sup>, Omar Oubari<sup>5</sup>, James C. Knight<sup>6</sup>, André van Schaik<sup>2</sup>, Ryad B. Benosman<sup>5,7,8</sup> and Giacomo Indiveri<sup>1</sup>

<sup>1</sup> Institute of Neuroinformatics, University of Zurich and ETH Zurich, Zurich, Switzerland, <sup>2</sup> International Centre for Neuromorphic Systems, MARCS Institute, Western Sydney University, Penrith, NSW, Australia, <sup>3</sup> Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany, <sup>4</sup> Max Planck School of Cognition, Leipzig, Germany, <sup>5</sup> Institut de la Vision, Sorbonne Université, Paris, France, <sup>6</sup> Centre for Computational Neuroscience and Robotics, School of Engineering and Informatics, University of Sussex, Brighton, United Kingdom, <sup>7</sup> University of Pittsburgh, Pittsburgh, PA, United States, <sup>8</sup> Carnegie Mellon University, Pittsburgh, PA, United States

## OPEN ACCESS

### Edited by:

Emre O. Neftci,  
University of California, Irvine,  
United States

### Reviewed by:

Timothée Masquelier,  
Centre National de la Recherche  
Scientifique (CNRS), France  
Arindam Basu,  
Nanyang Technological University,  
Singapore

### \*Correspondence:

Germain Haessig  
germain@ini.uzh.ch

### Specialty section:

This article was submitted to  
Neuromorphic Engineering,  
a section of the journal  
Frontiers in Neuroscience

**Received:** 15 December 2019

**Accepted:** 07 April 2020

**Published:** 19 May 2020

### Citation:

Haessig G, Milde MB, Aceituno PV, Oubari O, Knight JC, van Schaik A, Benosman RB and Indiveri G (2020) Event-Based Computation for Touch Localization Based on Precise Spike Timing. *Front. Neurosci.* 14:420. doi: 10.3389/fnins.2020.00420

Precise spike timing and temporal coding are used extensively within the nervous system of insects and in the sensory periphery of higher order animals. However, conventional Artificial Neural Networks (ANNs) and machine learning algorithms cannot take advantage of this coding strategy, due to their rate-based representation of signals. Even in the case of artificial Spiking Neural Networks (SNNs), identifying applications where temporal coding outperforms the rate coding strategies of ANNs is still an open challenge. Neuromorphic sensory-processing systems provide an ideal context for exploring the potential advantages of temporal coding, as they are able to efficiently extract the information required to cluster or classify spatio-temporal activity patterns from relative spike timing. Here we propose a neuromorphic model inspired by the sand scorpion to explore the benefits of temporal coding, and validate it in an event-based sensory-processing task. The task consists in localizing a target using only the relative spike timing of eight spatially-separated vibration sensors. We propose two different approaches in which the SNNs learns to cluster spatio-temporal patterns in an unsupervised manner and we demonstrate how the task can be solved both analytically and through numerical simulation of multiple SNN models. We argue that the models presented are optimal for spatio-temporal pattern classification using precise spike timing in a task that could be used as a standard benchmark for evaluating event-based sensory processing models based on temporal coding.

**Keywords:** temporal coding, event-based sensors, spatio-temporal patterns, spike-based computing, touch localization

## 1. INTRODUCTION

Information transmission in neural networks is often described in terms of the rate at which neurons emit action potentials. Neurons are typically assumed to encode values—such as the orientation of a bar—using their mean firing rate, with individual spikes emitted using a Poisson process (Dean, 1981). Neurons in higher processing areas of the brain (e.g., in primary visual cortex) have been shown to demonstrate variable spike timing in response to repetitions of identical stimuli (Hubel and Wiesel, 1962). This variability is commonly interpreted as being the result of

noise (or noisy background activity) which can be assumed to be an additive signal to the sensory input one (Baudot et al., 2013). This linear separation of signal and noise has been used to justify rate- and/or population-coding by averaging across time and/or neuronal populations (Shadlen and Newsome, 1998; Dayan and Abbott, 2001). These observations led to the common assumption that the main mode of information transmission in most brain areas is encoded in the neurons average spike-frequency. This assumption, supported by many experimental investigations (Softky and Koch, 1993; Dayan and Abbott, 2001), continues to be used in the field of machine learning.

However, the time at which spikes are emitted might also carry additional information. If this is the case, the temporal-correlation of such events can then be used an extra source of information for models of computation (Dayan and Abbott, 2001; Thorpe et al., 2001). This type of signal representation is described as a *temporal code*. In the last three decades, the advantages of temporal coding have been demonstrated in computational models of fast visual processing (Thorpe et al., 2001); for the classification of time-varying signals and balance (Gütig and Sompolinsky, 2006; Deneve and Machens, 2016); in temporal interval discrimination (Buonomano and Merzenich, 1995); in state-dependent computation (Buonomano and Maass, 2009), and even for fine motor control (Laje and Buonomano, 2013). Furthermore, experimental findings have shown that the information carried in the timing of spikes can be used by the brain to discriminate textures (Hipp et al., 2006; Wolfe et al., 2008), classify temporal patterns (Mainen and Sejnowski, 1995; Wehr and Zador, 2003; Baudot et al., 2013; Goel and Buonomano, 2016) or localize an animal in its environment (O'Keefe and Recce, 1993). This evidence demonstrates that neural networks—whether biological or artificial—can use spike timing information to extract relevant cues for behavior and generate events with precise timing precision in response to time-varying input patterns (Mainen and Sejnowski, 1995).

While both rate- and temporal-codes are used to convey information in the brain, conventional ANNs, for the most part, are based only on rate-codes. The contexts and tasks in which temporal-coding can outperform rate-coding remain elusive, especially as the performance in many tasks is measured purely in terms of classification accuracy and ignores additional metrics such as latency, energy consumption and computational complexity.

In this paper, we first describe a well-constrained spatio-temporal pattern classification task inspired by the sand scorpion: localizing the source of a vibration induced by tapping on a surface, using the spatio-temporal pattern detected by an array of sensors. We then present a step-by-step analysis of conventional algorithms and five different models based on spiking neural networks for classifying the data-set of spatio-temporal patterns using both supervised and unsupervised learning rules.

## 2. BACKGROUND

### 2.1. Sand Scorpion Prey Localization

Sand scorpions, such as the specimen shown in **Figure 1A**, are nocturnal predatory arachnids which, despite their primitive

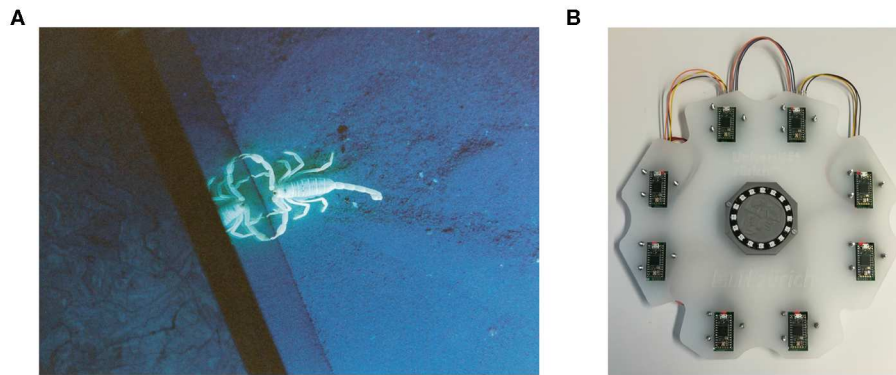
visual systems, can accurately locate prey such as crickets up to 50 cm away (Brownell, 1977). Brownell (1977) discovered that sand scorpions perform this feat using time-based computation based on two types of information propagated through the sand of their desert habitat: Transverse Rayleigh waves and compressional waves. Rayleigh waves travel slowly across the surface of the sand at a velocity of  $\approx 50 \text{ m s}^{-1}$  and are sensed by the scorpion's Basitarsal Compound Slit Sensilla (BCSS). Compressional waves diverge spherically from their source—traveling through the sand at  $\approx 150 \text{ m s}^{-1}$  and attenuating much more quickly than the Rayleigh waves (Brownell, 1977). Sand scorpions detect these waves using sensory hairs on their legs.

Both types of sensory organ are located on the ends of the scorpion's legs, maximizing the distance between the sensors and thus the difference in arrival time between signals measured at each one. While, theoretically, the arrival time of either type of wave could be used by the scorpion to detect the direction of its prey, Rayleigh waves travel and attenuate slower than compressional waves resulting in better range and larger time differences (1 ms rather than 0.4 ms). This intuition was supported by an ablation study in which Brownell and Farley (1979a) found that the BCSS was required for sensing direction.

As well as being able to detect the *direction* of their prey, sand scorpions can also estimate how far away it is. Brownell and Farley (1979b) suggested that the difference in amplitude of the signals received by the sensory hairs on different legs could be used to perform this computation. Here, the faster attenuation of the compressional waves is advantageous as it results in larger differences in amplitude between near and distant stimuli.

### 2.2. Computational Models of Spatio-Temporal Pattern Recognition

The ability to learn and recognize spatio-temporal sequences is a hallmark of biological neural information processing. Understanding spatio-temporal sequences is at the heart of object recognition, navigation and, in more general terms, all neuron-to-neuron communication. Each neuron receives a spatio-temporal pattern of pre-synaptic action potentials or spikes at its dendrites and sends output spikes to its post-synaptic partners. In the case of a single input channel, the problem of spatio-temporal sequence learning can be addressed by temporal coincidence detection (Carr and Konishi, 1990) or by temporal correlation detection (Krammer and Koch, 1997). The former approach provides binary outputs, whereas the latter approach provides a continuous output. In both cases, information is encoded in the timing of the incoming spike. On the other hand, if multiple input channels are present, spatio-temporal patterns can be represented by detecting coincidence or correlation of spikes arriving via the different input channels (Roy et al., 2016). Additionally, neurons have more options for capturing spatio-temporal patterns when multiple input channels are present. A neuron can use synaptic weight plasticity to emphasize certain channels over others, synaptic delay plasticity to delay certain input channels compared to others, or any combination of the two. To recognize spatio-temporal patterns, Gütig and Sompolinsky (2006) proposed the tempotron model in which synaptic weights are adjusted in a supervised manner, based on the deviation of the maximum (post-synaptic) voltage from the spiking threshold for wrongly



**FIGURE 1 | (A)** A sand scorpion in the lab. Image courtesy of Martin Reichert and Wolfgang Stuerzl. **(B)** Our prototype.

classified patterns. Roy et al. (2016) extended the tempotron approach by using an online structural plasticity mechanism in a competitive winner-takes-all (WTA) network relying on binary synapses. Alternatively, Izhikevich (2006) proposed a learning framework in which Spike-Time Dependent Plasticity (STDP) is used to adjust synaptic weights and synaptic propagation delays are randomly sampled at the beginning of the simulation and subsequently fixed. Both approaches lead to the learning of *polychronous* neural ensembles, each encoding a different spatio-temporal pattern. Wang et al. (2013) presented a hardware implementation of polychronous networks in which propagation delays are learned in a supervised manner, based on the expected firing time of the post-synaptic neuron. Another approach to learning synaptic delays is to sample synaptic time constants from a distribution and select relevant time constants via an STDP mechanism (Gerstner et al., 1996). Thus only synapses with fitting delays which trigger post-synaptic spikes are selected. In the next sections, we will present both biological and event-based mechanisms for synaptic and neuronal plasticity to learn spatio-temporal patterns.

### 2.3. Biological Mechanisms for Synaptic Delay Plasticity

Spikes are delivered to a neuron's post-synaptic partners through its axon with a transmission delay dictated by the axon's conduction velocity. The conduction velocity is dependent on both the diameter of the axon and the thickness of the *Myelin* sheath around it (Swadlow and Waxman, 2012). Myelin is a phospholipid substance formed by glial cells and its presence increases the conduction velocity of axons by wrapping around them and acting as an electrical insulator. Furthermore, it has recently been shown that the myelination of axons can be influenced by neural activity (Markram et al., 1997; Fields, 2015; Koudelka et al., 2016) suggesting that a form of "myelin plasticity" is also at work—something that should be taken into consideration when developing learning algorithms for spiking neural networks (Baldi and Atiya, 1994; Maass, 2001).

By optimizing conduction delays, a myelin plasticity-based model opens the way to directly learning the time dynamics of incoming spikes and extracting meaningful spatio-temporal

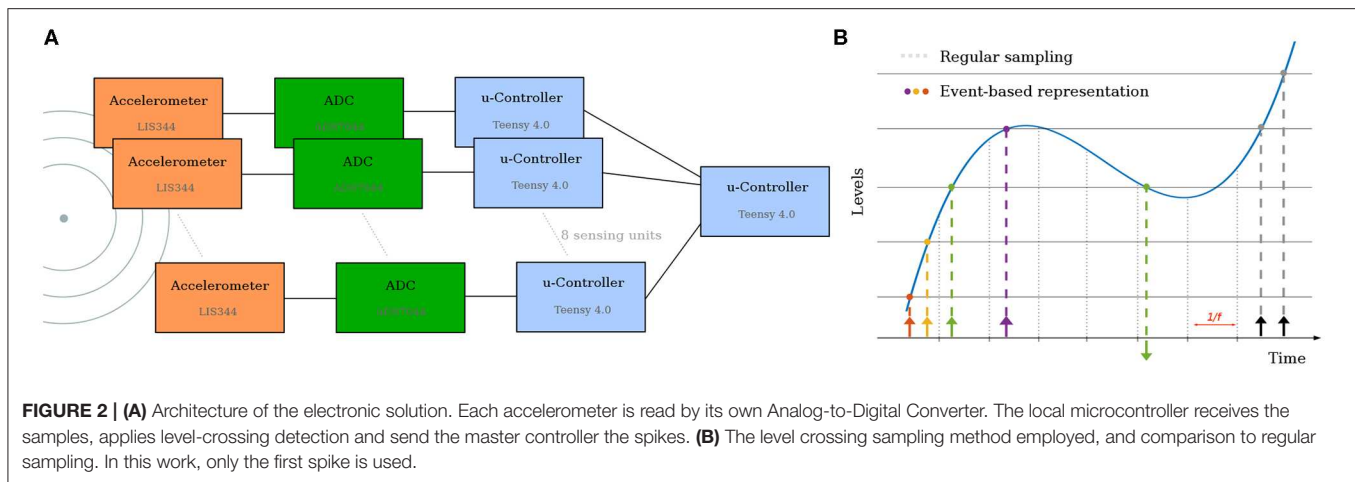
patterns. Previous conduction delay-based algorithms have not often been tested with practical tasks such as pattern recognition and clustering (Eurich et al., 1999, 2000). The DELTRON (Hussain et al., 2012) uses the tempotron model (Gütig and Sompolinsky, 2006) to adjust conduction delays through gradient descent dynamics. Paugam-Moisy et al. (2008) extended the polychronization model developed by Izhikevich (2006) to include learnable conduction delays for classification and Wang et al. (2013) applied this approach to pattern storage. Matsubara (2017) developed a probabilistic delay learning model which adjusts conduction delays and synaptic weights. However, Matsubara used this to classify time-invariant databases such as MNIST, which have no temporal structure making them a poor choice for evaluating computation based on spike timing.

