

# A Neuromorphic Vision and Feedback Sensor Fusion Based on Spiking Neural Networks for Real-Time Robot Adaption

Pablo López-Osorio,\* Juan Pedro Domínguez-Morales, and Fernando Perez-Peña

For some years now, the locomotion mechanisms used by vertebrate animals have been a major inspiration for the improvement of robotic systems. These mechanisms range from adapting their movements to move through the environment to the ability to chase prey, all thanks to senses such as sight, hearing, and touch. Neuromorphic engineering is inspired by brain problem-solving techniques with the goal of implementing models that take advantage of the characteristics of biological neural systems. While this is a well-defined and explored area in this field, there is no previous work that fuses analog and neuromorphic sensors to control and modify robotic behavior in real time. Herein, a system is presented based on spiking neural networks implemented on the SpiNNaker hardware platform that receives information from both analog (force-sensing resistor) and digital (neuromorphic retina) sensors and is able to adapt the speed and orientation of a hexapod robot depending on the stability of the terrain where it is located and the position of the target. These sensors are used to modify the behavior of different spiking central pattern generators, which in turn will adapt the speed and orientation of the robotic platform, all in real time. In particular, experiments show that the network is capable of correctly adapting to the stimuli received from the sensors, modifying the speed and heading of the robotic platform.

generating rhythmic patterns of neural activity even without requiring external sensory inputs.<sup>[6,7]</sup> They are located in the spinal cord of vertebrates or in the nervous system of invertebrates and are responsible for controlling and coordinating repetitive movements and sequences of activity, such as walking in mammals,<sup>[8]</sup> the rhythm of swimming in fish,<sup>[9]</sup> the control of the heartbeat,<sup>[10]</sup> or the basic rhythm of breathing.<sup>[11]</sup> Usually, these neural structures are made of, at least, two neuronal populations that alternate their cycles of activity, producing oscillations in the output without the need to receive oscillating signals as stimulus.

The research field called neuromorphic engineering aims to implement these neural structures in electronic devices, emulating the way living beings have solved complex problems using biological circuits.<sup>[12]</sup> Neuromorphic robotics combines both the neuromorphic engineering community and roboticists, leading to

potential applications that bring together expertise from both fields.<sup>[13,14]</sup>

The robotics field borrows concepts from biological CPGs to create locomotion in robotic platforms. To implement these neural structures, there are several options available, including using coupled oscillators, artificial neural networks (ANNs), or spiking neural networks (SNNs). Among these options, SNNs stand out as the closest to biology in terms of performance and behavior. This is primarily due to its temporal information encoding, which enhances its resilience against noise when compared to ANNs.<sup>[15]</sup> Furthermore, SNNs stand out in terms of biological synaptic plasticity and energy efficiency,<sup>[16]</sup> as they only consume energy when neurons emit spikes (action potentials), closely resembling the functioning of the brain. Consequently, when applied in robotics, the use of SNNs results in reduced resource requirements, lower energy consumption, and simplified algorithms, in contrast to traditional approaches that rely on ANNs or coupled oscillators.

The use of SNNs for locomotion generation gave rise to spiking central pattern generator (sCPG). Using spiking neurons, a rhythmic output pattern can be generated, which can be used to generate motion on robotic platforms. In addition, one of the advantages of SNNs is that they can generate very stable patterns


## 1. Introduction

It has long been known that biological locomotion is produced by a neural structure called central pattern generator (CPG).<sup>[1–5]</sup> This structure is made of neural circuits that are specialized in

---

P. López-Osorio, F. Perez-Peña  
School of Engineering  
Universidad de Cádiz  
11519 Puerto Real, Spain  
E-mail: pablo.osorio@uca.es

J. P. Domínguez-Morales  
Robotics and Technology of Computers Lab.  
ETSII-EP5  
Universidad de Sevilla  
Sevilla 41012, Spain

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300646>.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300646

without the need to incorporate any sensory or brain information to regulate their operation or oscillation frequency.<sup>[14,17,18]</sup>

Previous works that explore the implementation of CPGs as signal generators for robotic actuator control are: ref. [19] where two different topologies of CPGs with coupled oscillators are presented to provide different motion patterns to a quadruped robot; however, these patterns are not interchangeable on the fly; others like ref. [20] or ref. [14] use a set of CPGs to produce various motion patterns in hexapod robots; however, the pattern cannot be modified autonomously. The same occurs in ref. [19].

There are some previous works that use a sCPG similar to the one proposed in this article but they include an open-loop network that does not conceive the use of sensory inputs for real-time modification of the system. For example, in refs. [14,20,21], three sCPGs designs capable of implementing different gait patterns on robotic platforms are presented, using the SpiNNaker neuromorphic platform<sup>[22]</sup> together with a field programmable gate array (FPGA) to process the signals received from the neuromorphic chip. In ref. [21], in addition, a dynamic vision sensor (DVS) is used that influences the selection of the hexapod's walking rhythm, but a constant connection to a computer is required. In ref. [23], the authors propose another sCPG architecture with the ability to modify its amplitude, phase, and frequency without requiring any sensory information. Other neuromorphic platforms such as Loihi<sup>[24]</sup> have also been used to generate locomotion in a robotic lamprey that can select up to two different gait patterns but does not use a control loop over the generated<sup>[25]</sup> motion. These previous works demonstrate that the use of sCPGs provides significant energy savings compared to conventional actuation and control methods. Furthermore, these works highlight the fact that it would be interesting to incorporate external sensory feedback to modify the behavior of the neural network and, therefore, the behavior of the robotic platform.

There are some previous works that include sensory feedback.<sup>[26–28]</sup> In such cases, conventional oscillators have been used to model the behavior of the CPG, or Hopf oscillators, as in ref. [29] where it is used to generate the rhythmic pattern in the CPG. In another recent work,<sup>[30]</sup> they used different inertial measurement unit sensors and distance sensors for a robotic fish to modify in real time a series of coupled oscillators, autonomously selecting the most optimal swimming routine depending on the surroundings.

In ref. [31], 12 spiking neurons modulated by sensory feedback are used to shape the sCPG and the modification of the motion pattern of the robotic structure is achieved by adding or subtracting the neural structures, whereas in our work, it is done by an adaptive mechanism of the SNN. Furthermore, the authors used the Izhikevich neural model,<sup>[32]</sup> which involves a more complex computational model than the one proposed in this article: the leaky integrate-and-fire (LIF) model. Conversely, in ref. [33], a neuromorphic auditory sensor (NAS) is used as an input sensor to decide the pattern to be followed by the sCPG in SpiNNaker, depending on the audio processed by a FPGA and recognized by the SNN. In ref. [34], a closed-loop system is presented that incorporates a time difference encoder in SpiNNaker together with a FPGA to process the intermediate signals, which analyzes the information received from a NAS and a sound source localization system to modify the behavior of a robotic arm. In our work, we use a similar approach: a fusion

of the force-sensitive resistors (FSRs) and the DVS is used to modify the behavior of the neural network implemented in SpiNNaker in real time.

A recent work proposes the combination of firing and nonfiring neurons to simulate a neural network that regulates amplitude, phase, and frequency (each independently) in a closed-loop system.<sup>[35]</sup> On the one hand, the nonfiring neurons used in Strohmer's study are LIF neurons, with a threshold high enough to avoid neuronal firing when the analog input varies. On the other hand, the firing neurons employ the adaptive exponential integrate-and-fire (AdEx) model, a considerably more complex neuronal model than that used in the work presented in this article, with a complex implementation in neuromorphic hardware. The primary focus of the study is the search for an optimal combination of spiking and nonspiking neurons to create a sensorimotor neuron network capable of adjusting the network's output (spiking) based on analog parameters at the network's input (nonspiking). However, locomotion is not the primary objective of this article, although it is briefly mentioned. As mentioned earlier, the authors propose compatibility of their network with neuromorphic hardware, although this compatibility is limited only to the nonspiking neurons and a cloud-based simulation service called CloudBrain.

The proposal made in this article differs in several aspects: compared to ref. [35], a different spiking neuron model is used and all neurons are firing neurons. The second aspect that sets us apart is the use of information obtained from an FSR and a DVS that allows us to modify the behavior of the neural network controlling the hexapod robot. This enables us to adjust its speed and direction based on terrain characteristics and activity detected within the field of view of the DVS. The third and most important aspect to emphasize is the validation of our model on neuromorphic hardware, specifically on the SpiNNaker platform, where we utilize high-level software tools that enhance the reproducibility of the network for the scientific community.

The main goal of this work is to design and implement a SNN composed of a sCPG and a Winner Take All (WTA) network that change their behavior depending on external variables in the environment of a hexapod robot. On the one hand, the FSR sensors will increase or decrease the activity of the sCPG depending on the stiffness of the terrain, resulting in a higher or lower speed of the hexapod. On the other hand, the DVS sensor will detect the predominant direction of motion, generating spikes associated with this direction and sending them to the WTA network. The WTA network will then select the winner and transmit the spikes to the FPGA to modify the behavior of the robotic platform, implementing a prey-predator model. Furthermore, the SNN network presented in this work is able to adapt to these input stimuli and modify the behavior of the neural network only with the external sensory information provided. Although FSR and DVS sensors are used, the adaptation step presented in this work can be used with any type of sensor to introduce the external stimulus.

The main contributions of this work are: 1) To the authors' knowledge, this work is the first to combine neuromorphic and analog sensors as input to the SpiNNaker platform for closed-loop control of a hexapod using sCPGs; 2) The experiments have not only been simulated but also implemented in hardware, in this case using an FPGA and a spinn-3 platform;

and 3) The implemented workflow is fully functional in real time and with closed-loop online terrain adaptation and pattern change.

