NeuroPod: a real-time neuromorphic spiking CPG applied to robotics

Daniel Gutierrez-Galan, Juan P. Dominguez-Morales, Fernando Perez-Peña, Ángel Francisco Jiménez Fernández, Alejandro Linares-Barranco

[a]*Robotics and Computer Technology Lab. Universidad de Sevilla, Spain.*
[b]*Department of Computer Architecture and Technology, Universidad de Cádiz, Spain.*

Abstract

Initially, robots were developed with the aim of making our life easier, carrying out repetitive or dangerous tasks for humans. Although they were able to perform these tasks, the latest generation of robots are being designed to take a step further, by performing more complex tasks that have been carried out by smart animals or humans up to date. To this end, inspiration needs to be taken from biological examples. For instance, insects are able to optimally solve complex environment navigation problems, and many researchers have started to mimic how these insects behave. Recent interest in neuromorphic engineering has motivated us to present a real-time, neuromorphic, spike-based Central Pattern Generator of application in neurorobotics, using an arthropod-like robot. A Spiking Neural Network was designed and implemented on SpiNNaker. The network models a complex, online-change capable Central Pattern Generator which generates three gaits for a hexapod robot locomotion. Reconfigurable hardware was used to manage both the motors of the robot and the real-time communication interface with the Spiking Neural Networks. Real-time measurements confirm the simulation results, and locomotion tests show that NeuroPod can perform the gaits without any balance loss or added delay.

Keywords: Neurorobotics, SpiNNaker, Central Pattern Generator, Spiking Neural Network, Neuromorphic Hardware, FPGA

Email address: dgutierrez@atc.us.es (Daniel Gutierrez-Galan)

1. Introduction

A Central Pattern Generator (CPG) is a neural structure located at a spinal cord level. It can generate rhythm patterns which might be used for movements, such as the generation of various gaits, or swimming [1]. There is proven evidence of such structures in small animals [2] and possibly in humans [3, 4]. The activity of these structures is released and mediated by the brain stem and other sub-cortical regions of the brain. Regarding the feedback, the CPG receives sensory information to adapt its output to the environment.

Locomotion is probably one of the most complex tasks to be developed by roboticists due to stability issues when several legs are involved [5]. Therefore, from a neurorobotics point of view, the idea of these CPGs is borrowed from biology to implement locomotion in small robots with several legs. The reason for this is that these structures can generate a very stable pattern even without sensory information or brain activity. In fact, some cats that suffered severe spinal cord injuries, recovered their gaits after treadmill training sessions [6]. In this paper, we borrow this CPG feature: the ability to generate patterns in an open-loop manner. This can be useful as a first approach for the use of CPGs in neurorobotics, which is the target field of this paper.

There are many works where a CPG has been used within robotics; some of them mimic the idea of a CPG, although without implementing a spiking neural network. Instead, they modelled the CPG using differential equations of coupled oscillators. Examples are: [7], where the authors used a hexapod robot and they included feedback, [8], where the Van der Pol oscillator model was used and implemented on a Field Programmable Gate Array (FPGA), and [9], where the authors used a swimming and crawling fish robot and implemented the CPG on a microcontroller by solving the equations of coupled oscillators.

More closely related works, where neuromorphic hardware was used or a Spiking Neural Network (SNN) was proposed, are: [10], where the authors developed an analog neuromorphic dedicated chip which allocates coupled oscillators and a learning procedure to have the desired output, although they did not use well-known neuron models, and [11], in which the authors designed and implemented several CPGs segments to drive a lamprey-like robot. The CPGs were implemented using neuromorphic hardware in [12], although online changes of the pattern generated by the CPG are not pro-

vided. Likewise, the work presented in [13] proposed the implementation of the CPG using a SNN implemented on SpiNNaker [14], although it does not offer real time nor online change of the pattern produced by the CPG.

The work presented in this paper is based on the one presented in [13]. The objective is to implement a CPG closely related to its biological counterpart including plausible biological features in SpiNNaker and ready to be used within robotics: a hexapod robot was used to validate the design. The novelties of this research are: real-time operation of the CPG and online reconfiguration of the gait produced by the CPG.

This paper is structured as follows: subsection 2.1 describes the materials used and subsection 2.2 describes how the research was conducted and all the details needed to replicate the experiments. Then, the results achieved by the implementation of the CPG in a small robotic platform using neuromorphic hardware are described in section 3. Finally, section 4 presents a discussion over the obtained results and the conclusions drawn from this study.

2. Materials and Methods

This section describes the materials that were used in this project to design, assemble, and control this neuromorphic robot, as well as the methods applied to obtain the results shown in section 3.

2.1. Materials

The NeuroPod robot is divided into three main parts, and each of these has a specific role or functionality. These parts are the CPG, designed using a SNN and implemented on a neuromorphic hardware platform. This CPG generates the gait patterns. The movement controller takes the movement information from the CPG and controls a set of servomotors through an FPGA-based board. Finally, the skeleton defines the shape of the robot and also performs the movements. Further details are provided in the following sections. Figure 1 shows a global overview of the NeuroPod as a block diagram; it also shows the main parts and how they interface with each other.

