A SpiNNaker application: design, implementation and validation of SCPGs

Brayan Cuevas-Arteaga[1], Juan P. Dominguez-Morales[2], Horacio Rostro-Gonzalez[1], Andres Espinal[3], Angel F. Jimenez-Fernandez[2], Francisco Gomez-Rodriguez[2], and Alejandro Linares-Barranco[2]

[1] Departament of Electronics, DICIS-University of Guanajuato
Carr. Salamanca-Valle de Santiago Km 3.5+1.8 Palo Blanco
36885 Salamanca, Gto.,Mexico
brayan.cuevas@ugto.mx

[2] Robotic and Technology of Computers Lab., University of Seville
Av. Reina Mercedes s/n,
41012 Sevilla, Spain
jpdominguez@atc.us.es

[3] Departament of Organizational Studies, DCEA-University of Guanajuato
Fraccionamiento 1; Col. El Establo S/N
36250 Guanajuato Gto., Mexico

Abstract. In this paper, we present the numerical results of the implementation of a Spiking Central Pattern Generator (SCPG) on a SpiNNaker board. The SCPG is a network of current-based leaky integrate-and-fire (LIF) neurons, which generates periodic spike trains that correspond to different locomotion gaits (i.e. walk, trot, run). To generate such patterns, the SCPG has been configured with different topologies, and its parameters have been experimentally estimated. To validate our designs, we have implemented them on the SpiNNaker board using PyNN and we have embedded it on a hexapod robot. The system includes a Dynamic Vision Sensor system able to command a pattern to the robot depending on the frequency of the events fired. The more activity the DVS produces, the faster that the pattern that is commanded will be.

Keywords: SCPGs, legged robots locomotion, SpiNNaker, Spiking neurons, hardware based implementations

1 Introduction

Robotic locomotion is a highly active research field in artificial intelligence. Nowadays, several methods have been proposed to achieve locomotion in a variety of robots (i.e. wheeled robots, legged robots, swimming robots, flying robots, etc.); particularly for non-wheeled robots, bioinspired locomotion systems may be implemented [1]. These systems, commonly known as Central Pattern Generators (CPGs) imitate behaviors of biological neural mechanisms and they are capable of endogenously produce periodically rhythmic patterns to contribute in locomotion of living beings among other rhythmic activities such as digestion, swallowing, etc. [2]. However, they do not work isolatedly, since they interact with other parts of the central nervous system [3] and afferent sensory information may shape the CPG's outputs [4].

CPG-based locomotion systems have several advantages over non-bioinspired ones, such as: *rhytmicity*, *stability*, *adaptability* and *variety* (see [5] for a detailed explanation of these features). These systems have been designed and implemented for biped [6, 7], quadruped [8] and hexapod, [9, 10] among other kinds of non-wheeled robots (see [2, 1, 5] for comprehensive reviews). The implementation of CPGs for either software or hardware applications involves previous phases of modeling, analysis and modulation, which comprises the kind of neuron model to use, the coupling and the structure of the connections in a network. The last deals with parameter tunning and gait transitions [5].

As mentioned before, an important aspect on the implementation of CPGs is the selection of a neuron model. In this regard, there are several neuron models with different degrees of plausibility. To date, spiking neurons are considered as the most plausible neuron models, they form Spiking Neural Networks (SNNs) which are considered the third generation of Artificial Neural Networks [11]; CPGs built as SNNs are known as Spiking Central Pattern Generators (SCPGs).

In this paper a fully CPG-based locomotion system for the locomotion of a hexapod robot is proposed. Our proposal covers all phases suggested in [5]: modeling and analysis, modulation and implementation. The CPG-based locomotion system is conceptually based on works published in [9, 8, 10]; however we have changed the spiking neuron model and implementation platform by a more plausible one and a brain-like hardware platform respectively, i.e., the current-based integrate-and-fire model and a SpiNNaker board. Three different gaits are generated by the locomotion system, and simulated and implemented on the SpiNNaker board and validated on a real hexapod robot. Moreover, our proposal incorporates gait transitions according to the activity sensed by a DVS camera.

The rest of the paper is organized as follows: Section 2.1 provides information concerning the theoretical background around Spiking Central Pattern Generators. Section 2.2 provides information about the hardware used in this work, specifically on the SpiNNaker board. In Section 3, the design, implementation and validation of the system is explained in detail. Section 4 presents numerical results of the simulation. In section 5, we present a perspective of the work and we conclude in Section 6.

2 Materials & Methods

2.1 Spiking Central Pattern Generators

SCPGs are a variation of the well-known and widely studied CPGs, which are specialized neural networks capable of endogenously produce rhythmic patterns. The CPGs contribute to living beings to perform actions without consent-effort such as walking among others [2, 6]. Also, they have served as the basis of locomotion systems for non-wheeled robots with remarkable advantages over non-bioinspired locomotion systems [5].