The rest of the article is structured as follows: Section 2 is divided into two subsections: first, the materials (Section 2.1) used in this work are introduced, including the neuromorphic platform and the hardware used. The implemented methods are described in Section 2.2, along with the SNN models. The results obtained are presented in Section 3 and discussed in Section 4 and, finally, the conclusions of this work are briefly discussed in Section 5.

## 2. Experimental Section

### 2.1. Materials

This section describes both the software and hardware used to perform the experiments proposed in this work.

#### 2.1.1. Spiking Neural Network Architecture (SpiNNaker)

SpiNNaker<sup>[22,36,37]</sup> stands for “Spiking Neural Network Architecture” and is a research project and hardware platform designed to simulate large-scale biological neural networks. It was developed at the University of Manchester in the UK as part of the Human Brain Project (HBP)<sup>[38]</sup> and is primarily used in the fields of computational neuroscience and biologically inspired artificial intelligence.

The idea behind SpiNNaker is to use specialized hardware design to simulate how neural networks work in the human brain. SpiNNaker models the communication between neurons in a neural network using a large number of low-cost and effective processors coupled in a certain way instead of relying on conventional central processing units.<sup>[39]</sup> These processors can perform computations at the neuronal level and connect to each other in a low-latency, high-speed network. This configuration enables an asynchronous communication infrastructure for transmitting short packets, representing individual neuron firings,<sup>[40]</sup> identified using address event representation (AER).<sup>[41]</sup>

Various SpiNNaker machines were manufactured and made commercially available, including SpiNN-3 and SpiNN-5, which house 4 and 48 SpiNNaker chips, respectively. Additionally, they incorporate spiNNlinks,<sup>[42]</sup> which facilitates real-time input/output interfacing with neuromorphic sensors and other neuromorphic platforms such as FPGAs.<sup>[43–45]</sup> To design and implement SNNs on these machines, researchers can employ a PyNN-based software package<sup>[46]</sup> named sPyNNaker.<sup>[47]</sup>

Notably, the Advanced Processor Technologies group from the School of Computer Science at the University of Manchester recently constructed a million-core machine, accessible through the HBP portal. In this study, a local SpiNN-3 machine was utilized to implement the robotic platform.

#### 2.1.2. FPGA

An FPGA Nexys 4 DDR<sup>[48]</sup> development board was used for communication between the sensors and the neuromorphic

platform. This board is equipped with an FPGA Xilinx Artix-7 XC7A100T-1CSG324C, which offers a programmable logic capacity of 100 000 cells and digital signal processors slices, providing sufficient processing power for this work.

#### 2.1.3. DVS

The AER DVS128 retina chip (silicon retina)<sup>[49]</sup> is an array of autonomous photoreceptors (pixels) that respond to logarithmic changes in brightness intensity in real time by placing the address of that specific pixel on an asynchronous bus. Only pixels that are stimulated by a change in illumination generate spikes, saving storage and bandwidth over conventional vision sensors. The pixel addresses use of the AER protocol and contain the *X* and *Y* coordinates of the photoreceptor that detected the change in brightness.

In this work, a retina DVS sensor<sup>[50,51]</sup> was used, containing a  $128 \times 128$  pixels matrix, whose generated output is used as input to a WTA network to decide whether the robotic platform should continue forward or whether it should turn left or right, to target the direction that generates the highest spiking activity. The addresses associated with each pixel are composed of 7 bits associated with the *Y* coordinate, 7 bits associated with the *X* coordinate, and a polarity bit representing the change in contrast, where a logic 1 means an increase in brightness and a logic 0 means a decrease. This sensor is connected to a pAER interface that allows real-time visualization of the retinal output, while simultaneously sending the events to the FPGA platform through the CAVIAR connector.<sup>[52]</sup>

#### 2.1.4. FSRs

To modify the running speed of the robotic platform, different analog sensors have been used. These sensors will generate a voltage output depending on the pressure that is exerted on them. Since the weight of the robotic platform is stable and equally distributed on each leg, the only variable that will modify the value provided by these sensors will be the terrain on which they are located: If the robot is on firm, flat ground (such as concrete), the forces will be distributed evenly over the surface of the sensors, generating a higher pressure and therefore a higher voltage. If, in contrast, the platform is on more uneven ground (such as sand), the forces generated by the weight of the robot will not be evenly distributed over the entire surface of the sensors, thus generating less pressure on them and therefore a lower voltage.

#### 2.1.5. Dynamixel AX-12A Actuators

Eighteen Dynamixel AX-12A motors will be used to move the robotic platform.<sup>[53]</sup> These actuators have the ability to control their speed, temperature, position, voltage, and load supported. Each of these motors incorporates an ARM Cortex-M3 microcontroller that manages the internal logic of the motor, allows them to be easily controlled, and enables bidirectional communication via the Dynamixel network protocol. In addition, multiple motors can be daisy-chained on a single communication bus, allowing

multiple motors to be controlled with a single cable and a single interface.

### 2.1.6. Robotic Platform

A hexapod has been designed and manufactured using 3D printing, which allows all the necessary electronics to be housed on its surface. The robotic platform performs the movement thanks to the 18 Dynamixel motors that it incorporates, which are controlled by the FPGA.

### 2.1.7. Minimum Required Configuration

**Figure 1** illustrates the minimum configuration required to accurately replicate the tests conducted in this study. This setup includes a SpiNN-3 board, a DVS, an FSR, an FPGA unit, and two Dynamixel AX-12A motors. One of the Dynamixel motors is used to regulate speed in response to signals from the FSR, while the other adjusts orientation based on the information provided by the DVS.

## 2.2. Methodology

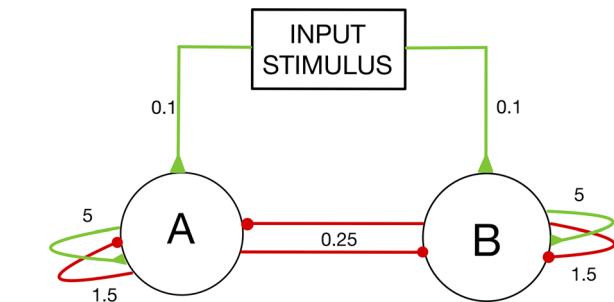
### 2.2.1. Neuron Models

The neural model used to implement the neurons of the network is the LIF model. This model is defined according to Equation (1) and (2)

$$\frac{dV}{dt} = \frac{-(V - V_r) + RI(t)}{\tau_m} \quad (1)$$

$$\text{if } V(t) = V_{th} \text{ then } \lim_{\delta \rightarrow 0; \delta > 0} V(t + \delta) = V_r \quad (2)$$

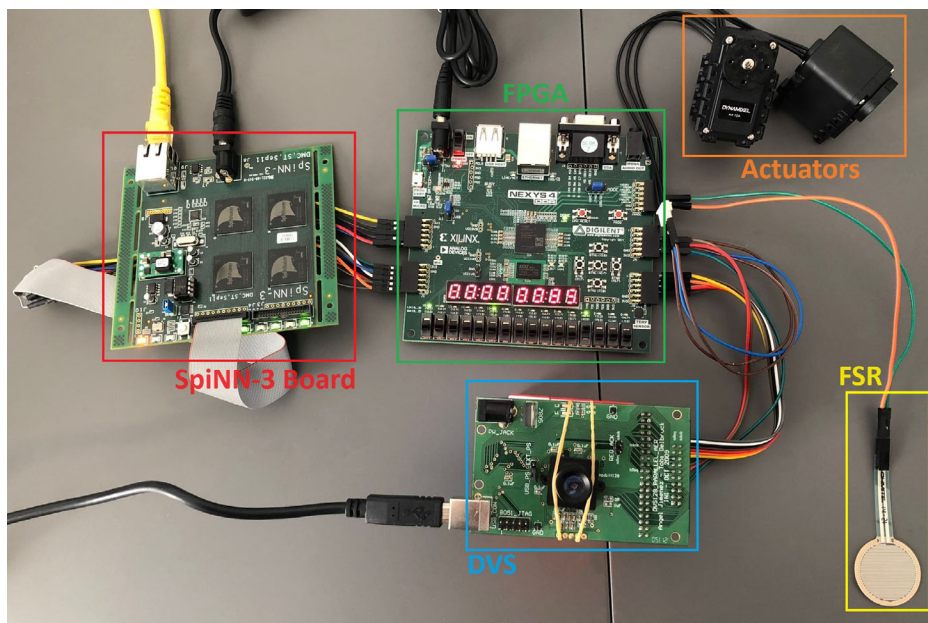
where  $V$  is the membrane potential of the neuron,  $R$  represents the resistance presented by this membrane,  $\tau_m$  the time constant



**Figure 2.** CPG<sub>AB</sub> network topology.

of the neuron,  $V_r$  the resting potential,  $V_{th}$  the threshold required to trigger the firing of the neuron, and  $I(t)$  is the stimulus (in the form of current) entering the neuron.