2.1.1. Robotic platform

An hexapod robot is a six-legged robot inspired by arthropod insects, such as ants or flies, among others. According to biology, the body of these insects can be divided into three different regions: the head, the thorax

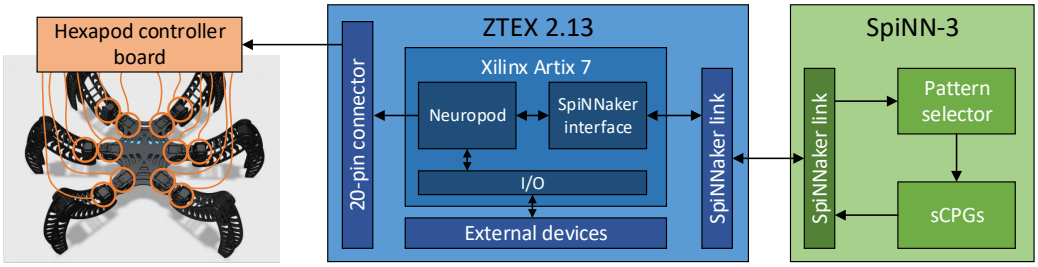

Figure 1: Block diagram of the entire system. It is composed of the SpiNNaker board, an FPGA-based board, and a 3D-printed hexapod robot frame.

and the abdomen. Moreover, each part could be sub-divided in segments according to their features.

The head is composed of eyes (located in the ocular segment) and a pair of antennae. Both the eyes and the antennae are used to collect sensory information about the environment, allowing the movement of the insect in complex scenarios by performing an obstacle avoidance task [15][16]. The thorax is composed of three segments: the prothorax, the mesothorax and the metathorax. Each segment has a pair of legs, and there are six in total. Up to five parts can be identified in each leg, although only three of these parts are relevant to motion: coxa, femur and tibia. Finally, the abdomen contains the vital organs of the insect, such as the respiratory or reproductive systems.

Recent focus on the development of smart robots by mimicking biological processes has motivated many research groups to develop accurate models of hexapod insects. HECTOR [17] is an example of that, where both the body features and the movements were inspired by the morphological details of the stick insect *Carausius morosus*.

In this work, a 3D-printed hexapod robot was used, based on the model featured in [13]. The original design[1] was adapted by designing a new body frame to allocate the electronic devices on it. The frame dimensions are 20 x 89 x 90 mm (height, width, depth), without the legs.

According to [18], insect legs are defined as multi-segmented limbs. Each leg consists of more than 5 segments, as it is represented in Fig.2A. However, only three of them are used when performing a movement: the coxa, the

[1]https://www.thingiverse.com/thing:1021540 (checked on April'2019)

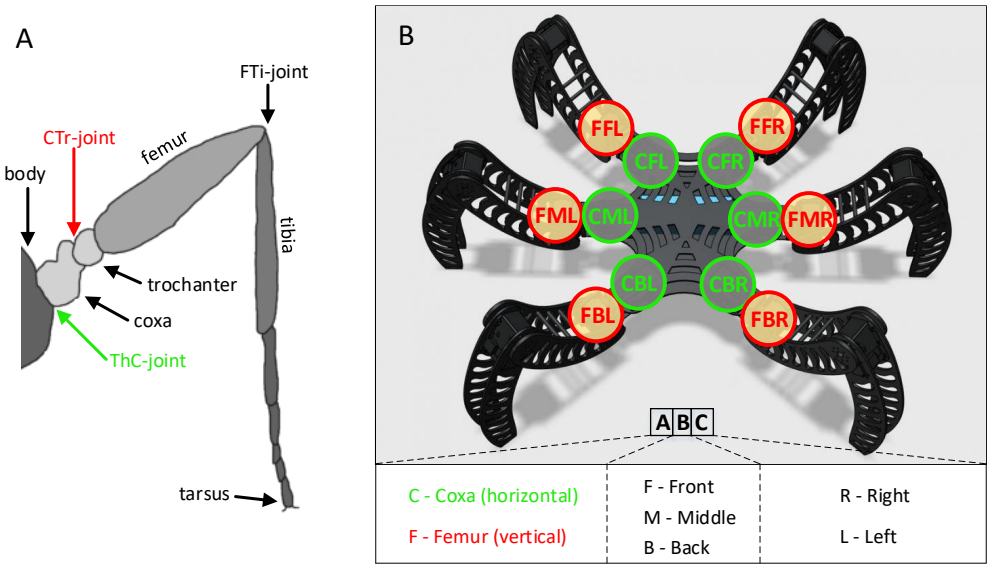

Figure 2: A) Biological representation of an arthropod's leg anatomy. B) Hexapod robot leg actuator IDs.

femur and the tibia. This is due to the fact that three main leg joints can be found in an insect leg: the thoraco-coxal (ThC-) joint, the coxa-trochanteral (CTr-) joint and the femur-tibia (FTi-) joint. The ThC-joint is responsible for carrying out back and forth movements (horizontal axis), the CTr-joint enables elevation and depression and the FTi-joint allows extension and flexion (both in the vertical axis).

The leg of each hexapod has three degrees of freedom (DOF), one per joint. However, to develop NeuroPod we only considered two of them because the movement of the robot can be performed mainly using the coxa and the femur [19]. Thus, only twelve DOF were taken into account, instead of eighteen, to implement the gait patterns.

In order to provide motion, one servomotor was placed on each joint, making a total of twelve servomotors (Ref. SG90). The maximum rotation angle is 180 degrees, although this range could be reduced due to mechanical constraints of the body design and the position of the servo on it.

Therefore, a calibration is required. After the calibration process, and knowing that the theoretical operation speed of the selected servos is 0.12 s/60 degrees, we were able to estimate the pattern period, which can be defined as the minimum time the robot needs to reach the backward position,

starting from the forward position, and then reach the forward position again. Measurements of these pattern periods are presented in section 3.