### 2.4. Event-Based Spatio-Temporal Pattern Recognition

The task solved by the sand scorpion can be described more generally as spatio-temporal pattern classification and recently, two complementary approaches, specifically designed for event-based sensory signals, were proposed. Both approaches feature homogeneous and fixed synaptic time constants and adapt synaptic weights to cluster spatio-temporal patterns. In the following subsections, we detail these two approaches.

#### 2.4.1. HOTS: A Hierarchy of Event-Based Time-Surfaces

Lagorce et al. (2017) proposed an algorithm in which events are converted into a continuous-valued time surface. This approach can be understood as convolving events within a pre-defined region of interest (ROI) with an exponential decaying kernel, with the reference time being the time of the central event in the ROI. These spatio-temporal contexts are then matched to learned features using online learning, offline clustering or other methods. HOTS can be employed in a hierarchical fashion, with an increasing ROI size and time-constant for the exponential kernels and has been successfully applied to variety of classification tasks (Cohen et al., 2016; Afshar et al., 2019a).



### 2.4.2. FEAST: Event-Based Feature Extraction Using Adaptive Selection Thresholds

To guarantee that all feature detector neurons are used equally when clustering time-surfaces, Afshar et al. (2019b) extended HOTS to feature coupled, adaptive thresholds. Every time a given feature detector emits a spike, its threshold is increased. This results in non-updated feature detectors being more likely to capture the next time-surface and means that all feature detector neurons are equally active across the data set<sup>1</sup>. If no feature detector captures the present time-surface, however, all thresholds are decreased. The adaptation of firing thresholds can be understood as a homeostatic plasticity mechanism (Turrigiano and Nelson, 2004; Qiao et al., 2016, 2017). In the context of continual learning, this “global”<sup>2</sup> threshold adaptation might prevent convergence if unrecognized patterns are common (Afshar et al., 2014).

## 3. METHODS

### 3.1. Neuromorphic Tactile Sensor Design

The problem of spatially localizing a stimulus on a 2D surface is well-defined with 5 sensors (Mahajan and Walworth, 2001; Hu and Yang, 2010). However, having an array of more than 5 sensors adds robustness to the system, so we developed the prototype shown in **Figure 1** with (arbitrarily) 8 sensors. A circular configuration of the sensor array would lead to badly conditioned cases—as depicted by Mahajan and Walworth (2001)—so our 8 sensors are arranged in the non-circular manner shown in **Figure 1B**. An acrylic plate makes a rigid connection between the 8 sensors. As the system is statically overconstrained (5 redundant contact points), a slightly flexible acrylic plate was chosen to ensure that all 8 sensors could still touch the surface if there was any fabrication misalignment.

Each sensing unit consists of a Piezoelectric accelerometer for sensing vibrations and a local microcontroller-based processing unit (Teensy 4.0, ARM Cortex-M7) which reads samples from

the Analog to Digital converter at 1 MHz, and then applies a level crossing detection to generate events (Astrom and Bernhardsson, 2002) (**Figure 2B**). All 8 sensors then transmit these events to an additional central processing unit which solves the analytical problem using the approach described in section 3.2.1 and saves the data for dataset creation (**Figure 2A**).

While in desert sand, a 1 ms resolution would be sufficient (Brownell, 1977), in order to work on more common mediums—which typically have faster propagation speed—we need higher temporal accuracy. Depending on the surfaces used in our experiments, a wave propagates at a speed between 200 and 300 m s<sup>-1</sup>, which result in a propagation time between 1 and 1.5 ms in our setup, between radially opposite sensors. However, standard accelerometers with digital output are limited to sampling rate of only a few kHz, so we decided to use an accelerometer with an analog output (*STMicro LIS344*), combined with a separate 1 MHz Analog to Digital converter (*Texas Instruments ADS7044*).

**Figure 2** illustrates the architecture of the electronic solution, as well as the spike generation method. This approach of fast sampling followed by level-crossing detection was chosen for its flexibility (different encoding schemes could be tested). However, other approaches such as the one introduced by Lee et al. (2017), or the VLSI event-generator proposed by Corradi and Indiveri (2015) could be used. While these might reduce the complexity of the sensing unit and (possibly) increase the time precision, this would come at the cost of reduced flexibility.

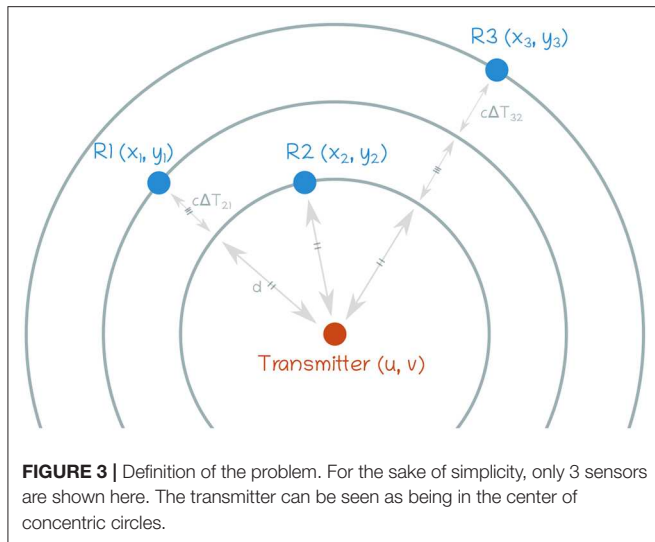
Using this sensor, we recorded a dataset consisting of 10 repetitions of 32 stimuli (8 different angles, every 45°, and 4 distances (200, 400, 600, 800 mm)). The stimuli, i.e., the surface vibrations, were induced by tapping with the index finger on a wooden surface.

### 3.2. Algorithms

In the following section we will present five different solutions to the problem of localizing the position of a vibration induced by tapping on a rigid surface. Not all approaches are entirely successful but, nonetheless, we hope to provide interesting concepts and ideas which try to emphasize how to extract task relevant information from the timing of incoming events.

<sup>1</sup>The occurrences of examples in the data set need to be balanced.

<sup>2</sup>Global in this context refers to the population of feature detectors at a given level in a hierarchy, not across a hierarchy of feature detectors.



**FIGURE 3** | Definition of the problem. For the sake of simplicity, only 3 sensors are shown here. The transmitter can be seen as being in the center of concentric circles.

We selected these algorithms to represent varying levels of complexity and biological plausibility as well as because they each require different amounts of information. Specifically, in section 3.2.1 we will first demonstrate how to localize the position of the tap analytically if the geometry of sensory array and the propagation speed are known. Then in sections 3.2.2 to 3.2.6, we will present more and more biological plausible implementations which try to solve the task with less and less external information.

While these algorithms do not represent a complete list of possible solutions, we still hope to provide the reader with a thorough analysis of several approaches for computation based on the precise timing of spikes as well as outlining some of the challenges the community needs to overcome to perform such computation using event-driven SNNs. More importantly however, we hope to provide a starting point for the development of novel algorithms, as well as providing a benchmark task for further comparison and evaluation.

### 3.2.1. Analytic Solution

The position of the source can be estimated based on the Time Difference Of Arrival (TDOA) between each pair of sensors. The 2D problem is shown in **Figure 3** and, given the sensor spatial positions  $R_i(x_i, y_i)$  and the TDOA for each pair of sensors, the source localization  $(u, v)$  and the propagation speed  $c$  in the chosen medium can be retrieved as follows:

$$\underbrace{\begin{bmatrix} x_1 - x_2 & y_1 - y_2 & -\Delta T_{12} & -\Delta T_{12}^2 \\ x_1 - x_3 & y_1 - y_3 & -\Delta T_{13} & -\Delta T_{13}^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1 - x_n & y_1 - y_n & -\Delta T_{1n} & -\Delta T_{1n}^2 \end{bmatrix}}_A * \underbrace{\begin{bmatrix} u \\ v \\ cd \\ c^2 \end{bmatrix}}_X = \underbrace{\begin{bmatrix} x_1^2 + y_1^2 - x_2^2 - y_2^2 \\ x_1^2 + y_1^2 - x_3^2 - y_3^2 \\ \vdots \\ x_1^2 + y_1^2 - x_n^2 - y_n^2 \end{bmatrix}}_B \quad (1)$$

for  $N$  sensors where  $A \in \mathbb{R}^{N \times 4}$ ,  $X \in \mathbb{R}^{4 \times 1}$  and  $B \in \mathbb{R}^{N \times 1}$ . This equation can then be solved using the pseudoinverse  $A^+$  of  $A$ . Because  $A^+$  has to be evaluated every time a stimulus is presented, we can exploit the fact that our matrices are well defined [rank( $A$ ) being equal to the number of columns of  $A$ ] and therefore:

$$A^+ = (A^T A)^{-1} A^T \quad (2)$$

allowing us to minimize the required computation and find an analytic solution to the problem, using bloc decomposition for the inverse  $(A^T A)^{-1}$ , given the fact that  $A^T A$  is a squared matrix ( $A^T A \in \mathbb{R}^{4 \times 4}$ ). An alternative approach would be to iteratively estimate the pseudoinverse, following the method described by Tapson and van Schaik (2013).

### 3.2.2. Temporal Coincidence Detection

A simple way to detect a particular position is to have a neuron associated with every target position, connected to each receptor with delayed synapses. In this set-up, each neuron receives one spike from every receptor and must only spike if the input came from the right place. The natural way to ensure that the receptive neuron will indeed cross the threshold is to have the spikes arrive at the same time, so that all the incoming spikes coincide and create a large increase in membrane potential.

Specifically, a decoding neuron at position  $p$  has  $N_s$  synapses, each with a corresponding delay  $d_p(k)$ , and parameters  $\tau$  and  $\theta$ , corresponding to the decay constant and the firing threshold of the neuron. For all input spikes to arrive simultaneously, we must associate the vibration wave generate at each position  $p$  to a delay vector  $\mathbf{d}_p \in \mathbb{R}^{N_s}$ . The sub-threshold membrane potential of the decoding neuron is then

$$v_p(t) = \sum_{k=1}^{N_s} e^{-\frac{t-t(k)-d_p(k)}{\tau}} \Theta(t - t_k - d_p(k)) \quad (3)$$

where the exponential corresponds to the decay of the membrane potential and  $\Theta$  is a step function that ensures that the input is only relevant after it arrived at the detector neuron at time  $t(k) + d_p(k)$ , where  $t(k)$  is the time of arrival of the ground vibration at the detector  $k$  and  $d_p(k)$  is the delay associated with synapse  $k$ . The leaky integrate-and-fire neuron will spike, indicating a stimulus at position  $p$ , if

$$v_{\max} > \theta, \quad v_{\max} = \max_t v_p(t). \quad (4)$$

Hence, we will try to maximize the value of  $v_{\max}$ . Since the exponential decay term in Equation (3) implies that each input spike is strongest at its arrival so, if we want to maximize the membrane potential, we must make sure that all those spikes arrive simultaneously, which can be achieved by setting

$$d_p(k) = d_p^* - t_p(k) \quad (5)$$

where  $t_p(k)$  is the  $k$ th entry of the vector  $\mathbf{t}_p \in \mathbb{R}^{N_s}$  which corresponds to time of arrival of the ground vibration from position  $p$  to each sensor  $k$ , and  $d_p^*$  is a value that ensures that

all the transmission delays are positive. Naturally, there are many possible values of  $d_p^*$ , but for simplicity we will set

$$d_p^* = \max_k t_p(k), \quad (6)$$

which implies that all the spikes arrive when the last spike is detected—as it is impossible to receive all of them earlier than that.

In the ideal case—where the spikes from a given position are perfectly timed— $\theta = N_s$ , so that, when the spikes arrive exactly at  $t_k$ , they will all add up and the membrane potential will cross the threshold. However, if the spike times vary even slightly, the fast decay will result in a sub-threshold membrane potential and the neuron would not fire.