**CPG Populations:** The CPG is the main neural structure of the network and is composed of the populations A and B of 100 neurons each, which form the CPG<sub>AB</sub>, represented in **Figure 2**. The number of neurons per population and their parameters have been set prioritizing the reduction of the total number of neurons in the network: using fewer neurons per population causes erratic network performance, such as loss of rhythmicity or network outputs that do not correspond to the expected result. These populations, on the one hand, are self-excited and self-inhibited with a probability of 25% and 75% and weights of 5 and 1.5 nA, respectively. These probabilistic parameters have been established since the number of excitatory synapses received by the adapting network populations from the FPGA-SpiNNaker input populations are more numerous than the inhibitory connections produced internally in the adapting network itself. Furthermore, the excitatory synaptic weight values are considerably higher than those of the inhibitory synaptic weights, maintaining a balance between the probability of connection and the strength with



**Figure 1.** Minimum required configuration to replicate the experiment.

which the connection is established. The probability of inhibition between one population and the other is also 75%, with a weight of 0.25 nA. Due to this configuration, when one of the populations has activity, the other will be inhibited accordingly, causing the oscillation in the network. It is important to note that these populations are designed to work as tonic neurons, so they are injected with a very low but constant external current to ensure that they are constantly generating activity. Therefore, in these two populations the neuronal model would be as follows

$$\frac{dV}{dt} = \frac{V_r - V + R(I_{exc} - I_{inh} + I_{st})}{\tau_m} \quad (3)$$

where  $I_{st}$  is the current injected into neurons in populations A and B. This value is set to 0.1 nA, which is sufficient to maintain the desired minimum oscillation. The neuronal parameters governing the behavior of these populations are presented in **Table 1**.

**Adaptive Populations:** In this part of the network, we can distinguish three populations: POP<sub>REF</sub> is composed of 50 neurons that directly receive the input stimuli linked to the FSR, and its neuronal parameters are those presented in Table 1. POP<sub>E</sub> and POP<sub>I</sub> are populations of 100 neurons each, and their weights are configured in such a way that, depending on the activity of POP<sub>REF</sub>, the activity of these neurons increases or decreases. In turn, these populations have inhibitory connections between them, achieving an oscillatory phenomenon between these two populations similar to that of CPG<sub>AB</sub>. These populations are based on the neuronal model LIF, but their neuronal parameters are adapted in such a way that the input–output ratio of these populations is 1:1, thus ensuring that their behavior is closely related to the input provided by the FSR. Their parameters are shown in **Table 2**, and their connection scheme is illustrated in **Figure 3**.

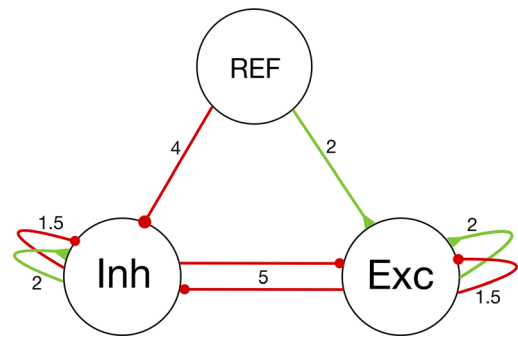
By connecting CPG<sub>AB</sub> to the adaptive network via excitatory and inhibitory synaptic connections, it is possible to modify its activity depending on the input received from the FSR sensors. The complete network is shown in **Figure 4** and its operation can be summarized in two possible cases: 1) If the activity of POP<sub>REF</sub> is greater than the activity of CPG, POP<sub>Exc</sub> will be strongly excited by POP<sub>REF</sub> and weakly inhibited by POP<sub>A</sub>, which will increase its activity, in turn causing an increase in the activity of CPG. Furthermore, POP<sub>Exc</sub> will strongly inhibit POP<sub>Inh</sub>,

**Table 1.** Neuron parameters for the proposed CPG in SpiNNaker hardware platform.

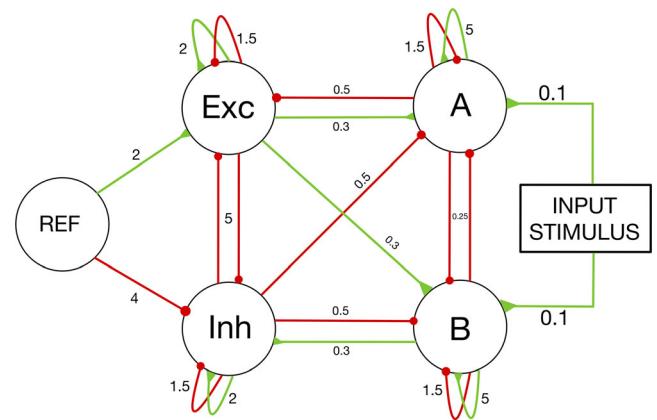
Parameter	Value
$C_m$	0.25 nF
$\tau_m$	12.5 ms
$\tau_r$	0.001 ms
$u_{reset}$	-70.0 mV
$u_{rest}$	-65.0 mV
$u_{th}$	-50.0 mV
$\tau_{syn_e}$	7.0 ms
$\tau_{syn_i}$	17.1 ms
$I_{bias}$	0.1 mA

**Table 2.** Proposed neural parameters for the adaptive populations on the SpiNNaker hardware platform.

Parameter	Value
$C_m$	0.1 nF
$\tau_m$	0.1 ms
$\tau_r$	0.0 ms
$u_{reset}$	-65.0 mV
$u_{rest}$	-65.0 mV
$u_{th}$	-64.91 mV
$\tau_{syn_e}$	0.1 ms
$\tau_{syn_i}$	0.1 ms
$I_{bias}$	0.0 mA



**Figure 3.** Topology of the CPG<sub>EI</sub> network.



**Figure 4.** FSR adaptive network topology.

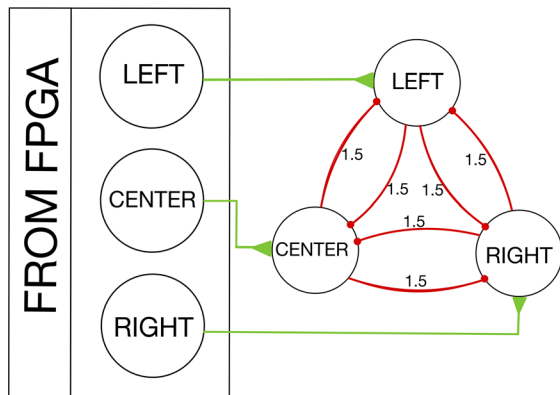
preventing the latter from inhibiting CPG; and 2) If the activity of POP<sub>REF</sub> is lower than the activity of CPG, POP<sub>Inh</sub> will be strongly excited by CPG and weakly inhibited by POP<sub>REF</sub>, which will increase its activity, which in turn will cause a decrease in the activity of CPG. In addition, POP<sub>Inh</sub> will strongly inhibit POP<sub>Exc</sub>, preventing the latter from exciting CPG.

The weights of this entire network have been tuned using a grid search-based weight search algorithm, and are configured to ensure stable oscillations of the CPG over the entire input range of the FSRs. This input will oscillate between 10 and

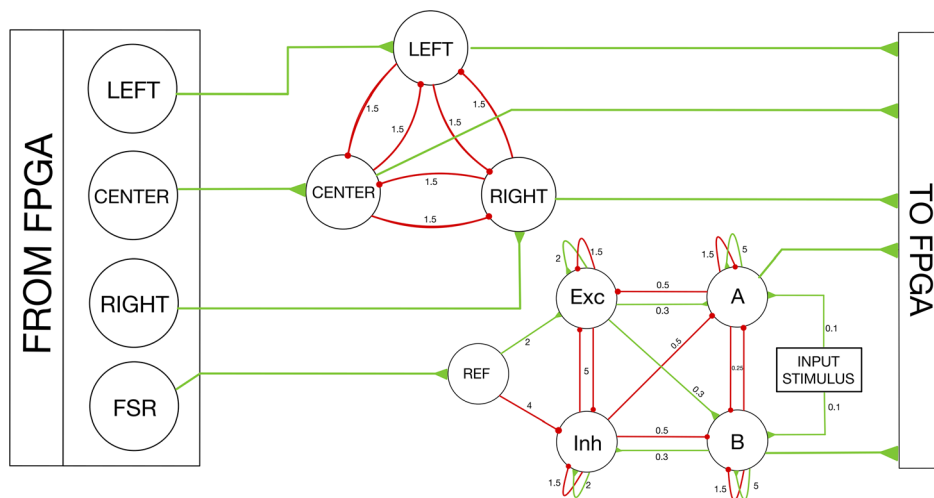
171 Hz (for minimum and maximum pressure values, respectively). Thanks to this configuration, the CPG will provide stable oscillations but with variable activity, which will be used to select the gait pattern to be adopted by the robotic platform.

**Table 3.** Neuron parameters for the proposed WTA network in SpiNNaker hardware platform.

Parameter	Value
$C_m$	0.25 nF
$\tau_m$	12.5 ms
$\tau_r$	0.001 ms
$U_{reset}$	-70.0 mV
$U_{rest}$	-65.0 mV
$U_{th}$	-50.0 mV
$\tau_{syn_e}$	7.0 ms
$\tau_{syn_i}$	17.1 ms
$I_{bias}$	0.0 mA



**Figure 5.** DVS adaptive network topology.



**Figure 6.** Complete topology of the adaptive network with a total of 9 populations and 579 neurons.

**WTA Network:** The neurons of the WTA network receive the activity generated by the DVS retina and their output will be used to select the direction to be followed by the robotic platform. These populations incorporate the same parameters as those of  $CPG_{AB}$ , except for the injected current (Table 3). For this reason, the activity or inactivity of these neurons is due solely to the input activity received by these populations from the FPGA. The topology of this network and its interaction with the DVS can be seen in Figure 5.

Since the retina is a  $128 \times 128$  pixel matrix, to select the direction of the robotic platform, it was decided to divide the X coordinate into three windows: the first 39 pixel columns are linked to the right direction, the next 50 to the center, and the last 39 to the left. In this way, when a pixel detects changes in brightness, the direction AER associated with that pixel will be sent to SpiNNaker. Depending on its X coordinate, this address will be used to excite one of the three populations in SpiNNaker. The weights of the synaptic connections are configured so that the population that generates the most activity is the winner and strongly inhibits the previous two; thanks to this, only the AER events of a single winning population will be transmitted to the robotic platform at a time, allowing it to decide which direction to follow.