2.1.2. SpiNNaker

The SpiNNaker project is based on a massive parallel multicore computing system that is able to run very large SNNs in real time [20]. The architecture of the SpiNNaker chip, which has an asynchronous packet switching network, makes it very efficient for neuromorphic applications [21].

In this work, the SpiNN-3 machine (4 SpiNNaker chips, 72 200MHz ARM9 cores) was used to implement the SNN model, which is described in section 2.2. The device is shown in Fig. 3. This board has an interface, 100 Mbps Ethernet link, which is used to control the SpiNNaker machine from the computer. It also has two spinn-link connectors that enable a connection to external devices such as FPGAs and neuromorphic sensors: retinas or cochleas. This board was connected to an FPGA for real-time input/output communication.

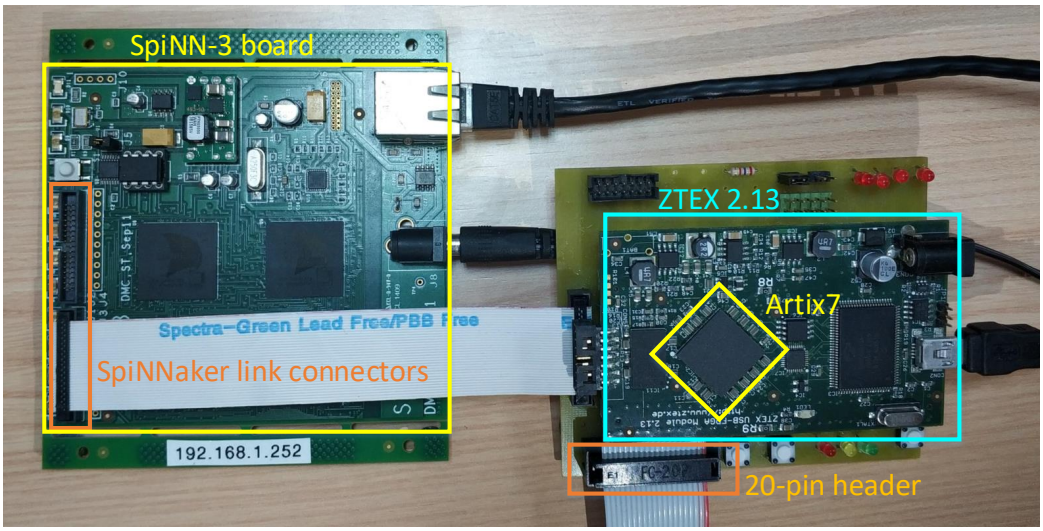

Figure 3: SpiNN-3 machine and ZTEX 2.13 board.

2.1.3. Reconfigurable hardware board

An FPGA-based board was used to implement a digital system design, which controls the neuromorphic robot platform. This approach was considered in other similar works to implement a hardware version of CPGs,

such as [8] and [19], and also in the field of neurorobotics and neuromorphic engineering[22]. This reconfigurable hardware offers flexibility against analog designs and adaptability in real time, in case of system failures.

From the Xilinx Artix-7 family, the XC7A75T chip was used, mounted on the ZTEX 2.13 USB-FPGA board with a 48 MHz clock source. This FPGA chip offers around 75500 logic cells, 100 GPIOs, USB 2.0 interface and DDR3 SDRAM memory. This board serves as a daughter board located on a custom base-board provided with several components: LEDs, user buttons, an AER 20-pin interface and a SpiNNaker link interface. Those interfaces will be used by the SpiNNaker machine to manage the hexapod robot through the FPGA board. An extended explanation is provided in section 2.2.2.

2.2. Methods

2.2.1. CPG

A CPG is a neural network in which interconnected excitatory and inhibitory neurons produce an oscillatory, rhythmic output as a motor pattern, such as walking, flying, running or swimming, with the absence of rhythmic inputs. In this work, we focus on three specific gaits: walk, trot and run, which are selected based on previous working bio-inspired implementations for hexapods [19, 13].

Fig. 4 (bottom) shows the basic structure for each of the CPGs implemented in this work. It consists of eight neurons: two neurons for the Spiking Central Pattern Generator (sCPG) and six output neurons to command the servomotors. The green and red neurons (Fig. 4) make the other neurons, ranging from 0 to 5, fire within different timings generating the selected gait. Each of these six neurons are then connected to two output neurons which will command two different servomotors. This is achievable due to the symmetry of the robot: pairs of servomotors always perform the chosen gait, independently of the selected sCPG.

Three sCPGs following this basic architecture (one per gait) are enclosed within a global SNN model shown in Fig. 4 (top). This global network acts as a mechanism to select which of the sCPGs has to be enabled in order to start generating the gait, inhibiting the other two at the same time. It receives a single spike from the FPGA with a specific neuron address (0, 1 or 2) (the Address Event Representation (AER) protocol is used) indicating the gait pattern that needs to be generated. When this spike reaches the pattern selector population (the three neurons that are closest to the sCPGs), this group of neurons transmit this spike to the correct sCPG, while inhibiting