In the real world, the ideal case is unlikely so we must account for the possibility of jitter by associating a target detector with an area rather than a point. Assuming that  $\tau$  is fixed, we must simply select the value of  $\theta$  that minimizes our classification error

$$e = \Pr[v_{\max} < \theta|p] + \Pr[v_{\max} > \theta|\neg p], \quad (7)$$

which is simply the sum of the probabilities of false negatives and false positives. The simplest way to do this is to realize that  $\Pr[v_{\max} < \theta|p]$  increases monotonically with  $\theta$ , while  $\Pr[v_{\max} > \theta|\neg p]$  decreases monotonically with  $\theta$ . Hence, the computation of the optimal  $\theta$  from a sample of  $m$  examples, where  $m_p$  examples were from position  $p$  and  $m_{\neg p}$  were not, can be done through a simple algorithm:

```

for all  $m$  examples do
  compute  $v_{\max}$ 
  if example position =  $p$  then
    add the tuple  $(v_{\max}, d = 1)$  to list  $L$ 
  else
    add the tuple  $(v_{\max}, d = -1)$  to list  $L$ 
  end if
end for
Sort  $L$  by  $v_{\max}$ , high to low.
 $e \leftarrow m_p$ 
for every tuple in  $L$ : do
   $e \leftarrow e - d$ 
  Add the tuple  $(v_{\max}, e)$  to list  $M$ 
end for
select the tuple with lowest  $e$  in  $M$ 
Set  $\theta \leftarrow v_{\max}$  from the tuple with lowest error

```

This approach gives us a simple way of using the leaky integrate-and-fire nature of neurons to achieve the desired detection as long as we can compute the appropriate delays *a priori*.

### 3.2.3. Complex Weights and Delays

The previous approach, while fundamentally correct, requires precise knowledge of the delays. If sensors or synapses have systematic measurement errors or there is significant jitter, it could be impossible to find delays  $d_p(k)$  that would be able to fully compensate for the effects of noise. Furthermore, unreliable sensors or synapses should be given less importance than if perfect noiseless sensors or synapses. In this section we present

a statistical method for computing the delays and associated weights to address this issue (State, 2019).

First we must redefine our leaky integrate and fire neuron model, described in Equation (13), to include synaptic weights:

$$v_p(t) = \sum_{k=1}^{N_s} w_p(k) e^{-\frac{t-t(k)-d_p(k)}{\tau}} \Theta(t - t_k - d_p(k)), \quad (8)$$

While our new synapses now have two parameters ( $w_p(k)$  and  $d_p(k)$ ), the logic from the previous section remains the same and our goal is to force spikes to arrive as synchronously as possible. In order to manipulate the spike times algebraically, we encode the input spike train – here consisting of a single spike per neuron – into  $N_s$  variables that can be studied using linear algebra. We do this by encoding spikes as phases of a complex number so each spike

$$s(k) = e^{\frac{j\pi(t(k)-t_0)}{T}} \quad (9)$$

where  $t_0$  is the time at which the first spike of a wave arrives (so that the time of the input wave is not considered) and  $j$  is the imaginary unit. Encoding time in the phase of a complex number is a known trick when dealing with spikes, often used in *phasor* networks (Hirose, 1992; Reichert and Serre, 2013; Frady and Sommer, 2019).  $T$  is the maximum time during which we can receive spikes and is given by

$$T = \frac{2r_{\max}}{c} \quad (10)$$

where  $c$  is the wave speed and  $r_{\max}$  is the radius of the sensors, meaning that the numerator is two times the maximum distance between two sensors. The value of  $T$  ensures that the phase of  $s(k)$  is in the interval  $[0, \pi]$ , which is necessary to avoid geometric inconsistencies (State, 2019).

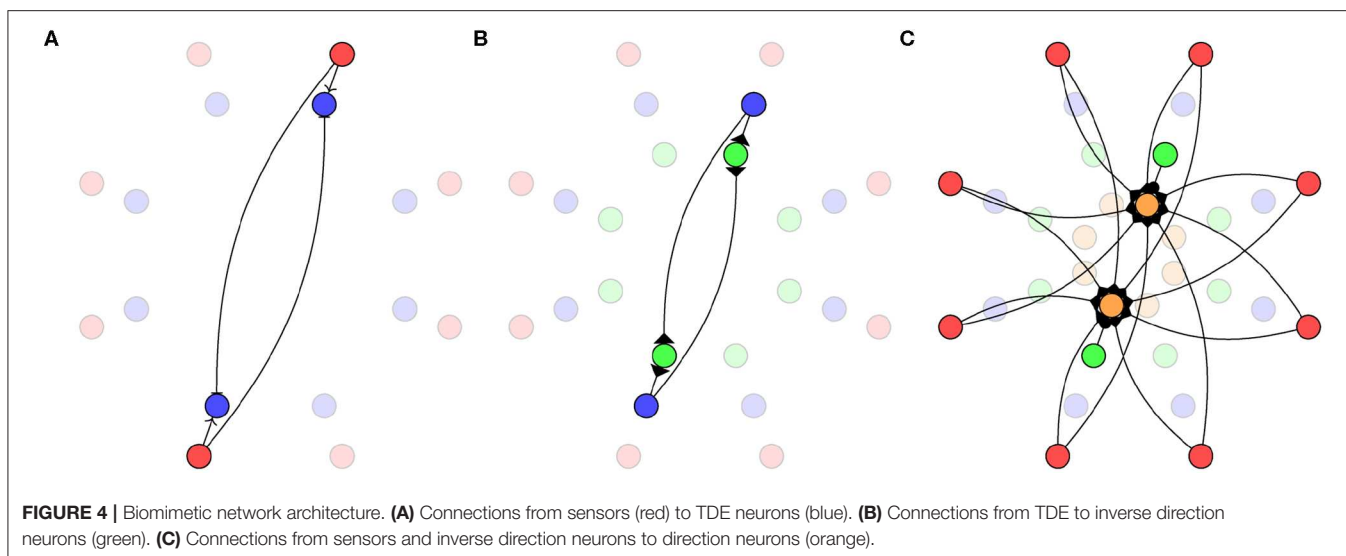
Now we can use least squares regression to obtain the delays and weights associated to each synapse. Thus, for every position  $p$  we will have

$$\epsilon = \frac{1}{N_e} \sum_{i=1}^{N_e} \|y_i - \sum_{k=1}^{N_s} \hat{w}_p(k) s_i(k)\|^2 \quad (11)$$

where  $N_e$  is the number of examples, indexed over  $i$  and  $y_i$  corresponds to the desired output of the perceiving neuron: one if the spikes were generated by a tap at position  $p$  and zero if the spikes come from a tap somewhere else.

Once we find the weight vector for position  $p$ ,  $\hat{w}_p = [\hat{w}_{p(1)}, \hat{w}_{p(2)}, \dots, \hat{w}_{p(n)}]$ , it will give us weights with complex entries. Naturally, this is not something we can put on a synapse, but rather a complex number that somehow relates  $s$  to its appropriate synapse. To obtain the delays and weights, we inverse the operation done in Equation (9) and obtain the delay from the phase and the weight from the absolute value,

$$\hat{w}_p(k) \rightarrow w_p(k) = |\hat{w}_p(k)|, \quad d_p(k) = \frac{T}{\pi} \arg \hat{w}_p(k). \quad (12)$$



It is worth noticing that the conversion from a complex weight to a weight and a delay used here ensures that all the weights are positive. This means that all synapses are excitatory, and is a natural consequence of the encoding chosen originally in Equation (9).

To understand this procedure, it is useful to look at value of  $\epsilon$ . When  $y_i = 1$ , the sum of the input to the neuron  $\sum_{k=1}^n \hat{w}(k)s_i(k)$  should be real and positive whereas, when  $y_i = 0$ , it should be close to zero. In the ideal case,  $\hat{w}(k)$  will have exactly the same phase as  $s_i(k)$  but the opposite sign, meaning that the product  $\hat{w}(k)s_i(k)$  must be real and positive and the phases somehow uniformly distributed in  $[0, 2\pi]$  when  $y_i = 0$  so that  $\sum_{k=1}^n \hat{w}(k)s_i(k)$  adds up to zero. Just as the delays were converted into phases in Equation (9), the phases must now be converted back into delays so that, when the phases of  $\hat{w}_p(k)$  and  $s(k)$  cancel each other, the delay of the synapse also cancels out the delay of the spike. The weights are also easy to interpret: the more reliable the value  $s(k)$  is for a certain position  $p$ , the higher  $|\hat{w}_p(k)|$ . This is because the least squares regression will “learn” that every time  $s(k)$  has a specific value and the product  $\hat{w}_p(k)s(k)$ —which is already real and positive due to the phase cancellation—should approach  $y = 1$  and hence be large.

As in the previous section, this complex conversion trick is simply a way to synchronize the arrival of spikes at the neuron encoding position and therefore we still need to compute the  $\theta_p$  for every neuron, for which we can, again, use Algorithm 3.2.2. It is also worth noticing that using the complex formulation intrinsically assumes that the spikes have the shape of a cosine, as opposed to a decaying exponential (Reichert and Serre, 2013; State, 2019), meaning that it is more appropriate to use a non-instantaneous synapse such as an EPSP (Takagi, 2000) with a flat value at the maximum such that the first derivative is the same; however, this does not affect our results.

The advantage of this approach compared to similar complex formulations (Reichert and Serre, 2013; Shrestha and Orchard, 2018) lies in the use of classical linear algebra. Besides being very

data efficient—as a single example would yield a solution just as well as a combination of examples—this approach easily handles cases where spikes are unreliable (State, 2019), something that is often difficult when using delays directly and it is resistant to over-fitting because the pseudoinverse guarantees that the weights will have the lowest possible modulus. However, its simplicity also make it less flexible, as it does not deal with multi-spike problems (Taherkhani et al., 2015; Shrestha and Orchard, 2018) nor does it work for SNNs with hidden units (Hirose, 1992; Frady and Sommer, 2019) as the linear algebra solution requires specific values as outputs, rather than step-by-step error feedback.

### 3.2.4. Temporal Difference Encoders

The approaches presented in the previous sections encode target position using individual neurons to represent each point in space. While this encoding allows for precise localization, it requires a large number of neurons. In this section, we will take inspiration from the ring-like neural structures present in the sand scorpion (Stürzl et al., 2000) to develop an alternative solution based on Temporal Difference Encoder (TDE) neurons (Milde et al., 2018) which requires many fewer neurons<sup>3</sup>.

In this model, each pair of opposite sensors is connected to an inner ring of 8 TDE neurons (Milde et al., 2018) as shown in **Figure 4A**. The sub-threshold behavior of the TDE neurons is modeled as a leaky integrate-and-fire neuron:

$$\tau_m \cdot \frac{dV(t)}{dt} = E_L - V(t) + R_m \cdot I(t) \quad (13)$$

<sup>3</sup>The TDE was originally named the spiking Elementary Motion Detector (sEMD) as it was designed to extract relative motion cues from spatially adjacent pixels of an event-based vision sensor. However, the computation performed by these neurons is of much more general nature as it calculates the temporal correlation of two events based on the difference in timing, irrespective of the sensory modality.

where  $E_L$  denotes the resting potential of the neuron,  $R_m$  is the membrane resistance,  $I(t)$  is the injected current at time  $t$ , and  $\tau_m$  is a decay constant. These neurons are then driven by an input current  $I(t)$  such that:

$$I(t) = \begin{cases} I_{trig} \cdot f & \text{if } f > 0 \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

where  $f$  represents a dimensionless “facilitating” input and  $I_{trig}$  represents the “trigger” input current. Both  $f$  and  $I_{trig}$  are exponentially shaped such that:

$$\tau_{syn} \cdot \frac{df}{dt} = -f \quad \tau_{syn} \cdot \frac{dI_{trig}}{dt} = -I_{trig} \quad (15)$$

where  $\tau_{syn}$  is the time constant of the synaptic dynamics. When a spike is received at a facilitating or trigger synapse, the synaptic weight ( $w_{fac}$  and  $w_{trig}$  respectively) is added to the appropriate input:

$$f \leftarrow f + w_{fac} \quad I_{trig} \leftarrow I_{trig} + w_{trig} \quad (16)$$

The dynamics described by Equations (14)–(16) result in an input current which is scaled non-linearly depending on the time difference between spikes arriving at the facilitating and trigger synapses. Additionally, as  $I_{trig}$  is “gated” by  $f$ , these synapses are also direction-selective. The time difference between the inputs received at the two opposite sensors will be largest when a stimulus is located on the line connecting them and smallest when it is on the line perpendicular to this meaning that the inner ring of TDE neurons will encode the direction of the stimuli as a vector in an over-complete 8-dimensional space.

While the directional information required could be decoded from the activity of the TDE neurons, the desired output for this system is a ring with a single active neuron identifying the direction of the stimuli. In order to achieve this, we connect the TDE neurons to a second ring of “inverse direction” neurons using the excitatory connections shown in **Figure 4B**. Weak connections from the TDE neuron to the adjacent inverse direction neuron and strong connections to the opposite inverse direction neuron result in this population of neurons having a minimum of activity in the direction of the stimulus.

Finally, the inverse direction neurons are connected to the innermost ring of “direction neurons” with inhibitory connections as shown in **Figure 4C**. These neurons are additionally provided with background excitation—direct from the sensors—tuned to produce a “1-hot” encoding of the stimuli direction. In order to maximize the accuracy of this encoding, we use 16 neurons in this ring with inhibitory weights calculated as:

$$w_{ij} = w_{peak} \cdot \max(0, \cos(\theta_i - \phi_j)) \quad (17)$$

where  $w_{peak}$  is the peak inhibitory weight,  $\theta_i$  is the angle of the sensor adjacent to inverse direction neuron  $i$  and  $\phi_j$  is the angle of the direction neuron  $j$ . While this approach does not currently provide an estimate of distance, if magnitude information were available, this could be provided in place of the excitatory input to the direction ring.

### 3.2.5. Synaptic Delay Plasticity

So far we addressed how one can adjust neuronal ( $\Theta$ ), axonal and synaptic ( $w, \tau$ ) parameters if the geometry of the sensor array is known or, in the absence of that information, if a set of training examples is given. Given this information, we have outlined how these parameters can be optimized so that one can localize the position of a vibration source even in the presence of temporal jitter. While we used biologically motivated neuron and synapse models to perform the *computation*, the *optimization* of the parameters was done offline using conventional regression algorithms such as the least square method. Such optimization procedures require non-local information<sup>4</sup> such as the neuronal firing thresholds of other neurons, the onset of the stimulus and the position of the stimulus itself. The decision on where the tap originated from is being made through adapting neuronal firing/spiking thresholds such that the designated spatio-temporal pattern is matched. In the subsequent sections we are going to address the localization task by applying three constraints on our model:

1. Only information which is local to a given pre-post synaptic neuron pair is used to update synaptic parameters.
2. No a priori knowledge of the sensory system is required.
3. The model parameters must be updated in an unsupervised manner.