### 2.2.2. Complete SNN Topology

The complete topology of the neural network that allows controlling the speed and orientation of the robotic platform is shown in Figure 6 and its distribution of populations and neurons is shown in Table 4.

## 3. Experiments and Results

This section describes the experiments conducted with respect to the proposed spiking architecture. Four experiments are carried out without the retina to check the performance of the network considering the external sensor changing value. Another four

**Table 4.** Number of neurons per population in the complete network.

Population	Number of neurons
A	100
B	100
Exc	100
Inh	100
Ref	50
Center	50
Right	39
Left	39
FSR	1

experiments include the retina to show the network decision considering the spiking input stimulus.

The first experiment includes a 10 Hz input stimulus from the population bound to the FSR (minimum value provided by the FSRs). The results are shown in **Figure 7**. It can be seen that the activity of POP<sub>EXC</sub> and POP<sub>FSR</sub> is very similar. Due to the interaction of POP<sub>EXC</sub> and POP<sub>INH</sub> with CPG<sub>AB</sub>, the average oscillation frequency of the obtained POP<sub>CPG</sub> is 7 Hz. It is important to note that the frequency of the CPG does not linearly adapt to the input frequency of the FSR. It rather moves in a narrow frequency range (between 5 and 20 Hz) which is enough to be able to select and modify the gait pattern of the robotic platform.

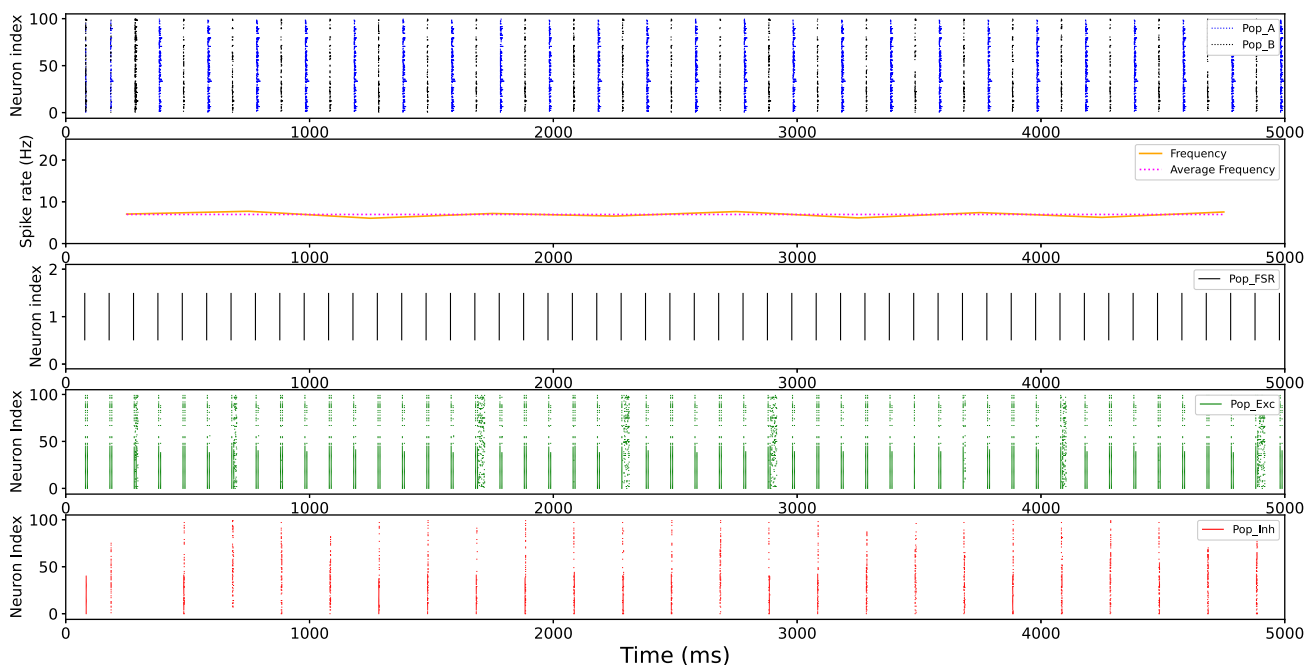
The second experiment includes, as stimulus, intermediate frequencies of 85 Hz from the population bound to the FSR. The results are shown in **Figure 8**. In this experiment, it can be observed how the activity of the POP<sub>EXC</sub> population is greatly

increased by the excitation received by POP<sub>FSR</sub>. It is interesting to highlight how, although increasing the frequency of POP<sub>FSR</sub> should decrease the frequency of POP<sub>INH</sub>, the opposite effect occurs due to the increase in the oscillation frequency of POP<sub>CPG</sub>, which excites, albeit weakly, POP<sub>INH</sub>. The average oscillation frequency of the CPG is 13 Hz.

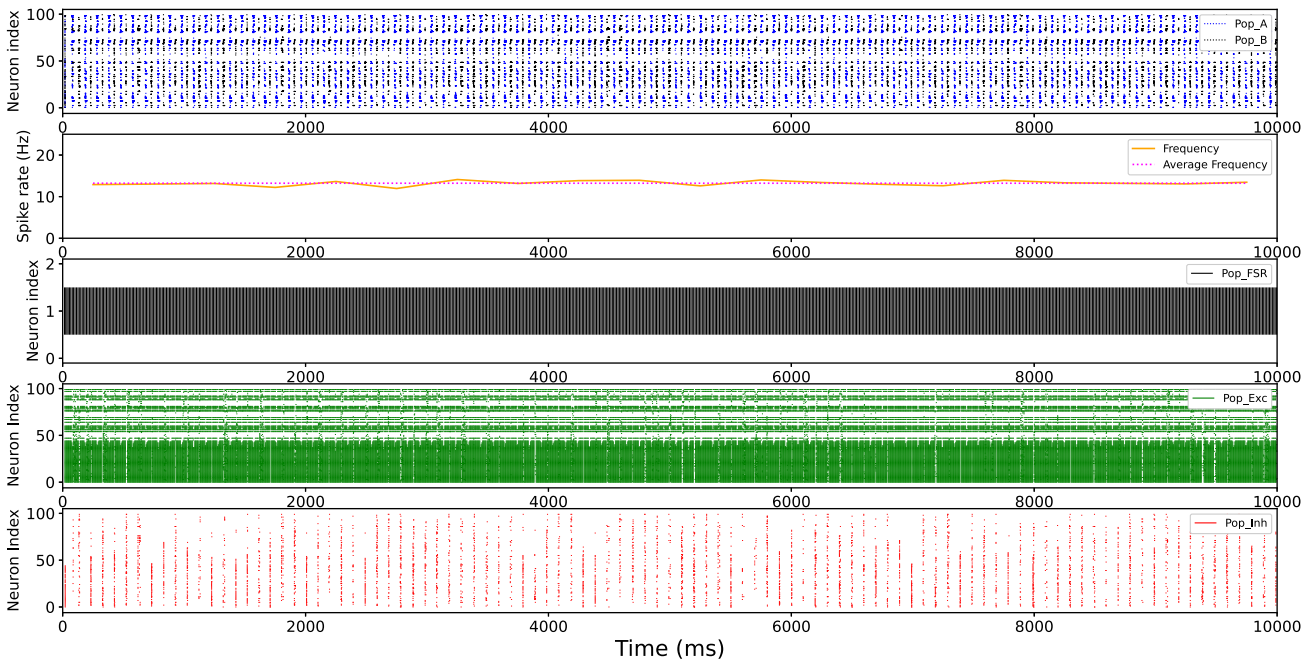
As a third experiment, the CPG network was subjected to constant inputs of 171 Hz (maximum values provided by the FSRs) from the population bound to them. The results are shown in **Figure 9**. In this case, it can be observed how POP<sub>EXC</sub> is strongly excited by POP<sub>FSR</sub>, while the activity of POP<sub>INH</sub> is considerably reduced. In this scenario, not only does the oscillation frequency of POP<sub>CPG</sub> increase, which is around 20 Hz, but also the number of shots at a given time instant in each population of POP<sub>CPG</sub> increases.

Finally, a last experiment has been performed in which the input linked to the FSR is oscillatory, and it can be observed how the CPG adapts to the input by increasing or decreasing its total frequency (the frequency obtained by considering both populations of the CPG) and its individual frequency (the frequency of each neuron within each population at individual time points when its population is firing). The results are shown in **Figure 10**.