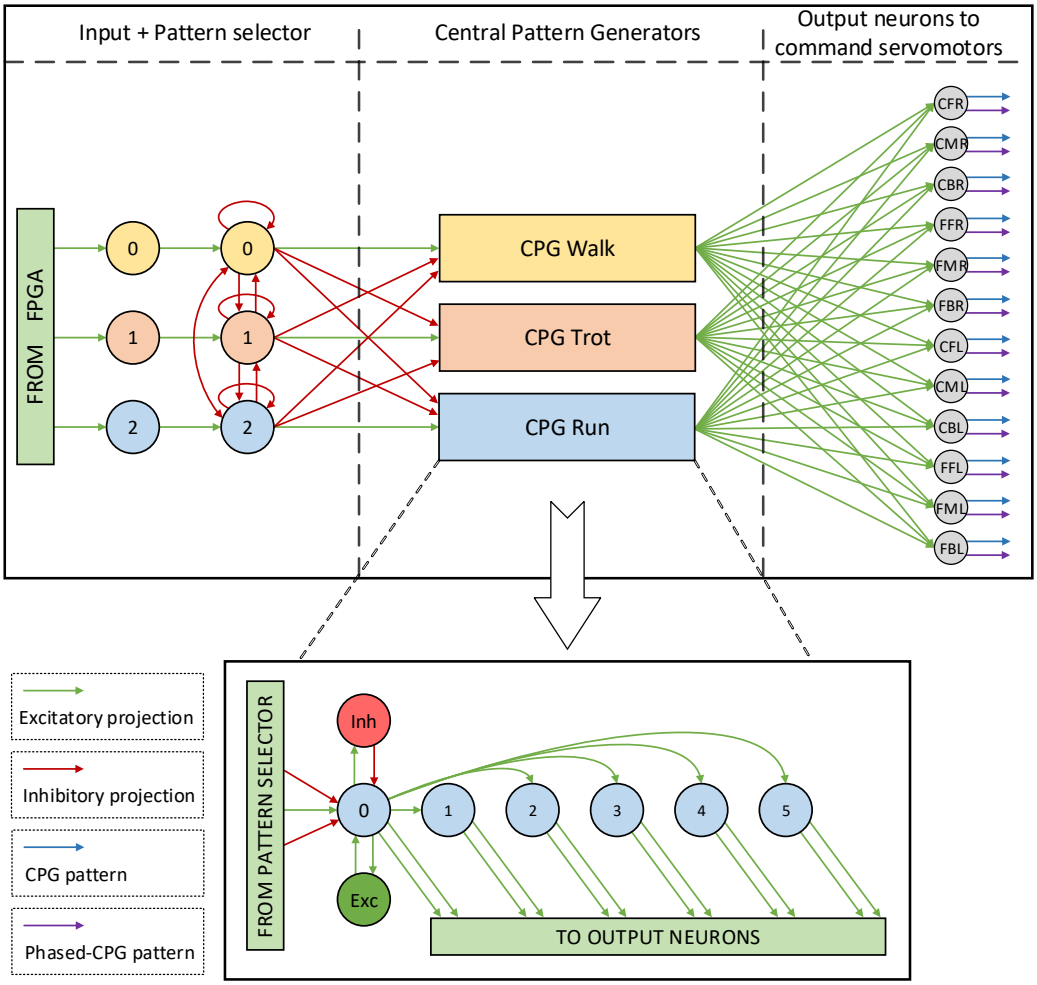

Figure 4: Diagram of the spiking neural network model used (top) with an in-depth view of the CPG architecture (bottom).

the other two. This mechanism allows real-time gait changes by activating the appropriate sCPGs without introducing a long delay (hundreds of ms), which is really important for real-time robotics applications.

A single spike is needed by the selected sCPG to start generating the spiking pattern. The spikes fired by the six sCPG neurons are sent to the last layer of the model, which consists of twelve neurons corresponding to each of the hexapod servomotors. These spikes are transmitted back to the FPGA, where a circuit commands each of the servomotors using the live

output spikes.

As is shown in Fig. 4, each neuron of the output layer, has two outputs from SpiNNaker to the FPGA. The first one is the regular spiking pattern generated by the sCPG to command the servomotors (extension action). The second one is exactly the same pattern, but phased with a delay of 1 tick, needed by the FPGA to command the servomotors back to the standard position (performing the flexion action).

2.2.2. Digital system

As previously mentioned in section 2.1.3, a digital system is needed to implement the neuromorphic robotic controller. This task is often carried out by using an FPGA-based board, running a custom digital system. Designs are generally implemented using Hardware Description Language (HDL), which allows defining any digital circuit model by describing either its behavior or its components' interconnection. Each component of the design is also known as a module, and many functional modules can be encapsulated by a top module, which defines both the input and output signals of the digital circuit.

Fig. 5 shows an overview of the proposed implementation of the Neuro-Pod control system top module. It performs three main functions: to select the gait which the sCPG implemented on SpiNNaker will generate, to send the gait information to the SpiNNaker machine, receive the live output pattern from it and, finally, to generate the Pulse Width Modulation (PWM) signals to control the servomotors. Further details are given next, starting with the pattern selector and then following accordingly with the work flow.

First, the CPG pattern selector module was implemented as a 2-bit up/down unsigned counter. Both up and down signals are declared as input and they are directly mapped to two buttons located on the base board in which the ZTEX board is connected. The current counter value indicates the gait: 0 for walking, 1 for trotting and 2 for running. This information is shown to the user by means of a pair of LEDs in binary format.

Every time the CPG pattern selector changes its value, an interruption is generated through the signal *new_mode* to notify that there is a new data available to the next module. This component is the AER out module, which converts a 2-bit value to a 16-bit AER event, and also handles the AER handshake protocol (*REQUEST* and *ACK* signals). This conversion to AER is carried out since the system needs to send information to the SpiNNaker to generate the pattern.