Drawing inspiration from the myelin plasticity discussed in section 2.3 and from previous work on delay shifts (Hussain et al., 2012; Wang et al., 2015), in this section we will extract temporal features by modulating conduction delays.

The proposed model uses gradient descent dynamics to synchronize spikes emitted by pre-synaptic neurons, by adjusting delays on the most recently active synapses within an experimentally set temporal window. Whenever a neuron fires, mutual inhibition is used to ensure that neurons specialize to a particular temporal pattern.

The delay plasticity model works in conjunction with leaky integrate-and-fire (LIF) neurons described in Equation (13). We chose an exponential excitatory post-synaptic current (EPSC) shape such that the input current  $I(t)$  at time  $t$  is:

$$I(t) = I_{inj} \cdot \sum_i w_i \cdot e^{-\frac{t-s_i}{\tau_{syn}}} \cdot \mathcal{H}(t) \quad (18)$$

where  $I_{inj}$  is the injected current every time a neuron fires,  $w_i$  is the synaptic weight of synapse  $i$ ,  $\tau_{syn}$  is the synaptic time constant, and  $\mathcal{H}(t)$  is the Heaviside step function.

When we study the dynamics of a single synapse  $i$ , we remove the discontinuities caused by the input signal by focusing on the range  $[s_i, t]$  where  $\mathcal{H}(t) = 1$ ,  $s_i$  being the time of arrival of a spike to a post-synaptic neuron. Assuming initial conditions such that  $V_i(s_i) = E_L$  as we are restricting the network to only one spike per synapse, and the membrane potential is reset between

<sup>4</sup>Local information is defined as to have direct access from a neuron perspective, e.g. post-synaptic density (estimate of synaptic weight), average firing rate (calcium concentration) or its own spiking threshold.



each training example through a WTA algorithm, the membrane equation now has a solution:

$$V_i(t) = E_L + \frac{R_m \cdot I_{inj} \cdot w_i \cdot \tau_{syn}}{\tau_m - \tau_{syn}} \cdot \left( e^{-\frac{t-s_i}{\tau_m}} - e^{-\frac{t-s_i}{\tau_{syn}}} \right) \quad (19)$$

The time course of the potential follows a bi-exponential model with a finite rising time. In order to maximize the membrane potential of a post-synaptic neuron—and ultimately associate it with a particular temporal pattern—we compute the gradient of the neuron's potential  $\frac{\partial V(t, s_i)}{\partial s_i}$  and modulate  $d_i$  until all spikes are aligned. The model assumes only one spike per synapse. The partial derivative of  $V(t, s_i)$  with respect to  $s_i$  can then be written as:

$$\frac{\partial V(t, s_i)}{\partial s_i} = \frac{R_m \cdot I_{inj} \cdot w_i}{\tau_m - \tau_{syn}} \cdot \left( e^{-\frac{t-s_i}{\tau_m}} - e^{-\frac{t-s_i}{\tau_{syn}}} \right) \quad (20)$$

The delay update rule can be represented by the following equation:

$$d_i^{t+1} = d_i^t + \eta \cdot \frac{\partial V(t, s_i)}{\partial s_i} \quad (21)$$

where  $\eta$  represents the learning rate of a neuron, with  $\eta > 0$ . We decay the learning rate across iterations to avoid large gradient steps.

### 3.2.6. Structural Plasticity

In this section we propose a neurally implemented, self-organizing, balanced network of excitatory and inhibitory (EI) neurons which fulfills the constraints outlined at the beginning section 3.2.5 by combining event-based STDP (Song et al., 2000) with a variant of structural plasticity (Bekkers, 2011; Knoblauch et al., 2014) and adaptive spiking thresholds (Afshar et al., 2019b). It has been demonstrated that the combination of STDP and a competitive EI network (i.e., WTA), in the context static inputs can account for disparity selectivity (Chauhan et al., 2018), in the context of non-static inputs account for the observed development of orientation selectivity (Masquelier, 2012), and even the formation of temporal memory (Kappel et al., 2014). These mechanisms become especially powerful when sensory information is encoded using relative latency, i.e., using a temporal code.

The EI network consists of  $N$  neurons of which 80% are excitatory and 20% are inhibitory<sup>5</sup>.  $N$  depends on the desired spatial resolution the network should be able to decode. Each excitatory neuron is connected via simple alpha synapse (Rall, 1967) with the following dynamics

$$I_{syn}(t) = \bar{I}_{syn} \frac{t - t_0}{\tau_{syn}} \exp\left(1 - \frac{t - t_0}{\tau_{syn}}\right) \quad (22)$$

<sup>5</sup>The excitatory neuron pool has  $N \times k$  incoming excitatory plastic synapses (see Equations 24 and 26), whereas the connectivity probability  $E \rightarrow I$  and  $I \rightarrow E$  is set to 0.7 with random and fixed synapses.

where  $I_{syn}$  is the EPSC,  $\bar{I}_{syn}$  is the peak EPSC and  $\tau_{syn}$  is the synaptic time constant. Given a pre-synaptic spike at  $t_0$ ,  $I_{syn}$  is updated as follows

$$I_{syn} = I_{syn} + w, \quad (23)$$

where  $w$  is the synaptic weight which is modified according to a STDP protocol:

$$\Delta w = \begin{cases} a_w^+ \cdot e^{-\frac{t_{pre}-t_{post}}{\tau^+}}, & \text{if } t_{pre} \leq t_{post} \\ a_w^- \cdot e^{-\frac{t_{pre}-t_{post}}{\tau^-}}, & \text{if } t_{pre} > t_{post}, \end{cases} \quad (24)$$

where  $a_w^+$  and  $a_w^-$  represent the magnitude of increments and decrements to the weight and can be seen as the learning rate of the plasticity mechanism.  $t_{pre}$  and  $t_{post}$  are the times at which the pre- and post-synaptic neuron emitted a spike and  $\tau^{+/-}$  defines the temporal window within which spikes result in weight changes. Such “additive” STDP updates (Song et al., 2000) often result in bimodal weight distributions with all weights ending up either at 0 or their maximum value. There are numerous approaches to addressing this problem (Goodhill and Barrow, 1994; Morrison et al., 2008), but here we implement an event-driven weight decay which is triggered whenever the post-synaptic neurons emits a spike. The weight decay depends on the synaptic weight and is calculated as follows:

$$w = w - (w \cdot \kappa \cdot \eta), \quad (25)$$

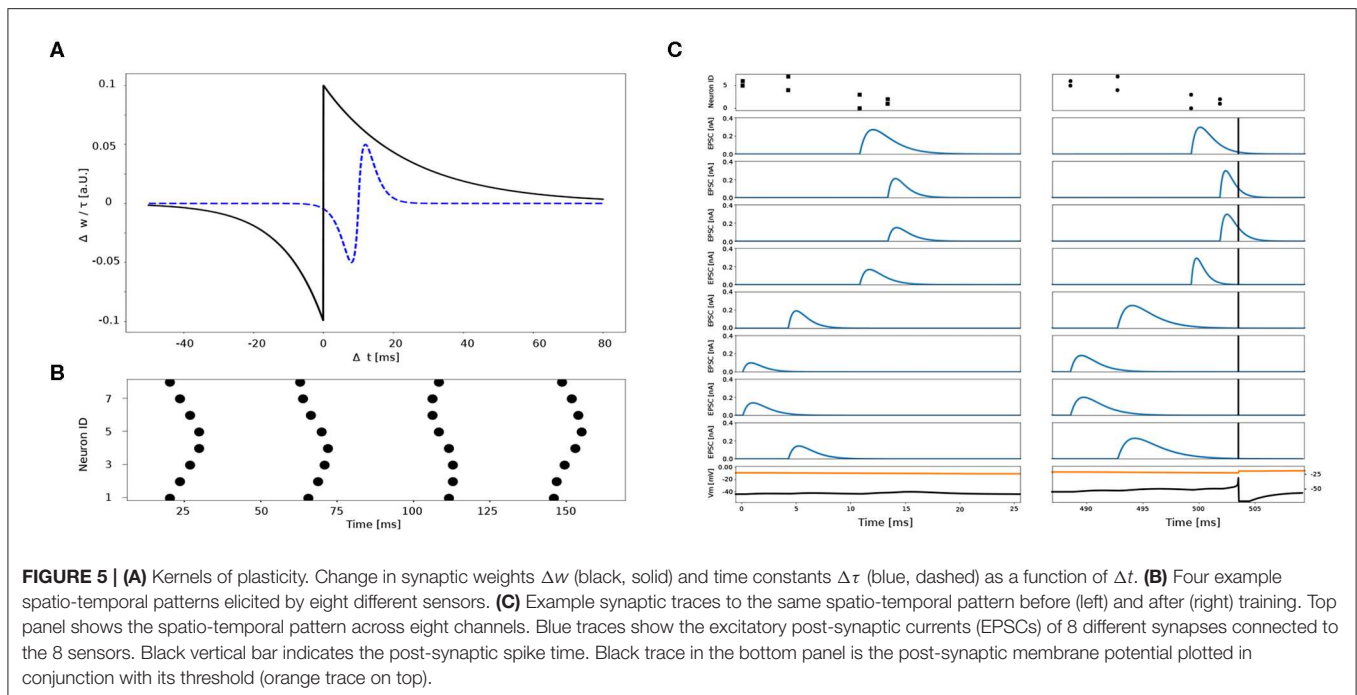
where  $\kappa$  is the ratio between weight increment and decrement ( $\kappa = \frac{a_w^+}{a_w^-}$ , for balanced STDP protocol  $\kappa = 1$ ) and  $\eta$  is the decay rate ( $\eta \ll 1$ ). The objective of the STDP learning rule is to detect spatio-temporally correlated activity in the input spike trains. Here we pair STDP for synaptic weights with a STDP rule for synaptic time constants. This plasticity rule aims to find a set of synaptic time constants which, given the post-synaptic activity, increase the overlap in synaptic input currents across the 8 input channels. The time constants of channels which transmit spikes early in the sequence are increased, whereas the time constants of channels which transmit spikes late in the sequence are decreased. This plasticity rule has the consequence that a neuron spikes as early as possible to a given spatio-temporal pattern given the provided competition of the other neurons in the EI network. The synaptic time constants  $\tau$  are updated as follows:

$$\Delta \tau = a_{\tau_{syn}} \cdot \frac{(\Delta t - s) \cdot e^{-\left\| \frac{\Delta t - s}{\tau^* - s} \right\|}}{(\tau^* - s) \cdot e^{-1}}, \quad (26)$$

where  $a_{\tau}$  is the learning rate of the synaptic time constant and is set such that the time scale of changes in the synaptic time constants is much slower than the time scale of weight plasticity ( $a_{\tau} \ll a_w^{+/-}$ ).  $\Delta t$  is calculated based on pre- and post-synaptic spike timing ( $\Delta t = t_{pre} - t_{post}$ ),  $\tau^*$  determines the peak in change of the time constant with relative to the offset  $s$  ( $\tau^* > s$ ).

The plasticity kernel for synaptic weights and time constants are depicted in **Figure 5C**.

The continuous changes to synaptic weights and time constants are combined with a form of structural plasticity,



operating on slower time-scale. If the average weight of all incoming synapses is below a user defined threshold, this rule deletes all (8) synapse and “re-spawns” a new set with a random weights and time constant sampled from a Gaussian distribution. This can be interpreted as modeling the retraction of a dendritic branch and its replacement by another at a different location hence its different time constant and strength (Zito et al., 1999). In this way, the synaptic weights and time constants are sampled and afterwards adjusted according to Equations (24) and (26) until each excitatory neuron within the EI network finds a unique set of  $w$  and  $\tau_{syn}$ . Thus, each neuron is sensitive to particular spatio-temporal pattern which, in the case of this task, represents a particular location (see **Figure 5C**). To prevent each neuron from learning multiple patterns we install adaptive thresholds, similar to (Afshar et al., 2019b). However, the adaption of the firing threshold depends only on the post-synaptic membrane potential, which reflects the networks activity indirectly via the recurrent connections.

## 4. RESULTS

In the following subsections, we will present the results obtained using each of the algorithms discussed in section 3.2.

### 4.1. Analytical Solution

Because the matrix  $A$  in Equation (1) is not constant, we need to compute its pseudo-inverse at run-time in order to use the approach presented in section 3.2.1. While it would be possible to perform this calculation on the microcontroller (either direct pseudo-inverse computation, or iterative methods), it is beyond the scope of this paper. Instead, we performed the calculation of this pseudo-inverse

offline on the host computer. Due to the temporal resolution of our system, the positional error (**Figure 6A**) is significant (73.4% accuracy) while angles are recovered reliably in almost all trials (99.7%). Nonetheless, 80% of the position error is less than 20 mm.