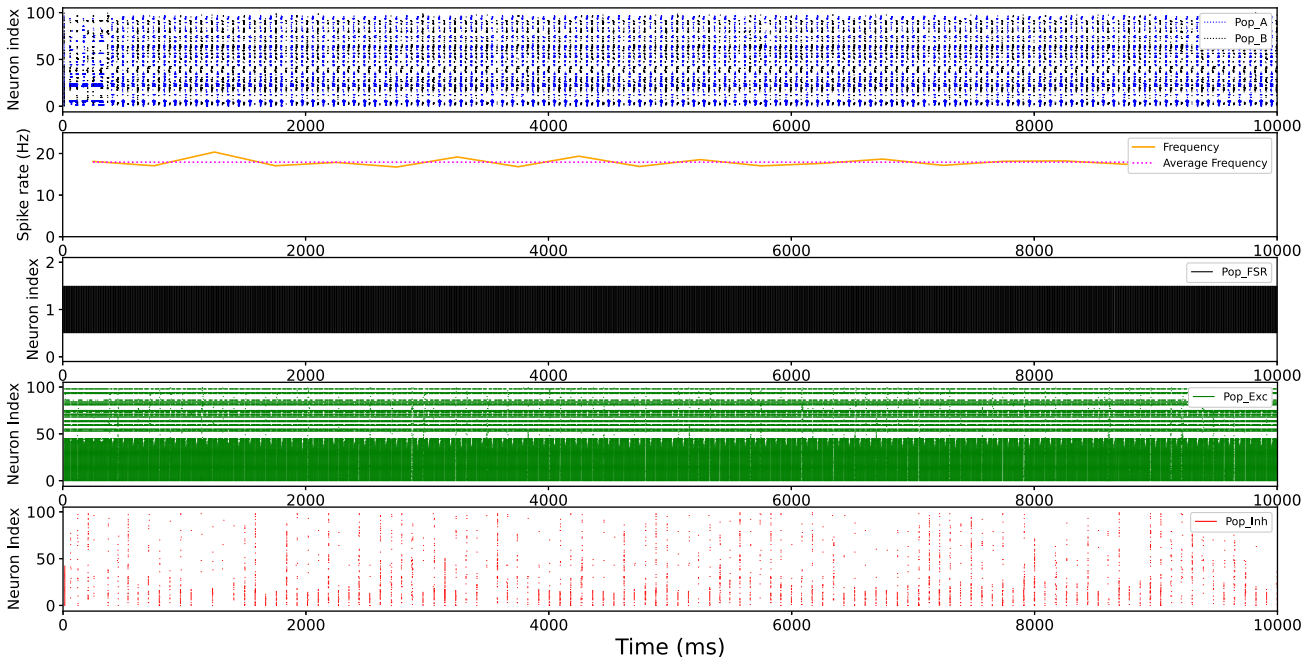
As mentioned previously, the WTA network will receive all AER addresses associated with retinal pixels that have detected changes in luminosity. It is possible to receive up to 20 000 addresses per second on SpiNNaker, each representing one of the 128 possible directions linked to the X-coordinate of the retina. Various experiments have been conducted to subject the network to different stimuli, using a pen to create motion in front of the retina and activate the pixels. It is important to note that these scenarios propose situations that will not occur in a real and



**Figure 7.** Behavior of the CPG network at 10 Hz inputs. Thanks to the regulating action of the POP<sub>EXC</sub> and POP<sub>INH</sub> populations, the output frequency of the CPG<sub>AB</sub> remains stable around 7 Hz.



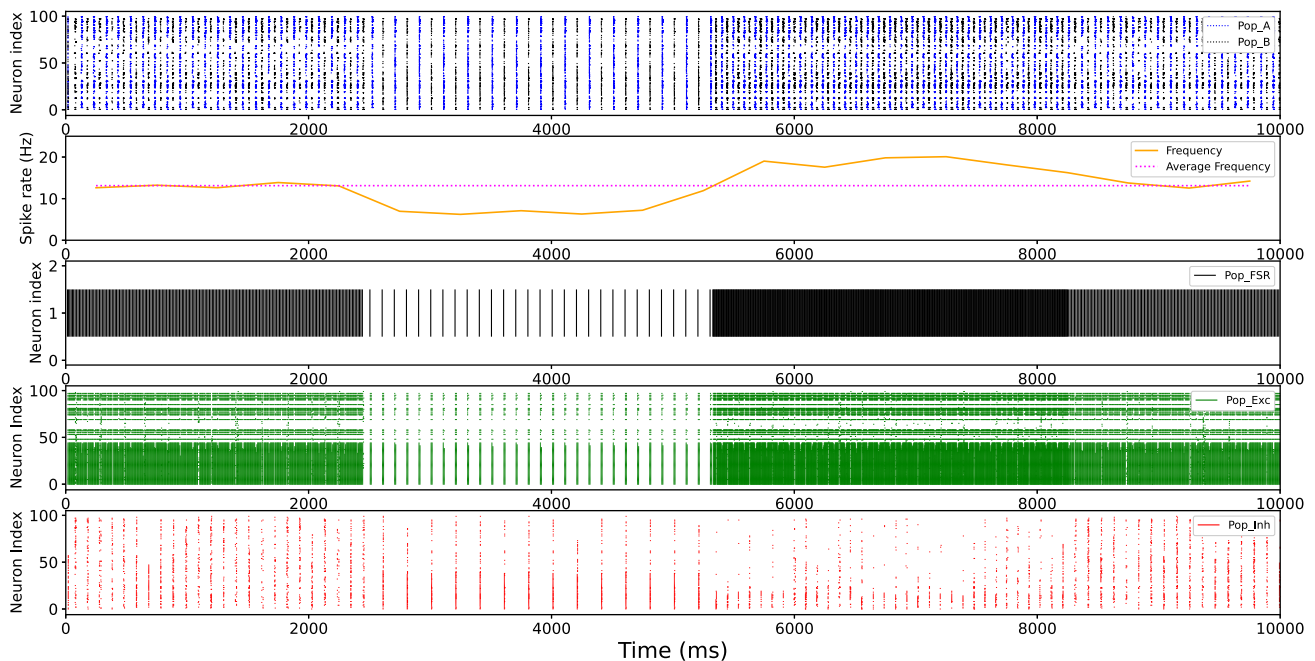
**Figure 8.** Behavior of the CPG network at 85 Hz inputs. Intermediate frequencies cause a generalized increase in oscillations in all populations, while POP<sub>INH</sub> and POP<sub>EXC</sub> keep the CPG stable throughout the duration of the experiment.



**Figure 9.** Behavior of the CPG network in response to 171 Hz inputs. It is interesting to observe how the significant excitation caused by POP<sub>EXC</sub> on the CPG not only increases its frequency to 20 Hz but also leads to an increase in the number of neurons firing in each population of the CPG during the moments when they are active.

controlled environment of the robotic platform, such as receiving similar changes in luminosity in more than one window simultaneously or introducing noise into one of the windows to assess the network's behavior in response to these phenomena.

First, a higher activity was generated in the left region of the DVS, where it can be observed that during the entire simulation the stimulus was generated intermittently in the POP<sub>LEFT</sub> region intermittently, generating peaks of minimum activity (e.g.,



**Figure 10.** Behavior of the CPG network in response to oscillatory inputs. The CPG adjusts its frequency based on the input associated with the FSRs, demonstrating its adaptation to external stimuli.

$t = 1000$ ,  $t = 2000$ , and  $t = 4000$  ms) in some instants due to the stimulus coming out of the visible range of the retina. The results are shown in **Figure 11**.

Another experiment was also carried out in which more activity was generated in the central region of the DVS. The results can be seen in **Figure 12**, and it is possible to check that the detection of events in all the windows does not imply activity in all of them at the output of the WTA network. The activity will be maintained only in the population that has generated the highest number of events in a given time.

The same experiment was also performed generating increased activity in the right region of the DVS. In this experiment, it is again verified that, although the central population has produced some events in the instants  $t = 800$  and  $t = 3000$  ms, the output of the WTA network only contemplates the events of the winning population. The results can be seen in **Figure 13**.

Finally, an experiment has also been performed in which the WTA network is subjected to random activity over the entire visual field of the DVS. The result of this experiment can be seen in **Figure 14**. It is interesting to see how the network shows tolerance to instantaneous noise. For example, although at  $t = 1200$  ms, the winning population at that instant is  $POP_{Left}$ , at the output of the WTA  $POP_{Center}$  remains the winner because  $POP_{Left}$  has not generated enough activity for long enough to exceed the activity previously generated by  $POP_{Center}$ .