The SpiNNaker link interface is based on the 2-of-7 protocol. Then, since most of the neuromorphic sensors use the AER protocol, to allow the communication in real-time between AER devices and the SpiNNaker, an AER-SpiNN HDL module was developed by the SpiNNaker team [23].

It takes AER events as input following the AER protocol, and generates packets under the 2-of-7 protocol for the communication from the AER device to the SpiNNaker board. For the reverse communication, it takes 2-of-7 packets and generates AER events. In addition, this module provides four status signals to check in real-time if the communication is working properly.

AER events received by the SpiNNaker, generated by the AER-SpiNN module, are captured by the AER in module, which implements the handshake and sets the value of the event as a 16-bit output signal. In the same way as the AER out module, an interruption is enabled every time the AER in module receives a new input event.

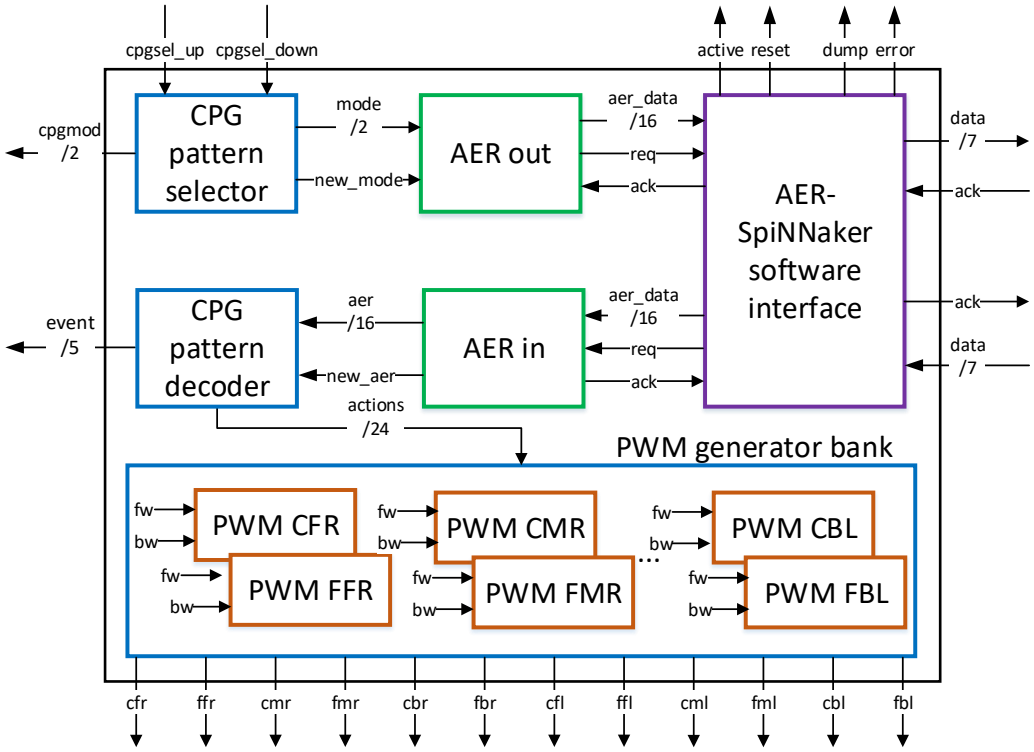

Figure 5: NeuroPod FPGA top module overview.

Those input events correspond to the spikes fired by the output layer

neurons of the SNN implemented on SpiNNaker. Since each output neuron manages the position of one servomotor, these events have to be mapped to the correct one. Therefore, depending on the address of the event, an action is performed over a servomotor.

There are two actions available: either move the servomotor towards the forward position or move the servomotor towards the backward position. A decoding scheme is summarized in Table 1, where each column represents the joints of the NeuroPod following the same nomenclature as in Fig. 2: the FW row means forward action, the BW row means backward action, and each value is the AER event which triggers the action.

The commands received are converted to motion through a PWM generator block, which receives the decoded AER events by means of a 24-bit signal (one enabling signal per action). This PWM generator block was implemented instantiating as many PWM generators as the number of joints there are in the NeuroPod.

Table 1: AER decodification scheme.

	CFR	**FFR**	**CMR**	**FMR**	**CBR**	**FBR**	**CFL**	**FFL**	**CML**	**FML**	**CBL**	**FBL**
FW	0	1	2	3	4	5	6	7	8	9	10	11
BW	12	13	14	15	16	17	18	19	20	21	22	23

The PWM generator implemented includes some features that make the NeuroPod motion control easier: up to three pulse width values can be set in the same VHDL module instead of only one. These values were used to define the positions that the servomotor had to reach. Those positions are: forward, backward (corresponding to the actions) and home. Two control signals were added to the PWM generator to select the configuration of the module: *fw*, which enables the generation of the PWM signal associated to the FW position, and *bw*, which enables the generation of the PWM signal associated to the BW position.

This module has two input signals, which are connected following the scheme shown in Table 1. When a control signal is set to high, either through a single pulse or constant signal, the pulse width value associated to that control signal is loaded in the configuration register. Then, the PWM output signal changes automatically, moving the servomotor to the commanded position. That output signal is held until the module receives a different action command. Finally, the home position is only activated when the global reset signal is released.

3. Results

The results obtained for the simulation of each of the gait patterns on SpiNNaker are shown in Fig. 6. This figure shows the output spikes that the sCPG generates.