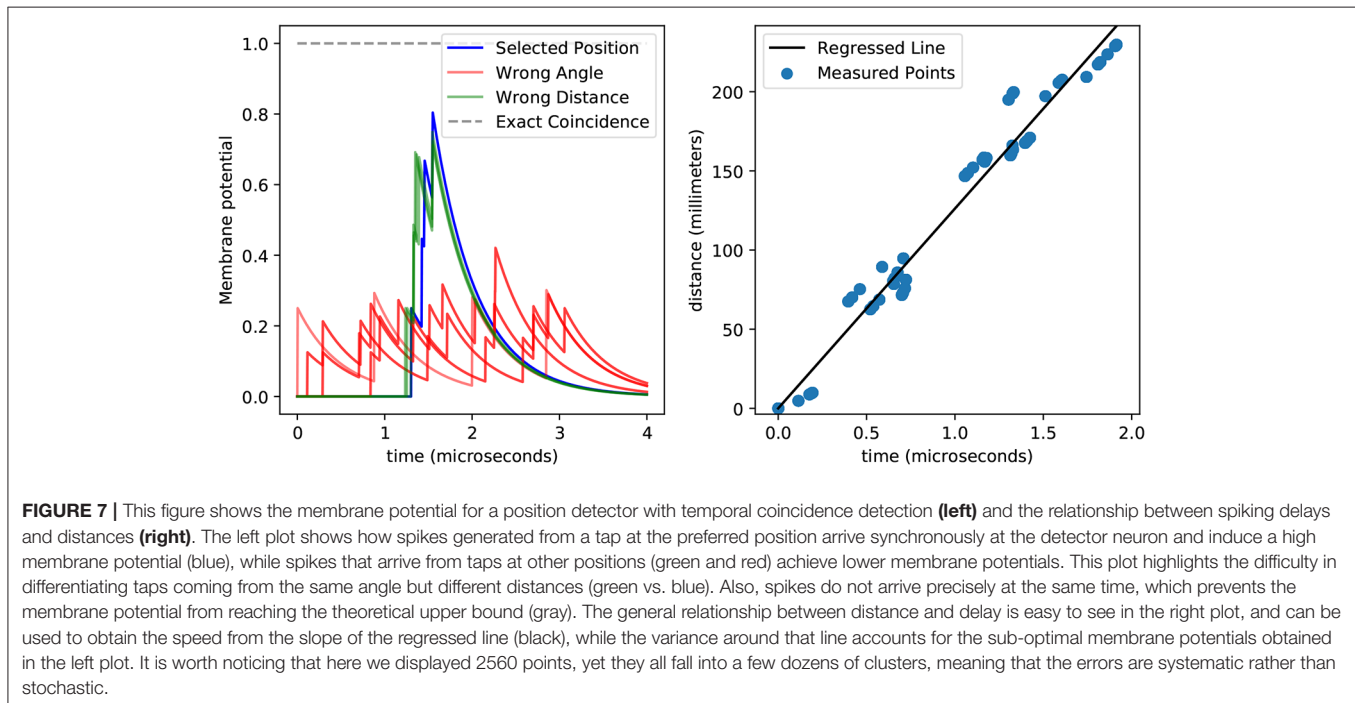
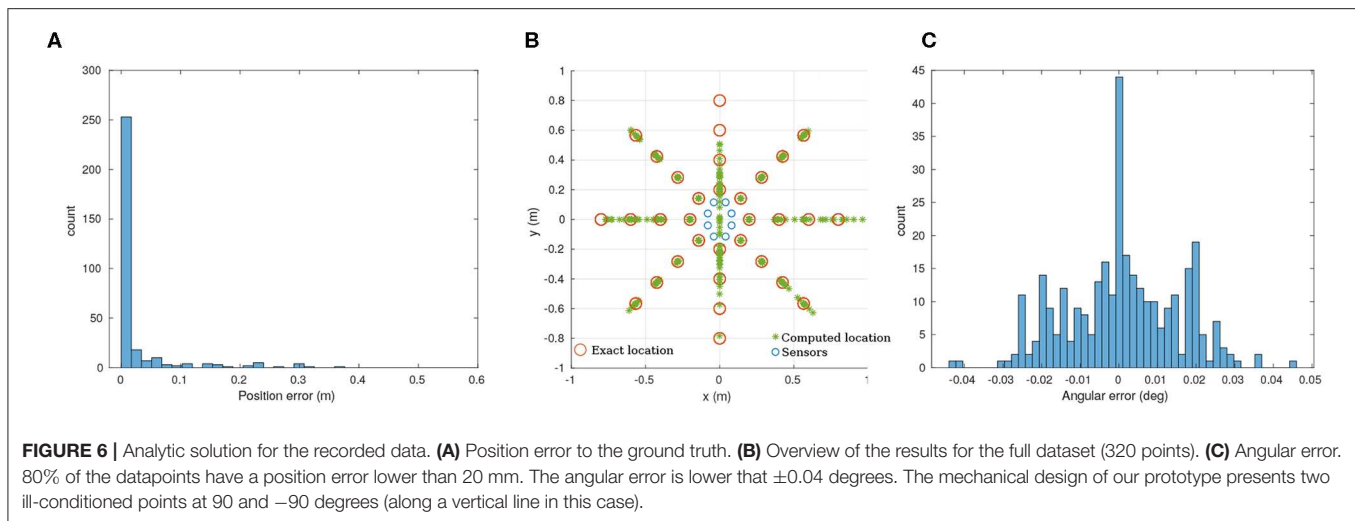
After differentiating Equation (1), it appears that our mechanical implementation has 2 ill-conditioned points, at 90 and  $-90$  degrees, as shown in **Figure 6B**. As **Figure 6C** illustrates, the angular precision is not affected by this problem.

### 4.2. Temporal Coincidence Detection

To compute the delays associated with every point and sensor, we first need to calculate the propagation speed of the vibration wave on wood. The simplest way to do this is to use linear regression on the distances between points and sensors as one variable and the arrival times as the second. From our recordings we get 2.560 spike times which, as we show in **Figure 7**, we can match to their corresponding delay. This gives us a speed of  $126 \text{ m s}^{-1}$ , which we can then use to compute the delays.

By using the delays calculated by dividing the distance by the propagation speed we find that we only able to recover 37.5% of the positions successfully. Within those errors, the angles are always perfectly recovered, but the distances are not. Although the distance recovery is better than chance—which would be 25%—the fact that the coincidence detection would mistake different distances implies that there are errors that we are not accounting for. This is not surprising given the variance around the regression line shown in **Figure 7**.

Interestingly, the percentage of errors remains the same when we change the number of measurement per point, hinting that the error in delays is not stochastic but rather systematic. This



can be observed in **Figure 7**, where despite having 2,560 points we only see a few of them, meaning that the measurements are systematically biased.

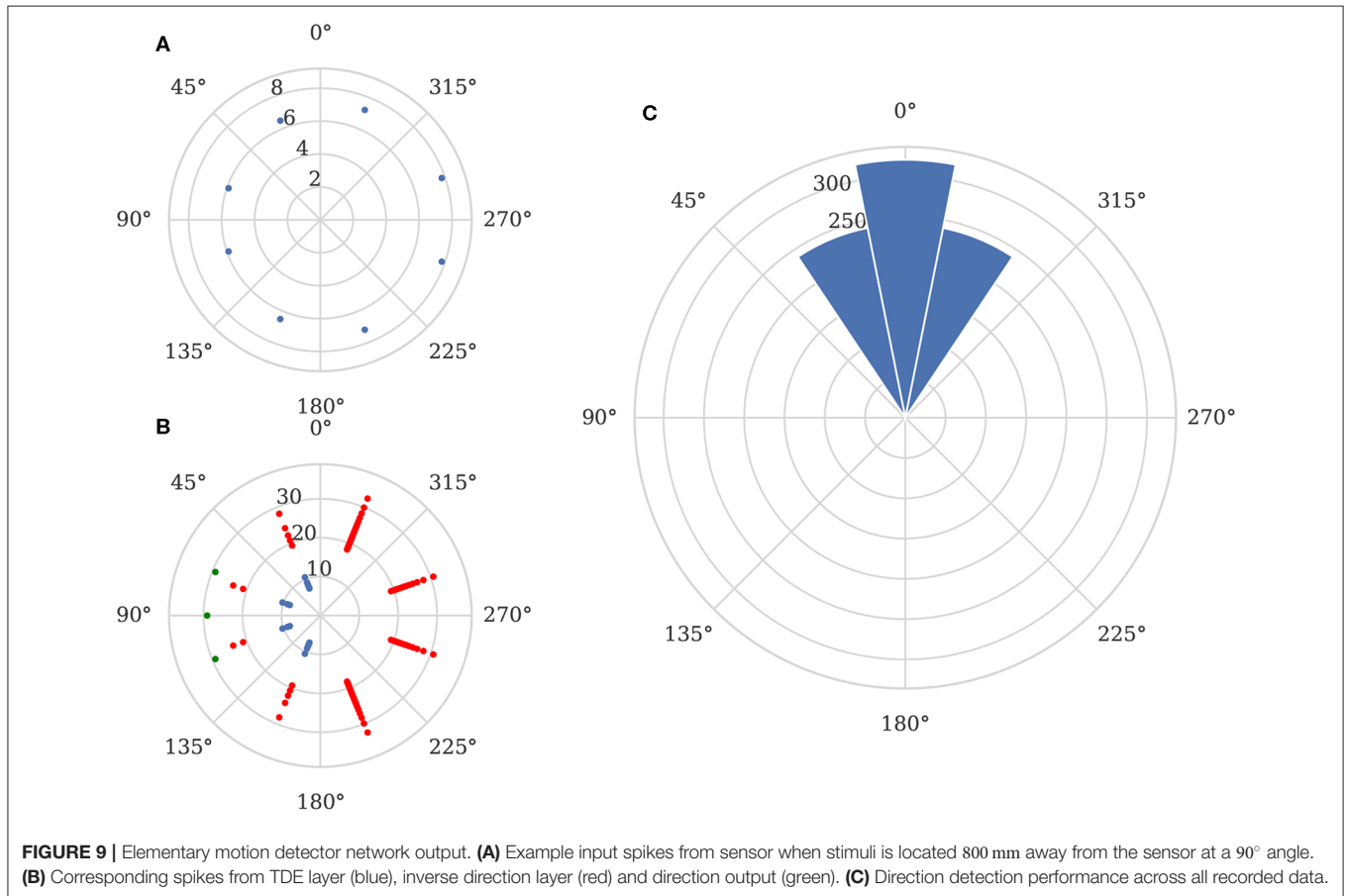
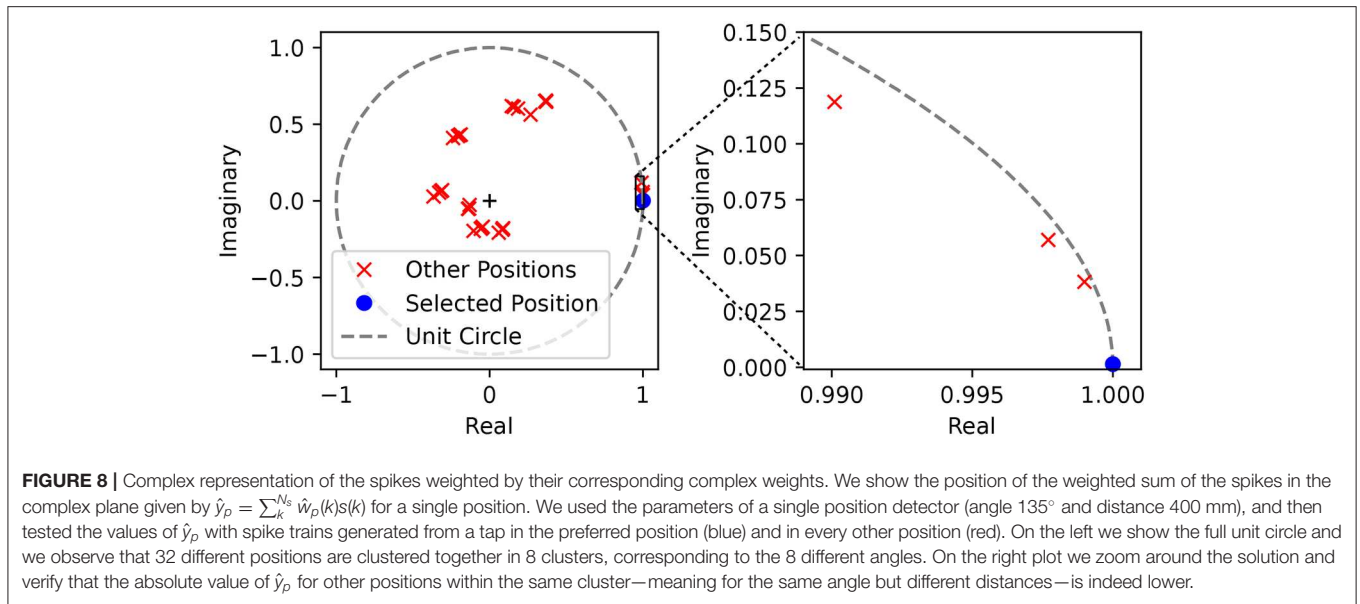
### 4.3. Complex Weights and Delays

Knowing that there is a systematic but unknown bias on the sensors implies that we must resort to techniques drawn from statistics and machine learning rather than purely analytical solutions which could extract the information of the biases automatically. As expected, using the linear regression in the complex domain yields perfect recovery of the points, implying that the systematic biases in the recorded times are not necessarily an impasse.

This can be illustrated in **Figure 8**, where we see that the value obtained by the cumulative weighted representation of the spikes in the complex plane reaches the unit circle only when the right input is presented. This can be interpreted in terms of spikes by saying that for any input spike train that does not correspond to the right input the complex representation of the spikes do not have their phases aligned, and therefore they do not arrive to the perceiving neuron at the same time.