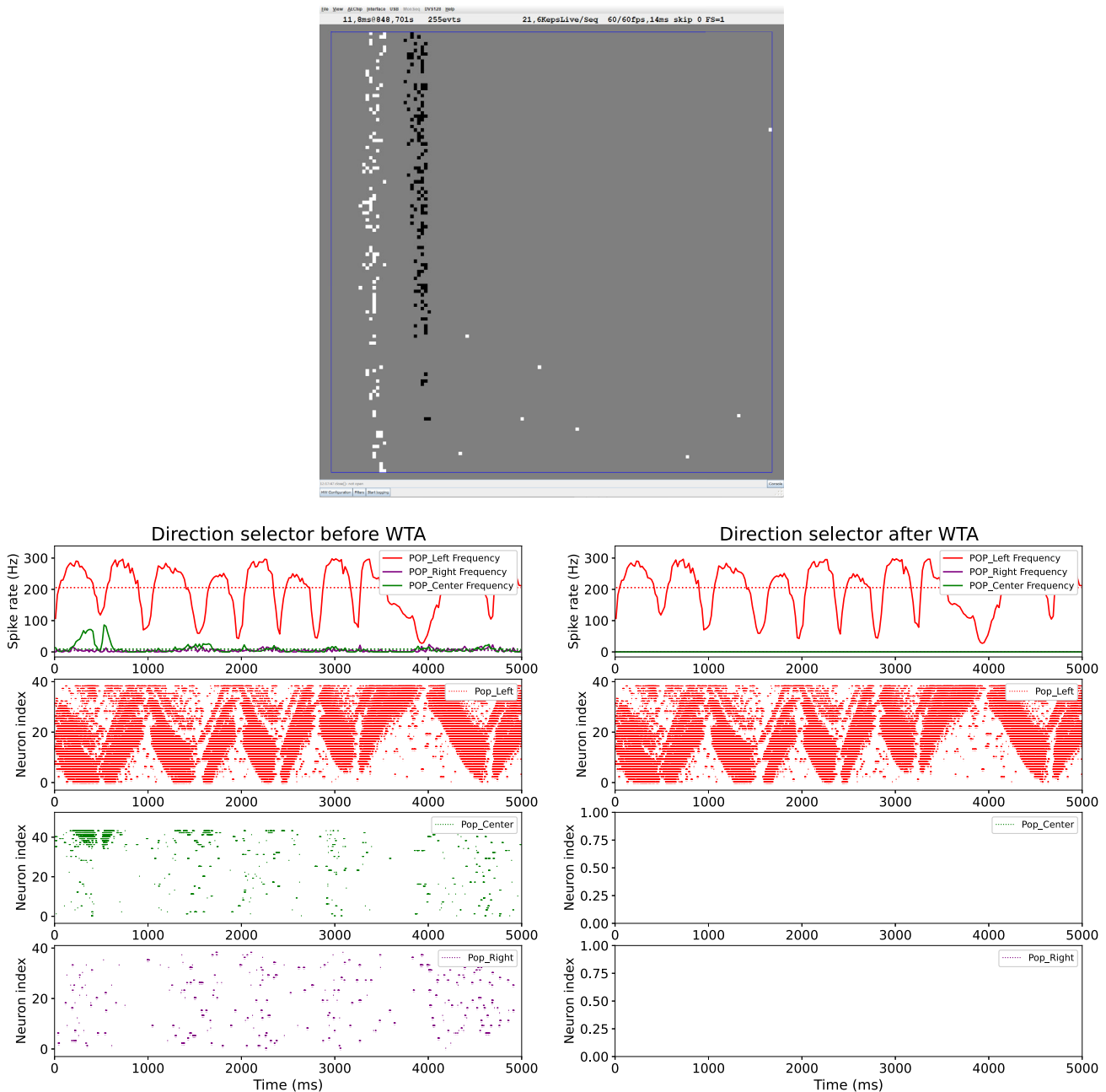
## 4. Discussion

The set of experiments conducted demonstrates: 1) the ability of the sCPG model to adapt to variable inputs from the FSRs by changing its oscillation frequency to increase or decrease the

speed of the robotic platform, respectively; and 2) the ability of the WTA network model to select the spatial region with the highest spiking activity from a DVS input, thus orienting the robotic platform toward that direction. Thanks to these two features, course and speed adaptation, it is possible to implement a fully functional “prey–predator” model where analog and neuromorphic sensor inputs are fused. This information can be used by the network implemented in SpiNNaker to make a decision about the behavior of the robotic platform.

From a bioinspired point of view, the biological features of the CPG (described in Section 1) show the functions that these neuronal structures can perform in living beings, as well as the characteristics of the neurons that form this type of neural structure. The CPG design includes biologically plausible structures.

In the biological model, the neurons that make up the CPG can be either tonic or phasic. In the former case, tonic neurons emit the neurotransmitter at a constant but low rate, whereas phasic neurons act almost instantly, emitting a rapid pulse of activity to bring postsynaptic neurons into action quickly. In this work, both types of neurons have been implemented: on the one hand,  $CPG_{AB}$  is made up entirely of tonic neurons. They include a very low (0.1 nA) but constant current injection, which ensures that even if no stimulus was received from the FSRs, the oscillation of the CPGs could be maintained over time, fulfilling the intrinsic rhythm generation that biological CPGs show. On the other hand, the  $POP_{Exc}$  and  $POP_{Inh}$  populations act as phasic neurons, since at a minimum excitation signal received, they are able to reach the threshold to produce a neuronal firing: exciting or inhibiting the  $CPG_{AB}$  almost instantly. With this feature, the centralized control to which the CPG is subjected is modeled. Although CPGs can function autonomously, they are often under



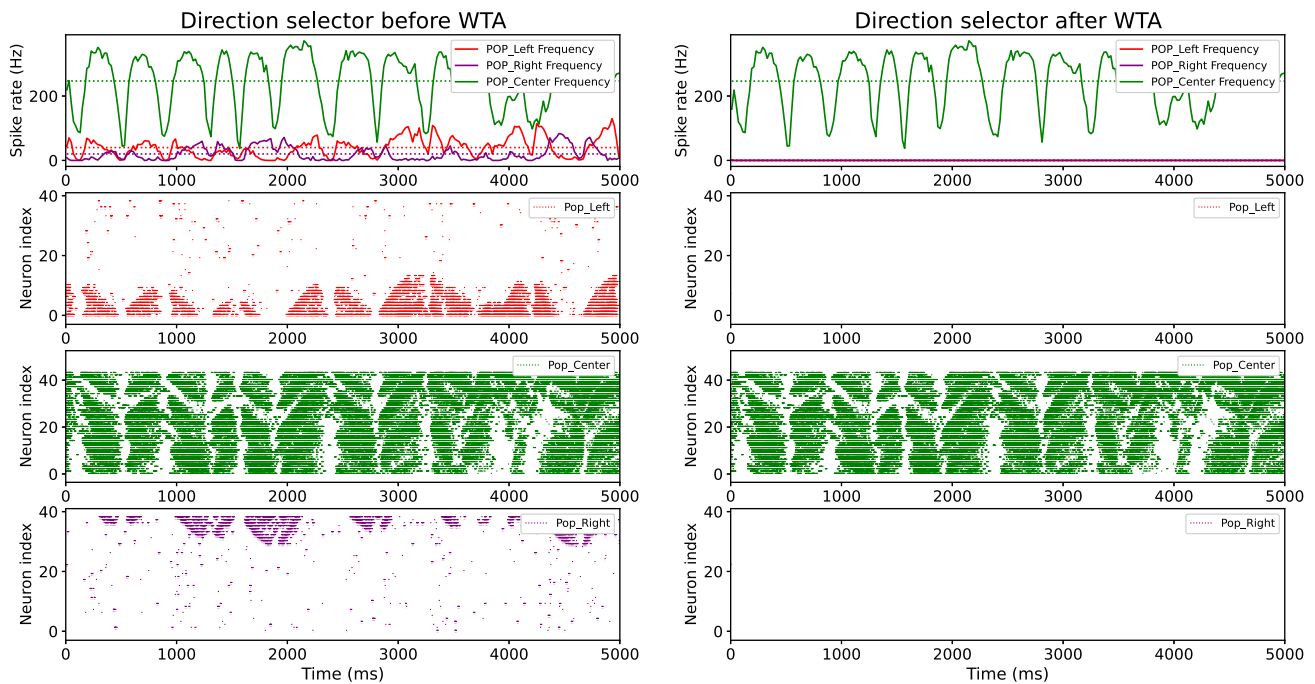
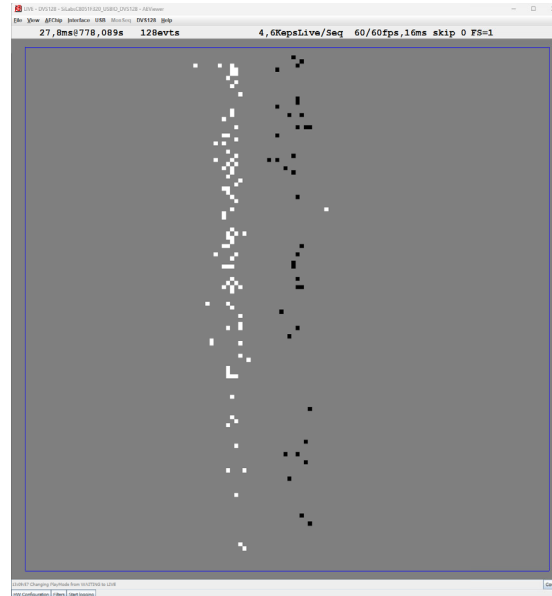
**Figure 11.** Behavior of the WTA network in response to events generated in the left region.

control of the central nervous system to adapt to the needs of the surrounding environment.

Another feature of biological CPGs is their plasticity, which means that they can adapt and change over time in response to different signals. In this work, the CPG implemented can adapt to the conditions of its environment just as any biological being would do. The FSRs modify the oscillation frequency of the CPG, which results in a regulation of the gait of the robotic platform depending on the terrain. This functioning is similar to that of the vestibular system of living beings. Thanks to this behavior, the condition of variability presented

by the biological CPGs, which adapt their patterns in response to different conditions, is also fulfilled. To complete the CPG model, the DVS is used in the network to select the orientation to be followed by the robotic platform, imitating the decision-making capacity that living beings have when taking a course based on what their optical system detects (prey, dangers, obstacles, etc.).

However, there are also features of the CPG presented in this work that may differ with respect to biological ones. The number of neurons used in this work has been estimated only with the stability of the whole system as a priority, reducing almost to zero