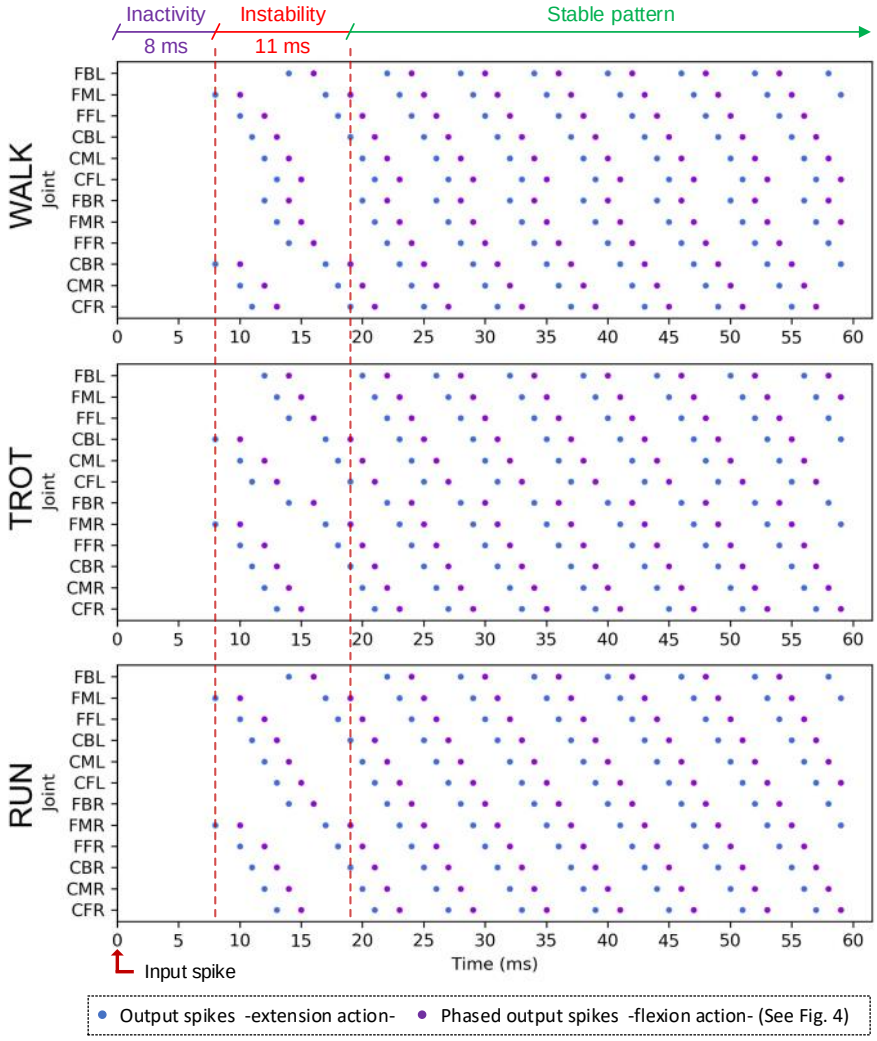

Figure 6: Output spikes for each gait pattern simulated on SpiNNaker.

Then, in Fig. 7, the same plot is shown for a different scenario in which we simulated and tested the behavior of the SNN when forcing the system to change from a specific sCPG to a different one. The figure shows how

the system is able to change from walk to trot and then to run, generating a stable pattern for each gait after a specific period of time, which, in this case, is 23 milliseconds. This delay is the time that the network takes to inhibit the neurons related to the previous gait that was being executed plus the time that the neurons related to the current pattern take to start generating the correct firing output in a stable way. Different delays related to these simulations were measured and are presented in the image.

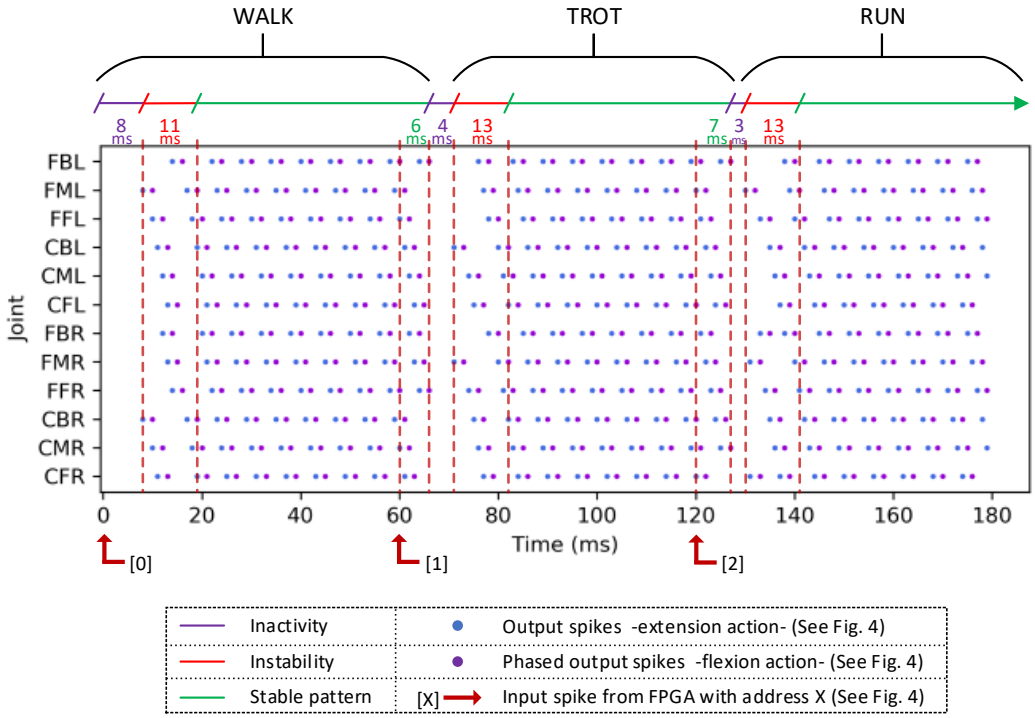

Figure 7: Output spikes from the SpiNNaker simulations that show the gait pattern change behavior.