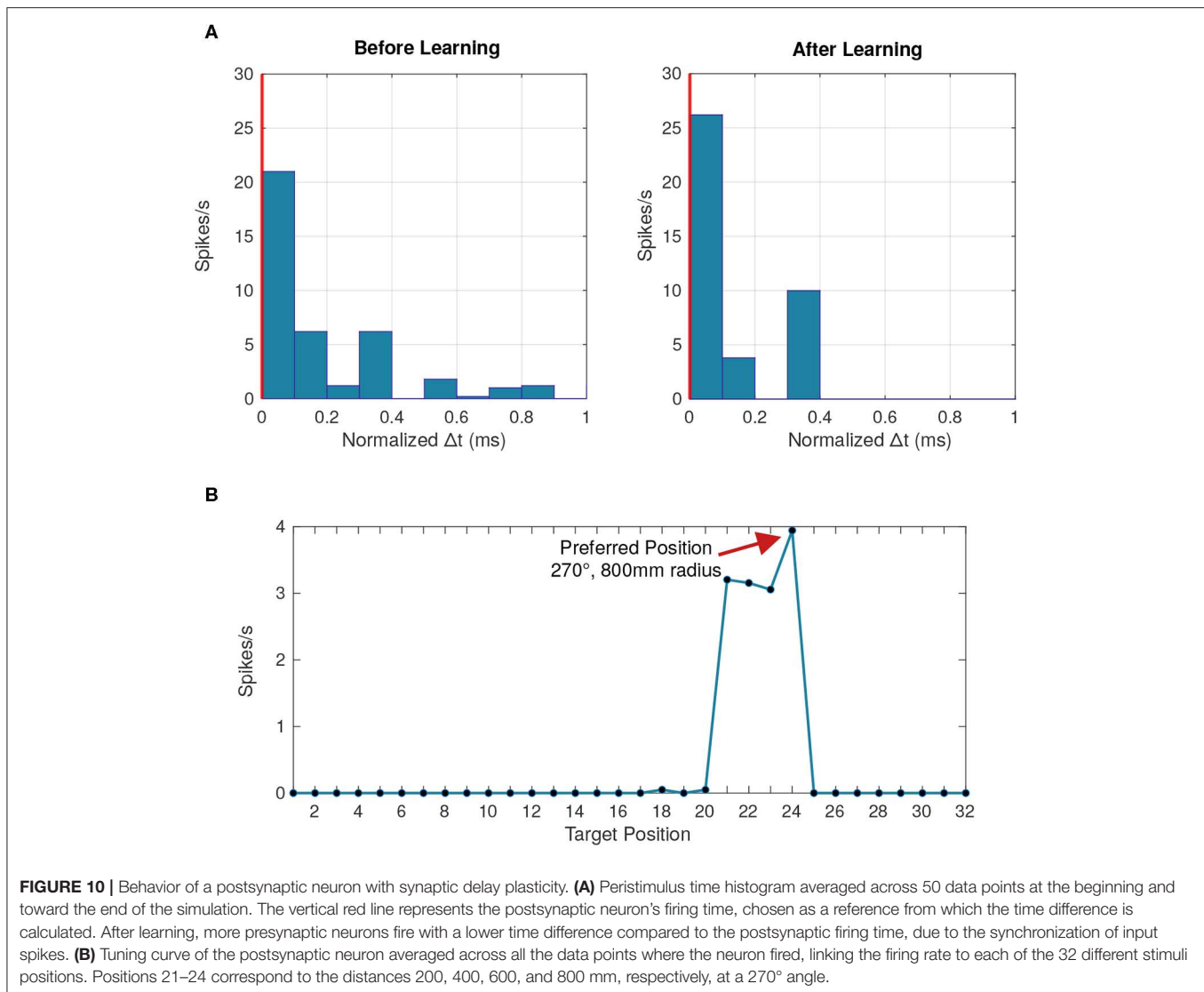
### 4.4. Temporal Difference Encoders

As the network described in section 3.2.4 is static, we simply presented the 320 sets of input spikes to the network as shown



in **Figure 9A**. Because this approach is only capable of finding the *angle* to the target, we can simply treat each output spikes from a “direction” neurons (green spikes in **Figure 9B**) as a “vote”

for the target being at the direction neuron’s corresponding angle. We can then subtract the correct stimuli angle from the angle associated with each spike and plot the circular



histogram shown in **Figure 9C**. This shows that the mean error of the classified directions is  $0^\circ$ , although a perfect one-hot encoding is not achieved resulting in a circular standard deviation of  $17.5^\circ$ .

Further investigation supports the existence of systematic biases in the sensor as, although the network is entirely symmetrical, the circular standard deviation is  $0^\circ$  for stimuli presented at angles of  $0^\circ$  and  $180^\circ$  whereas, for stimuli presented at all other angles, the circular standard deviation is  $18.5^\circ$ .

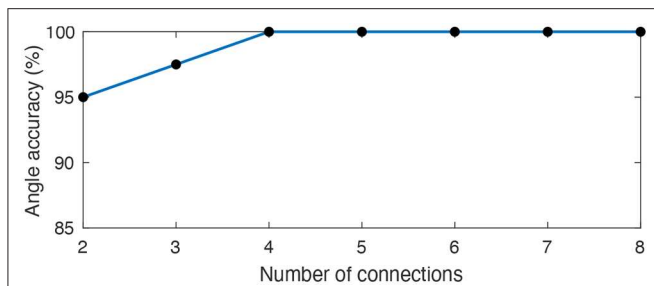
We used the GeNN library (Yavuz et al., 2016) to generate optimized CPU simulation code for this model. This simulation can be run  $10\times$  faster than real-time on a single ARM Cortex A57 core running at 2 GHz when using a 0.1 ms simulation time step, suggesting that this approach could be used for embedded online processing of spatio-temporal patterns.

#### 4.5. Synaptic Delay Plasticity

The synaptic delay plasticity network consists of eight presynaptic neurons, sparsely connected in a random fashion to

50 LIF neurons. Sparsity is achieved by limiting the number of connections toward a LIF neuron  $N = 4$ . Synaptic delays are randomly initialized according to a normal distribution with a mean of  $\mu = 0.5$  ms, and a standard deviation  $\sigma = 0.3$  ms and a fixed weight equal to  $\frac{w_0}{N}$  with  $w_0 = 1$ . The resting potential is set to  $E_L = -70$  mV. The LIF neuron's decay constant is set to  $\tau_m = 2$  ms, and the injected current  $I_{inj}$  is set to 180 nA to make sure that each presented spike train is capable of causing a LIF neuron to fire. The learning rate starts at  $\eta = 1$  and decays by 10% after every 100 input spike trains to help the network converge toward a local minimum.

Each post-synaptic neuron that responds starts specializing to a particular pattern by synchronizing its input spikes through a change in synaptic delays following Equation (21). The winner-take-all mechanism ensures that no other post-synaptic neurons synchronize their input spikes. With each subsequent presentation of the pattern, the time differences between input spikes gradually converge toward zero (**Figure 10A**).



**FIGURE 11 |** Average angle accuracy for an increasingly sparse synaptic delay network. We connected each LIF neuron to a random subset of 8 pre-synaptic neurons representing the hardware sensors. We varied the size of the subset across 35 different trials in order to assess the smallest number of connections between pre-synaptic and LIF neurons capable of preserving an accurate temporal representation of all 8 angles. A sparse synaptic delay network with only 2 connections per LIF can already represent all angles with an average accuracy of 95% and we can achieve 100% accuracy with only 4 connections.

With a temporal resolution of 0.1 ms, all eight angles were successfully represented by at least one LIF neuron (**Figure 10B**). Previous implementations of learned delays relied on an all-to-all connectivity scheme (Hussain et al., 2012), but we obtained similar performance with fewer connections through a randomly connected sparse network with high redundancy (**Figure 11**). The learned temporal patterns for a network with only two connections per LIF neuron are enough to represent all angles with an accuracy of 95% and we can achieve 100% accuracy with only four connections per LIF neuron. An all-to-all network would work just as well, but, in addition to being more efficient in terms of hardware, a randomly connected and highly redundant sparse network increases robustness against systemic noise or a faulty sensor.

We also wanted to determine whether the synaptic delay network could differentiate between stimuli at different distances. As seen in **Figure 10B**, an individual neuron seems to respond more frequently to a particular position. We expect the membrane potential to be maximized for a particular distance which was not the case as the membrane potential was similar across all distances.

While the delay plasticity network managed to specialize neurons to all directions, due to the slow attenuation of the waves being measured, the temporal signatures across the measured distances are not significantly different. Spike synchronization seems to have a limited impact on the membrane potential beyond a certain level of synchronization. An inhibitory plasticity rule could be explored to further specialize post-synaptic neurons to increasingly precise temporal patterns.

## 4.6. Structural Plasticity

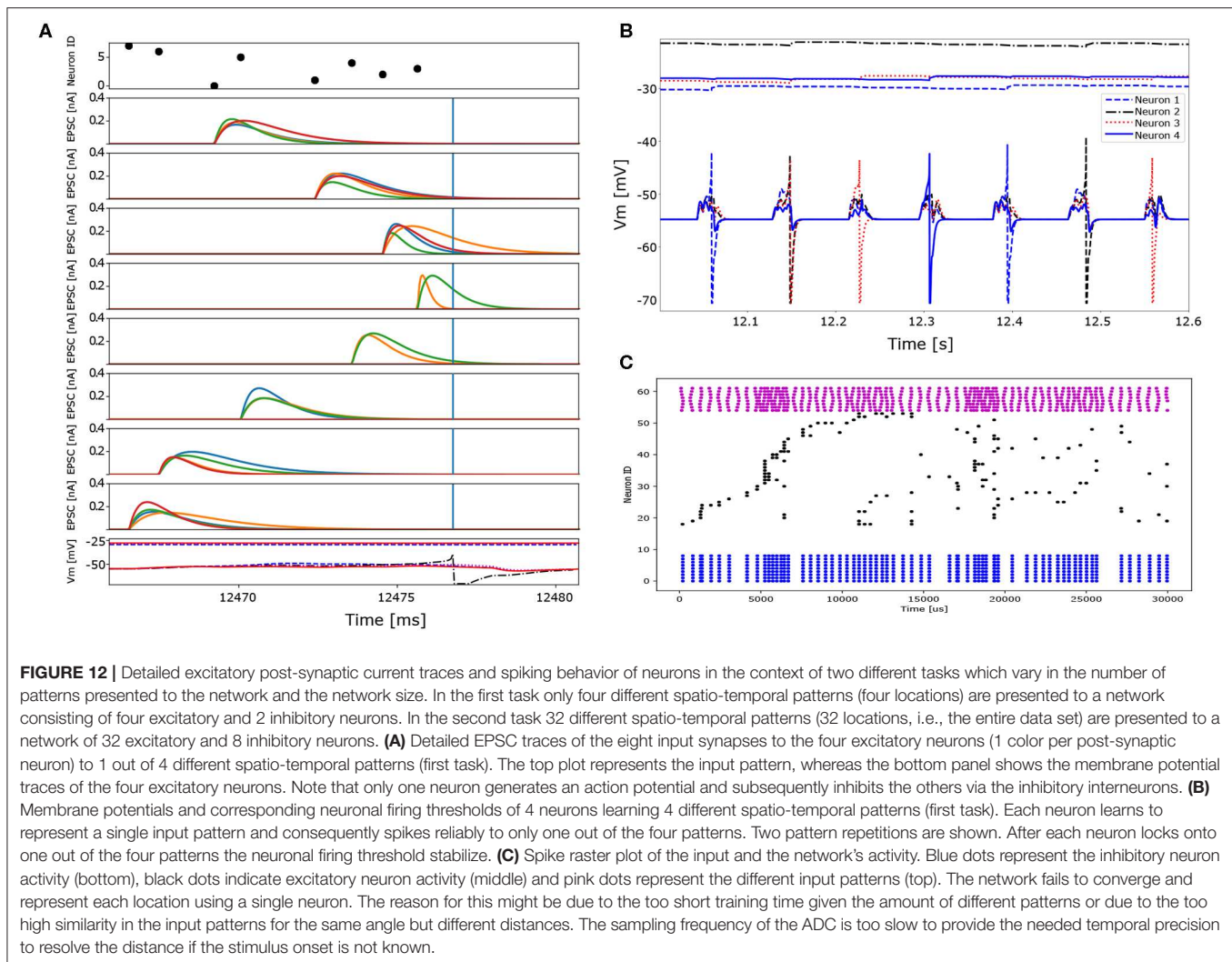
As a first step, we trained 4 neurons to learn 4 out of the 32 different spatio-temporal patterns (see **Figure 6B**). The network's free parameters, i.e., synaptic weights  $w$ , synaptic time constants  $\tau$  and firing thresholds, are randomly initialized at the beginning of the training. Each spatio-temporal pattern, corresponding to a unique location, is presented 20 times to the network. In

the beginning, the neurons sparsely capture incoming spatio-temporal patterns and therefore the thresholds slowly decrease. Each neuron starts to “lock on” to one particular pattern by decreasing the synaptic time constants of late spikes in the sequence and increasing the time constants of spikes early in the sequence following Equation (26) (for visualization of the time constant change see **Figure 6A**, blue dashed trace). The synaptic weights start to increase more for late spikes in the sequence, than for early spikes following Equation (24) (for visualization of the weight change see **Figure 6A**, black solid trace). Therefore the neuron begins to respond earlier to a given spatio-temporal pattern, while the mutual inhibition introduces competition on both the spike itself and the neuronal firing thresholds. After a given neuron's threshold—and thus its other free parameters—starts to stabilize, it reliably spikes in response to a particular pattern (see **Figure 12A**). After 15 stimuli presentations, the thresholds start to stabilize and each neuron locks onto 1 out of 4 different patterns (**Figure 12B**).

In a second step, we use 32 neurons and present all 32 patterns corresponding to different locations. Each pattern is presented 100 times to the network. The neurons fail to respond reliably to the different spatio-temporal patterns which is due to the jitter present in the data, leading to too similar spatio-temporal patterns. While neurons are capable of learning these patterns, the jitter prevents the stabilization of the firing threshold and the synaptic weights and time constants keep on changing (see **Figure 12C**). This case of failure might also be due to the number of presentation needed by the network to learn a unique set of parameters which scales non-linearly with the number of patterns to learn and number neurons in the network (see Afshar et al., 2014 for statistical analysis of this relation).