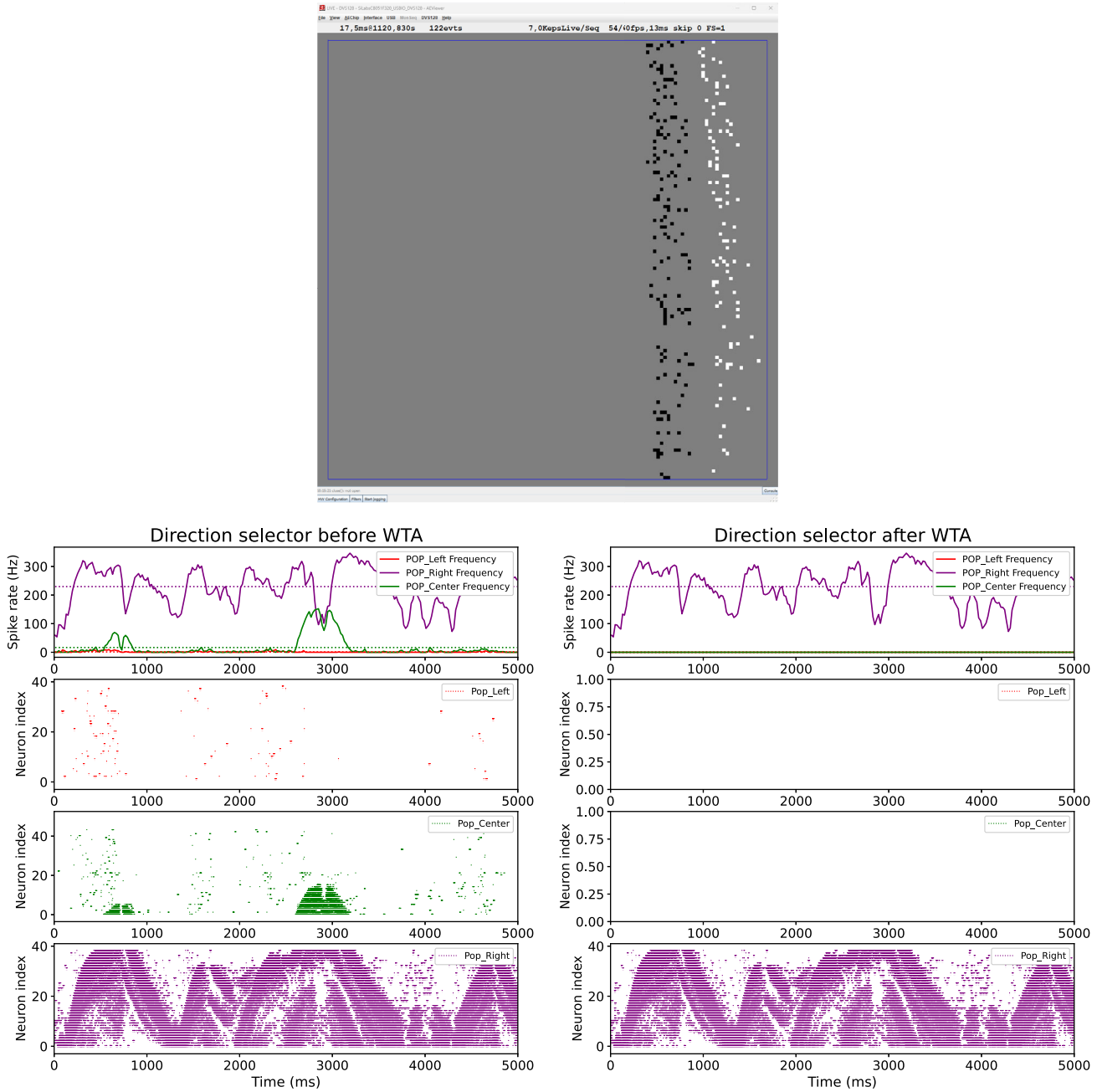


**Figure 12.** Behavior of the WTA network in response to events generated in the central region.

the scenarios in which the network output is unstable or noncoherent. In biological CPGs focused on locomotion, a relatively complex neural network is required (depending on factors such as species or movement pattern), but it is estimated that in mammals, such as humans, these CPGs can be composed of thousands of neurons.<sup>[2]</sup> Furthermore, while in this work the CPG is only responsible for adjusting the speed of the robotic platform, in the biological environment, these neural structures are used simultaneously to control speed, direction, balance, and other aspects of locomotion according to the specific needs of the

organism and environmental conditions, without being the only ones involved in this task.<sup>[1,7]</sup>

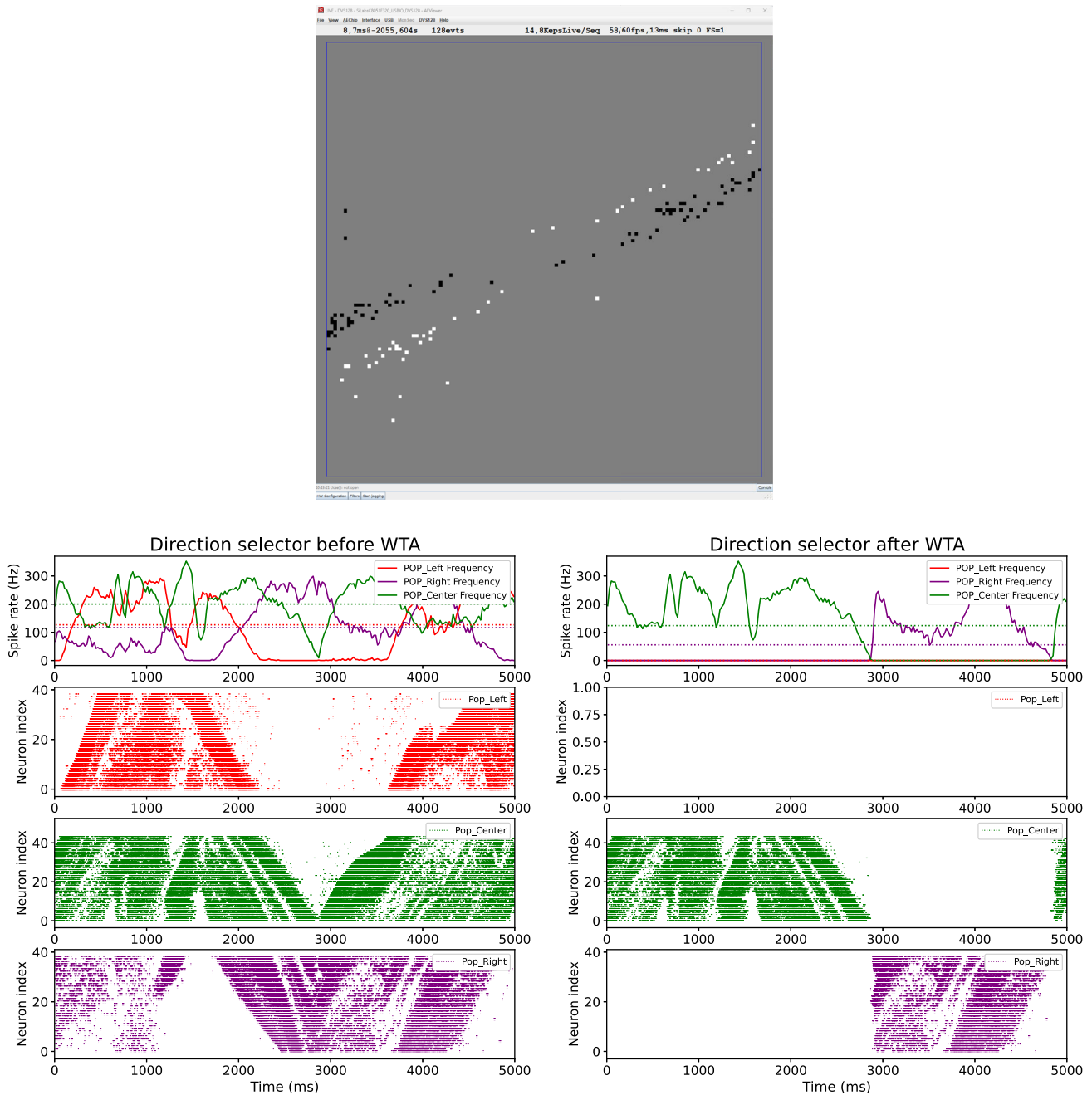
Another important aspect to mention, which is related to the previous one, is the use of a WTA network for decision making regarding the orientation of the robotic platform. While these networks may model certain aspects of competition and inhibition observed in some brain functions, they do not directly or specifically resemble a particular biological neural structure of the nervous system, leaving this function in the biological environment reserved, again, for the CPGs.<sup>[54]</sup>



**Figure 13.** Behavior of the WTA network in response to events generated in the right region.

Finally, in terms of computational resources and power consumption, the network presented in this work uses a total of 579 neurons with around 200.000 synapses, while other works such as refs. [20,31] use a much smaller number of neurons to control the robotic platform (both of these works used 12 neurons to build the entire network). Due to the high number of neurons used, more power consumption is expected to train and run the network. However, this system is potentially more efficient than those based on ANNs. This greater efficiency is mainly due to the fact that its neurons only activate in response

to a specific stimulus, thereby reducing the required computational calculation. In our system, this translates into lower energy consumption when processing sensory inputs. Particularly, by using the WTA network to process signals from a DVS, we observed reduced energy consumption compared to equivalent processing by ANNs. This is primarily due to the sporadic and event-based nature of the DVS, which consumes energy only when its pixels detect changes in brightness. In contrast, conventional cameras process frames constantly, implying higher energy consumption.



**Figure 14.** Behavior of the WTA network in response to randomly generated events.

## 5. Conclusion

In this article, a cCPG model implemented on SpiNNaker that receives information from analog and neuromorphic sensors simultaneously is proposed. The architecture presented includes a closed-loop control to adapt locomotion frequencies and orientation that can be applied to any legged robot. The information from the environment provided by the FRSs and the DVSs is processed instantaneously by SpiNNaker, which generates a frequency output that is sent to the FPGAs and eventually used to control the actuators.

The system has been validated using data from irregular surfaces. These data generate pressure differences on the FSRs, which in turn cause variations in the voltage emitted by these sensors. The obtained values are converted into spikes, which are received and processed by SpiNNaker. Subsequently, SpiNNaker adjusts the frequency of the CPG and the gait pattern of the robotic platform based on these signals.

Furthermore, the proper operation of the WTA network to determine the direction of the robotic platform should follow has been verified. The WTA network is based on activity detected

in various regions of the DVS. The results of the experiments show that the WTA network effectively selects the winning region, completely excluding the other two options. Similar to the CPG, the output of the WTA network is sent to the FPGA, which in turn adjusts the orientation of the robotic platform based on this information.

In future work, the architecture presented will be tested in robots with different leg configurations, which will allow us to demonstrate its reproducibility on other robotic platforms, regardless of the number of legs they have. Furthermore, the DVS will be used not only to follow the direction where the highest activity is, but also to incorporate into the network a learning routine that allows it to recognize specific shapes or patterns, which would induce the robotic platform to pursue a specific target by recognizing its shape and not only the generated activity. In addition, other neuromorphic sensors such as the NAS will be included in the loop to bring robotic behavior closer to biology.