It is important to mention that these delays were measured in simulation. For a real-time scenario, the delays do not match these values since, due to the fact that the servomotors were not able to work at such speed, we had to set the time_scale_factor parameter on the SpiNNaker board to 100. This made the whole simulation run 100 times slower in real time, which is very convenient for this approach.

Regarding the FPGA, a study of the VHDL module was carried out. A post-implementation resources consumption report was generated, and also

Table 2: FPGA resources consumption and delays

	Resources consumption		**Delays**
	LUTs	Registers	Time (Clock cycles)
CPG pattern selector	2 (<0.01%)	4 (<0.01%)	20.83 ns (1)
AER out	7 (0.01%)	5 (<0.01%)	104.15 ns (5)
AER-SpiNN interface	213 (0.45%)	272 (0.29%)	1374.78 ns (66)
AER in	7 (0.01%)	10 (0.01%)	41.66 ns (2)
CPG pattern decoder	12 (0.03%)	24 (0.03%)	20.83 ns (1)
PWM generator block	720 (1.53%)	576 (0.61%)	1895.53 ns (91)
NeuroPod Top	**986 (2.09%)**	**893 (0.95%)**	**3457.78 ns (166)**

post-implementation simulations were performed to measure the delays of every single VHDL module. The obtained results from those analysis are shown in Table 2. To obtain these times, a clock source of 48MHz was used.

The amount of resources used by the top module is around 2.1 % of the available LUTs and around 1% of the available number of registers. This low resources consumption allows implementing more complex spike-based motor control modules [24] as well as improving the NeuroPod top module by including input information about the environment. In addition, the delay added by the full design, in the worst case, is almost 3.5 µs, which is irrelevant compared to the delays presented in Fig. 7.

After the simulation results were obtained, a real-time analysis of the full system was carried out.

Both the latency from a high level command to the generation of the CPG and the actual motion of the leg were measured using an oscilloscope. These delays can be discarded since they are three orders of magnitude lower than the time taken by the SpiNNaker to generate the sCPG which is 800 ms (the simulated time updated with the time_scale_factor).

According to that latency and the time that a gait cycle takes to be per-

formed, we can conclude that the theoretical maximum value of the movement speed is 1.66 cm/s.

After that, we compared the simulation time with the real-time delays. To this end, four cases of study were defined: resting to moving, stabilization time, movement's period, and change time between two different gaits.

There were time differences since the parameter time_scale_factor was set to 100 in the SpiNNaker script in order to slowdown the SpiNNaker output.

4. Discussion and conclusions

A real time sCPG using neuromorphic hardware is presented. As was stated in the introduction, several legs, up to six, are controlled. The delays introduced in the open-loop control are very low, in the order of ms (see Table 2). Previous works have a 50ms delay of propagation [11], or a converge time of 5 seconds in [25]. Our results show a time of 23 ms (worst case) to converge and approx. 20ms to propagate the gait.
Another difference with [11] is that we propose to use SpiNNaker [26] instead of the neuromorphic chip ROLLS [12]. Also, six neurons less than [11] are used in this work.

As a future work, we propose to use some voice commands and a neuromorphic auditory sensor [27] to change the gait. This real-time change will be available based on the classification achieved by an SNN connected to the sCPG.
Furthermore, feedback will be introduced in the sCPG by including either touch or elevation sensors in the lower part of the leg of the robot. This feedback will be used to learn in order to provide learning capabilities to the system.

In this work, we presented and achieved what we believe is the first implementation of a real-time neuromorphic spiking CPG to command a hexapod robot using SpiNNaker. Furthermore, we included the possibility to change between three different gaits online.

Demonstration video is also available[2].

[2] `https://youtu.be/YZYAPDJHvLI`

Acknowledgements

This work was supported by the Spanish grant (with support from the European Regional Development Fund) COFNET (TEC2016-77785-P). The work of D. Gutierrez-Galan was supported by a "Formación de Personal Investigador" Scholarship from the Spanish Ministry of Education, Culture and Sport. This work was carried out during a research internship of D. Gutierrez-Galan and Juan P. Dominguez-Morales in the Department of Computer Architecture and Technology (Universidad de Cádiz, Spain).

References

References

[1] S. Grillner, P. Wallén, K. Saitoh, A. Kozlov, B. Robertson, Neural bases of goal-directed locomotion in vertebratesan overview, Brain research reviews 57 (2008) 2–12.

[2] J. Duysens, H. W. Van de Crommert, Neural control of locomotion; Part 1: The central pattern generator from cats to humans, Gait & Posture 7 (1998) 131–141.

[3] P. A. Guertin, The mammalian central pattern generator for locomotion, Brain Research Reviews 62 (2009) 45–56.