Structural plasticity mechanisms or variants thereof within competitive EI networks has been demonstrated before to learn spatio-temporal patterns of activity in static (Gerstner et al., 1996; George, 2018) and time-varying (Masquelier, 2012; Roy and Basu, 2016; Roy et al., 2016) conditions. Unlike Roy et al. (2016), our approach does not need a reference time, but rather relies on relative latency encoding similar to Masquelier (2012). The proposed unsupervised structural plasticity algorithm is designed to operate on time-continuous, event-based sensory data in which there exist no start- or end-point to a pattern, nor one can rely on batch-training. In contrast to the feature extraction approach proposed by Afshar et al. (2019b)—which inspired this work—neurons in our network adapt their neuronal firing threshold solely based on locally available signals. We do so by utilizing the inhibitory interneuron population which indirectly signal the presence of captured spatio-temporal pattern by other neurons by hyperpolarizing the non-spiking excitatory neurons.

Understanding the computational properties and emergent network dynamics resulting from recurrent excitation and inhibition mediated balanced activity is beyond the scope of this paper, but will be subject of future investigations. A promising next step would be to learn the temporal relations of different spatio-temporal patterns by exploiting recurrent excitatory synapses with STDP, as described in Kappel et al. (2014) and Milde (2019).



## 4.7. Comparison and Extensions

In the preceding sections of this work, we proposed several approaches for tackling the problem of spatio-temporal pattern classification, in the context of touch localization based on the precise timing of input events. **Table 1** shows an overview of the presented results. It is worth noting that, as a community, we lack clear metrics for assessing the performance of spiking networks. While a simple accuracy metric can be used, it fails to consider factors such as power consumption and suitability for real-time simulation as well as not reflecting constraints present in both biological SNNs and neuromorphic hardware (Nowotny, 2014). Further effort will have to be done by the community to overcome this problem and provide datasets and metrics which *do* consider these factors.

## 5. DISCUSSION

In this paper we demonstrated, through a simple task, different approaches for tackling spatio-temporal pattern classification with SNNs. The problem of separating spatio-temporal

**TABLE 1 |** Comparison for the hereby proposed methods.

Method	Angle accuracy (%)	Distance accuracy (%)
Analytic solution	99.7	73.4
Temporal coincidence	100	37.5
Complex weights and delays	100	100*
Temporal difference encoders	100	N.A.
Synaptic delay plasticity	100	N.A.
Structural plasticity	100	N.A.

*If all the approaches are able to distinguish between the arrival angles, the distance is still an open issue here. The analytic solution is able to (almost) correctly extract the distance, under the assumption of a known geometry. Some thoughts about this problem are detailed in the Discussion section (\*). The 100% accuracy for the distance in the Complex weight and delay method is under the assumption of using an extra linear classifier to process the output data.*

patterns into prototypical features or discrete classes by learning, clustering or any other form of transformation resides at the core of both event-driven computing and event-based neuromorphic processing (Chicca et al., 2014;

Indiveri and Sandamirskaya, 2019). This work is not intended to demonstrate high-precision computing, but rather to open new perspectives on learning these spatio-temporal patterns and on performing event-based tactile sensory processing. The presented algorithms were chosen to provide a qualitative overview given certain constraints of available information on how to extract task-relevant information from the timing of incoming events. Despite their different complexities, all of these approaches extract the required information solely from the precise timing of the incoming events. While a major drawback of all of these approaches is the need for temporal precision on the sensory side, our experiments expand on how information can be extracted from the timing of an incoming event in neurally-inspired processing paradigms.

As discussed in section 2.1, in sand, compressional waves attenuates rapidly with distance ( $G(d) = \frac{1}{d}$ ) meaning that the gradient of attenuation across the scorpion's outspread legs can be used to estimate distance to the stimuli. However, on the surface used in this work, attenuation is lower. Therefore, it seems unlikely that our system would sense a difference in amplitude between the sensors. Nevertheless, *mean* amplitude across the sensors could be used to estimate distance, although it would be unable to disambiguate between a distant stimuli with a large amplitude and a nearby stimuli with a smaller amplitude.

Although it is true that the case of multiple sources is not addressed in this work, we would highlight that the precision of the sensors is of a few microseconds, meaning that the vibration should have to be generated at two sources exactly at the same time, which is unlikely. We can, however, speculate that in the case of multiple sources, the methods with excitatory synapses only—such as complex weights—should promote the activation of the neurons corresponding to the two sources, while those with lateral inhibition—such as synaptic delay plasticity—would resolve a conflict one way or the other, giving one active source at a time.

Due to the nature of the stimuli, all of the approaches presented in this paper require simulations with high temporal resolution. While small models requiring high temporal resolution—such as the TDE-based approach discussed in section 4.4—can be simulated in real-time using simple CPU-based simulations, for larger models many current approaches are not capable of providing high temporal resolution and real-time simulation speed.

The majority of digital neuromorphic systems (Furber et al., 2014; Merolla et al., 2014; Davies et al., 2018; Frenkel et al., 2018) use a time-driven approach for simulating neurons with simulation time steps of around 1 ms. While some systems can operate at a higher temporal resolution, this typically requires increasing the clock speed, leading to increased power consumption. This programmability of the SpiNNaker platform (Furber et al., 2014) means that, although this platform was *designed* to operate on a 1 ms simulation time step, it has been recently demonstrated that a 0.1 ms time step is achievable through careful programming (Knight and Furber, 2016; Rhodes et al., 2020). Furthermore, when even higher temporal resolution

is required, truly event-driven models capable of learning temporal patterns with sub-millisecond precision have also been demonstrated on SpiNNaker (Lagorce et al., 2015). On the other hand, in terms of efficient processing with high temporal precision, mixed-signal analog/digital neuromorphic systems such as ROLLS (Qiao et al., 2015) or DYNAP-SE (Moradi et al., 2017) have a distinct advantage, as their neuronal dynamics arise from the physical characteristics of their analog circuits so time *represents itself*. As such, analog systems have been successfully used for a variety of complex spatio-temporal signal processing tasks, even exhibiting cognitive abilities Neftci et al. (2013), or including spike-based plasticity mechanism applied to learning auditory features from a silicon cochlea (Sheik et al., 2012), sequence learning (Kreiser et al., 2018a; Milde, 2019) and simultaneous localization and mapping using a silicon retina (Kreiser et al., 2018b).

We anticipate that this work will be extended to qualitatively and quantitatively assess solutions to the problem of spatio-temporal pattern learning, which exploit the fact that time represent itself in neural computation, and thus uses the precise timing of events to learn in a purely event-driven manner. The Neuromorphic Engineering community is facing and needs to overcome this canonical problem, to establish itself as a viable alternative to conventional clock-based sensing and processing systems.

## DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

## AUTHOR CONTRIBUTIONS

GH and MM developed the neuromorphic tactile sensor. GH developed the analytical solution to the localization problem. PA developed the temporal coincidence detection and complex weights and delays based algorithms. JK developed the Temporal Difference Encoder based algorithm. OO developed the synaptic delay plasticity algorithm. MM developed the structural plasticity algorithm. GH, MM, PA, OO, JK, and AS wrote the paper. AS, RB, and GI contributed to the paper writing process and supervised the research.

## FUNDING

This work was partially supported by the EPSRC (Brains on Board project, grant number EP/P006094/1), the Bundesministerium für Bildung und Forschung and the Max Planck Society through the Max Planck School of Cognition, the NEUROTECH CSA project through the Neurotech fellowship, the European Union's Horizon 2020 research and innovation programme under grant agreement no 732642 and the Swiss National Science Foundation (Sinergia project #CRSII5-18O316).



## ACKNOWLEDGMENTS

We authors would like to thank the organizers of the CapoCaccia Neuromorphic Engineering Workshop 2019, where intense

discussion and work on the approaches proposed in this paper began. Additionally, we would like to thank Wolfgang Stuerzl for kindly providing us with a library of beautiful photographs of sand scorpions, Martin Reichert for originally taking them.

## REFERENCES

- Afshar, S., George, L., Tapson, J., van Schaik, A., and Hamilton, T. J. (2014). Racing to learn: statistical inference and learning in a single spiking neuron with adaptive kernels. *Front. Neurosci.* 8:377. doi: 10.3389/fnins.2014.00377
- Afshar, S., Hamilton, T. J., Tapson, J., van Schaik, A., and Cohen, G. (2019a). Investigation of event-based surfaces for high-speed detection, unsupervised feature extraction, and object recognition. *Front. Neurosci.* 12:1047. doi: 10.3389/fnins.2018.01047
- Afshar, S., Xu, Y., Tapson, J., van Schaik, A., and Cohen, G. (2019b). Event-based feature extraction using adaptive selection thresholds. *arXiv preprint arXiv:1907.07853*. doi: 10.3390/s20061600
- Astrom, K. J., and Bernhardsson, B. M. (2002). "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proceedings of the 41st IEEE Conference on Decision and Control*, Vol. 2 (Las Vegas, NV: IEEE), 2011–2016. doi: 10.1109/CDC.2002.1184824
- Baldi, P., and Atiya, A. F. (1994). How delays affect neural dynamics and learning. *IEEE Trans. Neural Netw.* 5, 612–621. doi: 10.1109/72.298231
- Baudot, P., Levy, M., Marre, O., Monier, C., Pananceau, M., and Frégnac, Y. (2013). Animation of natural scene by virtual eye-movements evokes high precision and low noise in v1 neurons; handwritten. *Front. Neural Circ.* 7:206. doi: 10.3389/fncir.2013.00206
- Bekkers, J. M. (2011). Changes in dendritic axial resistance alter synaptic integration in cerebellar Purkinje cells. *Biophys. J.* 100, 1198–1206. doi: 10.1016/j.bpj.2011.01.042
- Brownell, P., and Farley, R. D. (1979a). Detection of vibrations in sand by tarsal sense organs of the nocturnal scorpion, *Paruroctonus mesaensis*. *J. Comp. Physiol. A* 131, 23–30. doi: 10.1007/BF00613080
- Brownell, P., and Farley, R. D. (1979b). Orientation to vibrations in sand by the nocturnal scorpion *Paruroctonus mesaensis*: mechanism of target localization. *J. Comp. Physiol. A* 131, 31–38. doi: 10.1007/BF00613081
- Brownell, P. H. (1977). Compressional and surface waves in sand: Used by desert scorpions to locate prey. *Science* 197, 479–482. doi: 10.1126/science.197.4302.479
- Buonomano, D. V., and Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nat. Rev. Neurosci.* 10, 113–125. doi: 10.1038/nrn2558
- Buonomano, D. V., and Merzenich, M. M. (1995). Temporal information transformed into a spatial code by a neural network with realistic properties. *Science* 267, 1028–1030. doi: 10.1126/science.7863330
- Carr, C. E., and Konishi, M. (1990). A circuit for detection of interaural time differences in the brain-stem of the barn owl. *J. Neurosci.* 10, 3227–3246. doi: 10.1523/JNEUROSCI.10-10-03227.1990
- Chauhan, T., Masquelier, T., Montlibert, A., and Cottreau, B. R. (2018). Emergence of binocular disparity selectivity through Hebbian learning. *J. Neurosci.* 38, 9563–9578. doi: 10.1523/JNEUROSCI.1259-18.2018
- Chicca, E., Stefanini, F., Bartolozzi, C., and Indiveri, G. (2014). Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. IEEE* 102, 1367–1388. doi: 10.1109/JPROC.2014.2313954
- Cohen, G. K., Orchard, G., Leng, S.-H., Tapson, J., Benosman, R. B., and van Schaik, A. (2016). Skimming digits: neuromorphic classification of spike-encoded images. *Front. Neurosci.* 10:184. doi: 10.3389/fnins.2016.00184
- Corradi, F., and Indiveri, G. (2015). A neuromorphic event-based neural recording system for smart brain-machine-interfaces. *IEEE Trans. Biomed. Circ. Syst.* 9, 699–709. doi: 10.1109/TBCAS.2015.2479256
- Davies, M., Srinivasa, N., Lin, T.-H., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 30, 82–99. doi: 10.1109/MM.2018.112130359
- Dayan, P., and Abbott, L. (2001). *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. Cambridge, MA: MIT Press.
- Dean, A. (1981). The variability of discharge of simple cells in the cat striate cortex. *Exp. Brain Res.* 44, 437–440. doi: 10.1007/BF00238837
- Deneve, S., and Machens, C. K. (2016). Efficient codes and balanced networks. *Nat. Neurosci.* 19:375. doi: 10.1038/nn.4243
- Eurich, C. W., Pawelzik, K., Ernst, U., Cowan, J. D., and Milton, J. G. (1999). Dynamics of self-organized delay adaptation. *Phys. Rev. Lett.* 82, 1594–1597. doi: 10.1103/PhysRevLett.82.1594
- Eurich, C. W., Pawelzik, K., Ernst, U., Thiel, A., Cowan, J. D., and Milton, J. G. (2000). Delay adaptation in the nervous system. *Neurocomputing* 32, 741–748. doi: 10.1016/S0925-2312(00)00239-3
- Fields, R. D. (2015). A new mechanism of nervous system plasticity: activity-dependent myelination. *Nat. Rev. Neurosci.* 16, 756–767. doi: 10.1038/nrn4023
- Frady, E. P., and Sommer, F. T. (2019). Robust computation with rhythmic spike patterns. *Proc. Natl. Acad. Sci. U.S.A.* 116, 18050–18059. doi: 10.1073/pnas.1902653116
- Frenkel, C., Lefebvre, M., Legat, J.-D., and Bol, D. (2018). A 0.086-mm<sup>2</sup> 12.7-pJ/SOP 64k-synapse 256-neuron online-learning digital spiking neuromorphic processor in 28nm CMOS. *IEEE Trans. Biomed. Circ. Syst.* 13, 145–158. doi: 10.1109/TBCAS.2018.2880425
- Furber, S. B., Galluppi, F., Temple, S., and Plana, L. A. (2014). The SpiNNaker project. *Proc. IEEE* 102, 652–665. doi: 10.1109/JPROC.2014.2304638
- George, R. M. (2018). *Structural plasticity in neuromorphic systems* (Ph.D. thesis). University of Zurich, Zurich, Switzerland. doi: 10.1109/BIOCAS.2017.8325074
- Gerstner, W., Kempter, R., van Hemmen, J. L., and Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature* 383:76. doi: 10.1038/383076a0
- Goel, A., and Buonomano, D. V. (2016). Temporal interval learning in cortical cultures is encoded in intrinsic network dynamics. *Neuron* 91, 320–327. doi: 10.1016/j.neuron.2016.05.042
- Goodhill, G. J., and Barrow, H. G. (1994). The role of weight normalization in competitive learning. *Neural Comput.* 6, 255–269. doi: 10.1162/neco.1994.6.2.255
- Gütig, R., and Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing-based decisions. *Nat. Neurosci.* 9, 420–428. doi: 10.1038/nn1643
- Hipp, J., Arabzadeh, E., Zorzin, E., Conradt, J., Kayser, C., Diamond, M. E., et al. (2006). Texture signals in whisker vibrations. *J. Neurophysiol.* 95, 1792–1799. doi: 10.1152/jn.01104.2005
- Hirose, A. (1992). Continuous complex-valued back-propagation learning. *Electron. Lett.* 28, 1854–1855. doi: 10.1049/el:19921186
- Hu, J.-S., and Yang, C.-H. (2010). Estimation of sound source number and directions under a multisource reverberant environment. *EURASIP J. Adv. Signal Process.* 2010:870756. doi: 10.1155/2010/870756
- Hubel, D., and Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160, 106–54. doi: 10.1113/jphysiol.1962.sp006837
- Hussain, S., Basu, A., Wang, M., and Hamilton, T. J. (2012). "Deltron: neuromorphic architectures for delay based learning," in *2012 IEEE Asia Pacific Conference on Circuits and Systems* (Kaohsiung: IEEE), 304–307. doi: 10.1109/APCCAS.2012.6419032
- Indiveri, G., and Sandamirskaya, Y. (2019). The importance of space and time for signal processing in neuromorphic agents. *IEEE Signal Process. Mag.* 36, 16–28. doi: 10.1109/MSP.2019.2928376
- Izhikevich, E. (2006). Polychronization: computation with spikes. *Neural Comput.* 18, 245–282. doi: 10.1162/089976606775093882
- Kappel, D., Nessler, B., and Maass, W. (2014). STDP installs in winner-take-all circuits an online approximation to hidden markov model learning. *PLoS Comput. Biol.* 10:e1003511. doi: 10.1371/journal.pcbi.1003511
- Knight, J., and Furber, S. (2016). Synapse-centric mapping of cortical models to the SpiNNaker neuromorphic architecture. *Front. Neurosci.* 10:420. doi: 10.3389/fnins.2016.00420