## Acknowledgements

This research was partially supported by project PID2019-105556GB-C33 funded by MCIN/AEI/10.13039/501100011033.

## Conflict of Interest

The authors declare no conflict of interest.

## Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## Keywords

adaptative learning, central pattern generators, neuromorphic hardware, neurorobotics, spiking neural networks, SpiNNaker

Received: October 9, 2023

Revised: December 19, 2023

Published online: February 17, 2024

- [1] S. Grillner, P. Wallen, *Annu. Rev. Neurosci.* **1985**, 8, 233.
- [2] M. MacKay-Lyons, *Phys. Ther.* **2002**, 82, 69.
- [3] G. M. Hughes, C. A. Wiersma, *J. Exp. Biol.* **1960**, 37, 657.
- [4] R. M. Harris-Warrick, *Prog. Brain Res.* **2010**, 187, 213.
- [5] P. A. Guertin, *Front. Neurol.* **2013**, 3, 183.
- [6] S. L. Hooper, *Curr. Biol.* **2000**, 10, R176.
- [7] E. Marder, D. Bucher, *Curr. Biol.* **2001**, 11, R986.
- [8] A. J. Ijspeert, *Neural networks* **2008**, 21, 642.
- [9] J. Kahn, A. Roberts, *Philos. Trans. R. Soc. London, Ser. B* **1982**, 296, 229.
- [10] E. A. Arbas, R. L. Calabrese, *J. Comp. Physiol. A* **1984**, 155, 783.
- [11] C. Von Euler, *J. Appl. Physiol.* **1983**, 55, 1647.
- [12] C. Mead, *Nat. Electron.* **2020**, 3, 434.
- [13] A. Linares-Barranco, F. Perez-Peña, A. Jimenez-Fernandez, E. Chicca, *Front. Neurobot.* **2020**, 14, 590163.
- [14] D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, A. Linares-Barranco, *Neurocomputing* **2020**, 381, 10.
- [15] Y. Kim, Y. Li, H. Park, Y. Venkatesha, A. Hambitzer, P. Panda, in *Proc. of the AAAI Conf. on Artificial Intelligence*, Vol. 37, Washington D.C., February **2023**, pp. 8308–8316.
- [16] G.-I. Uleru, M. Hulea, A. Barleanu, *Biomimetics* **2023**, 8, 28.
- [17] P. Lopez-Osorio, A. Patiño-Saucedo, J. P. Dominguez-Morales, H. Rostro-Gonzalez, F. Perez-Peña, *Neurocomputing* **2022**, 502, 57.
- [18] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. V. Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Häfliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, K. Boahen, *Front. Neurosci.* **2011**, 5, 73.
- [19] L. Righetti, A. J. Ijspeert, in *2008 IEEE Int. Conf. on Robotics and Automation*, IEEE, Pasadena, CA, May **2008**, pp. 819–824.
- [20] H. Rostro-Gonzalez, P. A. Cerna-Garcia, G. Trejo-Caballero, C. H. Garcia-Capulin, M. A. Ibarra-Manzano, J. G. Avina-Cervantes, C. Torres-Huitzil, *Neurocomputing* **2015**, 170, 47.
- [21] B. Cuevas-Arteaga, J. P. Dominguez-Morales, H. Rostro-Gonzalez, A. Espinal, A. F. Jimenez-Fernandez, F. Gomez-Rodriguez, A. Linares-Barranco, in *Advances in Computational Intelligence: 14th Int. Work-Conf. on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14–16, 2017, Proceedings, Part I 14*, Springer, Cádiz, Spain, June **2017**, pp. 548–559.
- [22] S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, *Proc. IEEE* **2014**, 102, 652.
- [23] B. Strohmmer, P. Manoonpong, L. B. Larsen, *Front. Neurobot.* **2020**, 14, 41.
- [24] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, H. Wang, *IEEE Micro* **2018**, 38, 82.
- [25] E. Angelidis, E. Buchholz, J. Arreguit, A. Rougé, T. Stewart, A. von Arnim, A. Knoll, A. Ijspeert, *Neuromorph. Comput. Eng.* **2021**, 1, 014005.
- [26] G. Endo, J. Morimoto, T. Matsubara, J. Nakanishi, G. Cheng, *Int. J. Rob. Res.* **2008**, 27, 213.
- [27] G. Li, H. Zhang, J. Zhang, R. T. Bye, *Adv. Rob.* **2014**, 28, 389.
- [28] S. A. Kurkin, D. D. Kulminskiy, V. I. Ponomarenko, M. D. Prokhorov, S. V. Astakhov, A. E. Hramov, *Chaos* **2022**, 3232, 033117.
- [29] M. Wang, J. Yu, M. Tan, *Int. J. Adv. Rob. Syst.* **2014**, 11, 170.
- [30] D. Korkmaz, G. Ozmen Koca, G. Li, C. Bal, M. Ay, Z. H. Akpolat, *J. Mar. Eng. Technol.* **2021**, 20, 125.
- [31] A. Spaeth, M. Tebyani, D. Haussler, M. Teodorescu, in *2020 3rd IEEE Int. Conf. on Soft Robotics (RoboSoft)*, IEEE, New Haven, CT, May **2020**, pp. 46–51.
- [32] E. M. Izhikevich, *IEEE Trans. Neural Networks* **2004**, 15, 1063.
- [33] D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, A. Linares-Barranco, in *2019 IEEE Int. Symp. on Circuits and Systems (ISCAS)*, IEEE, Sapporo, Hokkaido, Japan, May **2019**, pp. 1–1.
- [34] T. Schoepe, D. Gutierrez-Galan, J. P. Dominguez-Morales, H. Greatorex, A. F. Jiménez Fernández, A. Linares-Barranco, E. Chicca, *Neuromorph. Comput. Eng.* **2023**, 3, 024009.
- [35] B. Strohmmer, R. K. Stagsted, P. Manoonpong, L. B. Larsen, *Front. Neurosci.* **2021**, 15, 633945.
- [36] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, A. D. Brown, *IEEE Trans. Comput.* **2012**, 62, 2454.
- [37] S. Furber, P. Bogdan, *SpiNNaker: A Spiking Neural Network Architecture*, now publishers, Boston-Delft **2020**.
- [38] H. Markram, *Sci. Am.* **2012**, 306, 50.

- [39] E. Painkras, L. A. Plana, J. Garside, S. Temple, F. Galluppi, C. Patterson, D. R. Lester, A. D. Brown, S. B. Furber, *IEEE J. Solid-State Circuits* **2013**, 48, 1943.
- [40] L. A. Plana, S. B. Furber, S. Temple, M. Khan, Y. Shi, J. Wu, S. Yang, *IEEE Des. Test Comput.* **2007**, 24, 454.
- [41] M. Mahowald, Ph.D. Dissertation, California Institute of Technology Pasadena **1992**.
- [42] L. A. Plana, J. Garside, J. Heathcote, J. Pepper, S. Temple, S. Davidson, M. Luján, S. Furber, *IEEE Access* **2020**, 8, 84918.
- [43] A. Yousefzadeh, M. Jabłoński, T. Iakymchuk, A. Linares-Barranco, A. Rosado, L. A. Plana, S. Temple, T. Serrano-Gotarredona, S. B. Furber, B. Linares-Barranco, *IEEE Trans. Biomed. Circuits and Syst.* **2017**, 11, 1133.
- [44] J. P. Dominguez-Morales, A. Jimenez-Fernandez, A. Rios-Navarro, E. Cerezuela-Escudero, D. Gutierrez-Galan, M. J. Dominguez-Morales, G. Jimenez-Moreno, in *Int. Conf. on Artificial Neural Networks*, Springer, Barcelona, Spain, September **2016**, pp. 45–53.
- [45] T. Schoepe, D. Gutierrez-Galan, J. P. Dominguez-Morales, A. Jimenez-Fernandez, A. Linares-Barranco, E. Chicca, in *2019 IEEE Biomedical Circuits and Systems Conf. (BioCAS)*. IEEE, Nara, Japan, October **2019**, pp. 1–4.
- [46] A. P. Davison, D. Brüderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, P. Yger, *Front. Neuroinf.* **2009**, 2, 11.
- [47] O. Rhodes, P. A. Bogdan, C. Brennkmeijer, S. Davidson, D. Fellows, A. Gait, D. R. Lester, M. Mikaitis, L. A. Plana, A. G. Rowley, A. B. Stokes, S. B. Furber, *Front. Neurosci.* **2018**, 12, 816.
- [48] Digilent, <https://digilent.com/reference/programmable-logic/nexys-4-ddr/reference-manual>, visitado el 8 de septiembre de **2023**.
- [49] P. Lichtsteiner, C. Posch, T. Delbruck, *IEEE J. Solid-State Circuits* **2008**, 43, 566.
- [50] F. Perez-Peña, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gomez-Rodriguez, G. Jimenez-Moreno, J. Lopez-Coronado, *Sensors* **2013**, 13, 15805.
- [51] A. Linares-Barranco, F. Gomez-Rodriguez, A. Jimenez-Fernandez, T. Delbruck, P. Lichtensteiner, in *2007 IEEE Int. Symp. on Circuits and Systems*, New Orleans, LA, May **2007**, pp. 1192–1195.
- [52] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gómez-Rodríguez, L. Camuñas-Mesa, R. Berner, M. Rivas-Pérez, T. Delbruck, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. C. Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, B. Linares-Barranco, *IEEE Trans. Neural networks* **2009**, 20, 1417.
- [53] Ax-12a - Robotis e-manual, <https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/> (accessed: August 2023).
- [54] R. Coultrip, R. Granger, G. Lynch, *Neural Networks* **1992**, 5, 47.