[4] K. Minassian, U. S. Hofstoetter, F. Dzeladini, P. A. Guertin, A. Ijspeert, The human central pattern generator for locomotion: Does it exist and contribute to walking?, The Neuroscientist 23 (2017) 649–663.

[5] M. Schilling, T. Hoinville, J. Schmitz, H. Cruse, Walknet, a bio-inspired controller for hexapod walking, Biological Cybernetics 107 (2013) 397–419.

[6] R. J. Vogelstein, F. V. Tenore, L. Guevremont, R. Etienne-Cummings, V. K. Mushahwar, A silicon central pattern generator controls locomotion in vivo, IEEE transactions on biomedical circuits and systems 2 (2008) 212–222.

[7] G. Sartoretti, S. Shaw, K. Lam, N. Fan, M. Travers, H. Choset, Central pattern generator with inertial feedback for stable locomotion and climbing in unstructured terrain, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 1–5.

[8] J. H. Barron-Zambrano, C. Torres-Huitzil, B. Girau, Hardware implementation of a CPG-based locomotion control for quadruped robots, in: International Conference on Artificial Neural Networks, Springer, pp. 276–285.

[9] A. Crespi, D. Lachat, A. Pasquier, A. J. Ijspeert, Controlling swimming and crawling in a fish robot using a central pattern generator, Autonomous Robots 25 (2008) 3–13.

[10] S. Still, B. Schölkopf, K. Hepp, R. J. Douglas, Four-legged walking gait control using a neuromorphic chip interfaced to a support vector learning algorithm, in: Advances in neural information processing systems, pp. 741–747.

[11] E. Donati, F. Corradi, C. Stefanini, G. Indiveri, A spiking implementation of the lamprey's Central Pattern Generator in neuromorphic VLSI, in: IEEE 2014 Biomedical Circuits and Systems Conference, BioCAS 2014 - Proceedings, pp. 512–515.

[12] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, G. Indiveri, A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses, Frontiers in neuroscience 9 (2015) 141.

[13] B. Cuevas-Arteaga, J. P. Dominguez-Morales, H. Rostro-Gonzalez, A. Espinal, A. F. Jimenez-Fernandez, F. Gomez-Rodriguez, A. Linares-Barranco, A SpiNNaker application: design, implementation and validation of SCPGs, in: International Work-Conference on Artificial Neural Networks, Springer, pp. 548–559.

[14] S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, The SpiNNaker Project, Proceedings of the IEEE (2014).

[15] J. K. Douglass, N. J. Strausfeld, Visual motion detection circuits in flies: peripheral motion computation by identified small-field retinotopic neurons, Journal of Neuroscience 15 (1995) 5596–5611.

[16] M. B. Milde, O. J. Bertrand, R. Benosman, M. Egelhaaf, E. Chicca, Bioinspired event-driven collision avoidance algorithm based on optic flow, in: Event-based Control, Communication, and Signal Processing (EBCCSP), 2015 International Conference on, IEEE, pp. 1–7.

[17] A. Schneider, J. Paskarbeit, M. Schaeffersmann, J. Schmitz, Hector, a new hexapod robot platform with increased mobility-control approach, design and communication, in: Advances in Autonomous Mini Robots, Springer, 2012, pp. 249–264.

[18] A. Büschges, T. Akay, J. P. Gabriel, J. Schmidt, Organizing network action for locomotion: insights from studying insect walking, Brain research reviews 57 (2008) 162–171.

[19] 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, A CPG system based on spiking neurons for hexapod robot locomotion, Neurocomputing 170 (2015) 47–54.

[20] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, A. D. Brown, Overview of the SpiNNaker system architecture, IEEE Transactions on Computers 62 (2013) 2454–2467.

[21] L. A. Plana, S. B. Furber, S. Temple, M. Khan, Y. Shi, J. Wu, S. Yang, A GALS infrastructure for a massively parallel multiprocessor, IEEE Design & Test of Computers 24 (2007).

[22] 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, On multiple AER handshaking channels over high-speed bit-serial bidirectional LVDS links with flow-control and clock-correction on commercial FPGAs for scalable neuromorphic systems, IEEE transactions on biomedical circuits and systems 11 (2017) 1133–1147.

[23] L. Plana, J. Heathcote, J. Pepper, S. Davidson, J. Garside, S. Temple, S. Furber, spI/O: A library of FPGA designs and reusable modules for I/O in SpiNNaker systems (2014).

[24] A. Jimenez-Fernandez, G. Jimenez-Moreno, A. Linares-Barranco, M. J. Dominguez-Morales, R. Paz-Vicente, A. Civit-Balcells, A neuro-inspired spike-based PID motor controller for multi-motor robots with low cost FPGAs, Sensors 12 (2012) 3831–3856.

[25] A. Crespi, A. J. Ijspeert, et al., AmphiBot II: An amphibious snake robot that crawls and swims using a central pattern generator, in: Proceedings of the 9th international conference on climbing and walking robots (CLAWAR 2006), volume 11, pp. 19–27.

[26] S. B. Furber, F. Galluppi, S. Temple, L. A. Plana, The SpiNNaker project, Proceedings of the IEEE 102 (2014) 652–665.

[27] A. Jiménez-Fernández, E. Cerezuela-Escudero, L. Miró-Amarante, M. J. Domínguez-Morales, F. de Asís Gómez-Rodríguez, A. Linares-Barranco, G. Jiménez-Moreno, A binaural neuromorphic auditory sensor for FPGA: a spike signal processing approach, IEEE transactions on neural networks and learning systems 28 (2017) 804–818.