- Knoblauch, A., Körner, E., Körner, U., and Sommer, F. T. (2014). Structural synaptic plasticity has high memory capacity and can explain graded amnesia, catastrophic forgetting, and the spacing effect. *PLoS ONE* 9:e96485. doi: 10.1371/journal.pone.0096485
- Koudelka, S., Voas, M. G., Almeida, R. G., Baraban, M., Soetaert, J., Meyer, M. P., et al. (2016). Individual neuronal subtypes exhibit diversity in CNS myelination mediated by synaptic vesicle release. *Curr. Biol.* 26, 1447–1455. doi: 10.1016/j.cub.2016.03.070
- Krammer, J., and Koch, C. (1997). Pulse-based analog VLSI velocity sensors. *IEEE Trans. Circ. Syst. II Anal. Digital Signal Process.* 44, 86–101. doi: 10.1109/82.554431
- Kreiser, R., Aathmani, D., Qiao, N., Indiveri, G., and Sandamirskaya, Y. (2018a). Organizing sequential memory in a neuromorphic device using dynamic neural fields. *Front. Neurosci.* 12:717. doi: 10.3389/fnins.2018.00717
- Kreiser, R., Renner, A., Sandamirskaya, Y., and Pienroj, P. (2018b). “Pose estimation and map formation with spiking neural networks: toward neuromorphic slam,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 2159–2166. doi: 10.1109/IROS.2018.8594228
- Lagorce, X., Orchard, G., Galluppi, F., Shi, B. E., and Benosman, R. B. (2017). Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 1346–1359. doi: 10.1109/TPAMI.2016.2574707
- Lagorce, X., Stomatias, E., Galluppi, F., Plana, L. A., Liu, S.-C., Furber, S. B., et al. (2015). Breaking the millisecond barrier on SpiNNaker: implementing asynchronous event-based plastic models with microsecond resolution. *Front. Neurosci.* 9:206. doi: 10.3389/fnins.2015.00206
- Laje, R., and Buonomano, D. V. (2013). Robust timing and motor patterns by taming chaos in recurrent neural networks. *Nat. Neurosci.* 16:925. doi: 10.1038/nn.3405
- Lee, W. W., Kukreja, S. L., and Thakor, N. V. (2017). Discrimination of dynamic tactile contact by temporally precise event sensing in spiking neuromorphic networks. *Front. Neurosci.* 11:5. doi: 10.3389/fnins.2017.00005
- Maass, W. (2001). On the relevance of time in neural computation and learning. *Theor. Comput. Sci.* 261, 157–178. doi: 10.1016/S0304-3975(00)00137-7
- Mahajan, A., and Walworth, M. (2001). 3D position sensing using the differences in the time-of-flights from a wave source to various receivers. *IEEE Trans. Robot. Autom.* 17, 91–94. doi: 10.1109/70.917087
- Mainen, Z., and Sejnowski, T. (1995). Reliability of spike timing in neocortical neurons. *Science* 268, 1503–1506. doi: 10.1126/science.7770778
- Markram, H., Lübke, J., Frotscher, M., and Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APS and EPSPs. *Science* 275, 213–215. doi: 10.1126/science.275.5297.213
- Masquelier, T. (2012). Relative spike time coding and STDP-based orientation selectivity in the early visual system in natural continuous and saccadic vision: a computational model. *J. Comput. Neurosci.* 32, 425–441. doi: 10.1007/s10827-011-0361-9
- Matsubara, T. (2017). Spike timing-dependent conduction delay learning model classifying spatio-temporal spike patterns. *Front. Comput. Neurosci.* 11:104. doi: 10.3389/fncom.2017.00104
- Merolla, P. A., Arthur, J. V., Alvarez-Icaza, R., Cassidy, A. S., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642
- Milde, M. B. (2019). *Spike-based computational primitives for vision-based scene understanding* (Ph.D. thesis), University of Zurich.
- Milde, M. B., Bertrand, O. J. N., Ramachandran, H., Egelhaaf, M., and Chicca, E. (2018). Spiking elementary motion detector in neuromorphic systems. *Neural Comput.* 30, 2384–2417. doi: 10.1162/neco\_a\_01112
- Moradi, S., Qiao, N., Stefanini, F., and Indiveri, G. (2017). A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (DYNAPs). *IEEE Trans. Biomed. Circ. Syst.* 12, 106–122. doi: 10.1109/TBCAS.2017.2759700
- Morrison, A., Diesmann, M., and Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. *Biol. Cybernet.* 98, 459–478. doi: 10.1007/s00422-008-0233-1
- Neftci, E., Binas, J., Rutishauser, U., Chicca, E., Indiveri, G., and Douglas, R. (2013). Synthesizing cognition in neuromorphic electronic systems. *Proc. Natl. Acad. Sci. U.S.A.* 110, E3468–E3476. doi: 10.1073/pnas.1212083110
- Nowotny, T. (2014). Two challenges of correct validation in pattern recognition. *Front. Robot. AI* 1:5. doi: 10.3389/frobt.2014.00005
- O’Keefe, J., and Recce, M. L. (1993). Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus* 3, 317–330. doi: 10.1002/hipo.450030307
- Paugam-Moisy, H., Martinez, R., and Bengio, S. (2008). Delay learning and polychronization for reservoir computing. *Neurocomputing* 71, 1143–1158. doi: 10.1016/j.neucom.2007.12.027
- Qiao, N., Bartolozzi, C., and Indiveri, G. (2016). “Automatic gain control of ultra-low leakage synaptic scaling homeostatic plasticity circuits,” in *Biomedical Circuits and Systems Conference (BioCAS)* (Shanghai: IEEE), 156–159. doi: 10.1109/BioCAS.2016.7833755
- Qiao, N., Bartolozzi, C., and Indiveri, G. (2017). An ultralow leakage synaptic scaling homeostatic plasticity circuit with configurable time scales up to 100 ks. *IEEE Trans. Biomed. Circ. Syst.* 11, 1271–1277. doi: 10.1109/TBCAS.2017.2754383
- Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., et al. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Front. Neurosci.* 9:141. doi: 10.3389/fnins.2015.00141
- Rall, W. (1967). Distinguishing theoretical synaptic potentials computed for different soma-dendritic distributions of synaptic input. *J. Neurophysiol.* 30, 1138–1168. doi: 10.1152/jn.1967.30.5.1138
- Reichert, D. P., and Serre, T. (2013). Neuronal synchrony in complex-valued deep networks. *arXiv [Preprint] arXiv:1312.6115*.
- Rhodes, O., Peres, L., Rowley, A. G. D., Gait, A., Plana, L. A., Brenninkmeijer, C., et al. (2020). Real-time cortical simulation on neuromorphic hardware. *Philos. Trans. R. Soc. A* 378:20190160. doi: 10.1098/rsta.2019.0160
- Roy, S., and Basu, A. (2016). An online unsupervised structural plasticity algorithm for spiking neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* 28, 900–910. doi: 10.1109/TNNLS.2016.2582517
- Roy, S., San, P. P., Hussain, S., Wei, L. W., and Basu, A. (2016). Learning spike time codes through morphological learning with binary synapses. *IEEE Trans. Neural Netw. Learn. Syst.* 27, 1572–1577. doi: 10.1109/TNNLS.2015.2447011
- Shadlen, M. N., and Newsome, W. T. (1998). The variable discharge of cortical neurons: implications for connectivity, computation, and information coding. *J. Neurosci.* 18, 3870–3896. doi: 10.1523/JNEUROSCI.18-10-03870.1998
- Sheik, S., Coath, M., Indiveri, G., Denham, S. L., Wennekers, T., and Chicca, E. (2012). Emergent auditory feature tuning in a real-time neuromorphic VLSI system. *Front. Neurosci.* 6:17. doi: 10.3389/fnins.2012.00017
- Shrestha, S. B., and Orchard, G. (2018). “Slayer: spike layer error reassignment in time,” in *Advances in Neural Information Processing Systems* (Montreal, QC), 1412–1421.
- Softky, W. R., and Koch, C. (1993). The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. *J. Neurosci.* 13, 334–350. doi: 10.1523/JNEUROSCI.13-01-00334.1993
- Song, S., Miller, K. D., and Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.* 3, 919–926. doi: 10.1038/78829
- State, L. (2019). *Training delays in spiking neural networks* (Master’s thesis). Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany. Available from the Preprint repository of the Max Planck Institute for Mathematics in the Sciences, Preprint 96/2019.
- Stürzl, W., Kempter, R., and Van Hemmen, J. L. (2000). Theory of arachnid prey localization. *Phys. Rev. Lett.* 84, 5668–5671. doi: 10.1103/PhysRevLett.84.5668
- Swadlow, H. A., and Waxman, S. G. (2012). Axonal conduction delays. *Scholarpedia* 7:1451. doi: 10.4249/scholarpedia.1451
- Taherkhani, A., Belatreche, A., Li, Y., and Maguire, L. P. (2015). DL-resume: a delay learning-based remote supervised method for spiking neurons. *IEEE Trans. Neural Netw. Learn. Syst.* 26, 3137–3149. doi: 10.1109/TNNLS.2015.2404938
- Takagi, H. (2000). Roles of ion channels in EPSP integration at neuronal dendrites. *Neurosci. Res.* 37, 167–171. doi: 10.1016/S0168-0102(00)00120-6
- Tapson, J., and van Schaik, A. (2013). Learning the pseudoinverse solution to network weights. *Neural Netw.* 45, 94–100. doi: 10.1016/j.neunet.2013.02.008
- Thorpe, S., Delorme, A., Rullen, R. V., et al. (2001). Spike-based strategies for rapid processing. *Neural Netw.* 14, 715–725. doi: 10.1016/S0893-6080(01)00083-1
- Turrigiano, G., and Nelson, S. (2004). Homeostatic plasticity in the developing nervous system. *Nat. Rev. Neurosci.* 5, 97–107. doi: 10.1038/nrn1327

- Wang, R., Cohen, G., Stiefel, K., Hamilton, T., Tapson, J., and van Schaik, A. (2013). An FPGA implementation of a polychronous spiking neural network with delay adaptation. *Front. Neurosci.* 7:14. doi: 10.3389/fnins.2013.00014
- Wang, R. M., Hamilton, T. J., Tapson, J. C., and van Schaik, A. (2015). A neuromorphic implementation of multiple spike-timing synaptic plasticity rules for large-scale neural networks. *Front. Neurosci.* 9:180. doi: 10.3389/fnins.2015.00180
- Wehr, M., and Zador, A. M. (2003). Balanced inhibition underlies tuning and sharpens spike timing in auditory cortex. *Nature* 426:442. doi: 10.1038/nature02116
- Wolfe, J., Hill, D. N., Pahlavan, S., Drew, P. J., Kleinfeld, D., and Feldman, D. E. (2008). Texture coding in the rat whisker system: slip-stick versus differential resonance. *PLoS Biology* 6:e215. doi: 10.1371/journal.pbio.0060215
- Yavuz, E., Turner, J., and Nowotny, T. (2016). GeNN: a code generation framework for accelerated brain simulations. *Sci. Rep.* 6:18854. doi: 10.1038/sr5ep18854
- Zito, K., Parnas, D., Fetter, R. D., Isacoff, E. Y., and Goodman, C. S. (1999). Watching a synapse grow: noninvasive confocal imaging of synaptic growth in *Drosophila*. *Neuron* 22, 719–729. doi: 10.1016/S0896-6273(00)80731-X

**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.

Copyright © 2020 Haessig, Milde, Aceituno, Oubari, Knight, van Schaik, Benosman and Indiveri. 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.