SCPGs naturally handle spatiotemporal information as locomotion requires [10], which means that they receive and send information over time. In [9, 8, 10] are developed, implemented and tested SCPGs based on a biological study of insect locomotion [12] for legged robots by means of BMS neuron models [13]. The locomotion of legged robots is achieved in these works by means of spike-time-activity where each servo-motor receives a spike train; the presence or absence of spikes indicates the state of a servo-motor at the current time.

The SCPGs use the current-based leaky integrate-and-fire neuron model [14] as processing unit. Such model is one of the standard models in PyNN [15]. Eq. (1) shows the equation of the model with fixed threshold and decaying-exponential post-synaptic current, excitatory injection in Eq. (2) and inhibitory injection given by Eq. (3).

$$\frac{dv}{dt} = \frac{ie + ii + i\_offset + i\_inj}{c\_m} + \frac{v\_rest - v}{tau\_m} \tag{1}$$

$$\frac{die}{dt} = -\frac{ie}{tau\_syn\_E} \tag{2}$$

$$\frac{dii}{dt} = -\frac{ii}{tau\_syn\_I} \tag{3}$$

where v stands for the current of membrane potential. The excitatory and inhibitory current injections are expressed by ie and ii respectively. *i_offset* represents a base input current added each timestep. *i_inj* is an external current injection but in this case it is equal to zero. *c_m* is the capacitance of the leaky integrate-and-fire neuron in nano-Farads. *v_reset* is the voltage to set the neuron at immediately after a spike. *tau_m* means the time-constant of the RC circuit, in milliseconds. *tau_syn_E* and *tau_syn_I* are the excitatory and inhibitory input current decay time constant respectively. The neuron model uses a *tau_refrac* value for representing the refractory period, in milliseconds and finally, *v_thresh* stands for the threshold voltage at which the neuron will spike.

In PyNN the model is described as **if_curr_exp** and the code reads:

$eqs = brian.Equations('''$
$dv/dt = (ie + ii + i_offset + i_inj)/c_m + (v_rest - v)/tau_m : mV$
$die/dt = -ie/tau_syn_E : nA$
$dii/dt = -ii/tau_syn_I : nA$
$tau_syn_E : ms$
$tau_syn_I : ms$
$tau_m : ms$
$c_m : nF$
$v_rest : mV$
$i_offset : nA$

$i_i nj : nA$
'''
)

2.2 Spiking Neural Network Architecture (SpiNNaker)

To implement and validate our SCPGs we used a SpiNNaker board [16]. SpiNNaker is a massively-parallel multicore computing system designed for modeling very large spiking neural networks in real time. Both the system architecture and the design of the SpiNNaker chip have been developed by the Advanced Processor Technologies Research Group (APT) [17], which is based on the School of Computer Science at the University of Manchester. Each SpiNNaker chip consists of 18,200 MHz general-purpose ARM968 cores. The communication between them is done via packets carried by a custom interconnect fabric. The transmission of these packets is brokered entirely by hardware, giving the overall engine and extremely high bisection bandwidth.

In this work, a SpiNNaker 102 machine is used. This board comprises 4 SpiNNaker chips and, hence, it has 72 ARM processor cores deployed as 4 monitor processors, 64 application cores and 4 spare cores. A 100 Mbps Ethernet connection is used as control and I/O interface between the computer and the SpiNNaker board. 5V-1A supply is required for this machine. This platform has been used in previous works by the authors [18] [19], proving its robustness and versatility.

Fig. 1. SpiNNaker 102 machine.

2.3 Dynamic Vision Sensor

The AER DVS128 retina chip (silicon retina) [20] consists of an array of autonomous pixels that respond to relative changes in light intensity in real-time by placing the address of that specific pixel in an arbitrary asynchronous bus. Only pixels that are stimulated by any change of lightning transmit their addresses (events are produced). Hence, scenarios with no motion do not generate output events. These addresses are called Address Event (AE) and contains the x and y coordinates of the pixel that produced the event.

In this work, an AER DVS128 sensor was used to switch among the three different gaits, which has an array of 128x128 pixels. 7 bits are needed to encode each dimension of the array of pixels in this case. This Dynamic Vision Sensor also generates a polarity bit that represents the contrast change, where positive means a light increment and, negative, a light decrement. The DVS128 sensor is placed on the PAER interface that allows parallel AER through the CAVIAR connector [21].

3 Design, implementation and validation

3.1 Design

Three Spiking Central Pattern Generators (SCPGs) have been implemented into SpiNNaker. Each of them represents a different locomotion gait of the hexapod: walk, trot and run. These SCPGs were deployed in the machine using the PyNN library, which allows to easily create populations of neurons, connect them and assign weights to the connections.

SCPGs need an initial potential stimulus to start running. In this case, the resting potential and the threshold of the neurons are set to -65 mV and -50 mV, respectively. Hence, an initial potential of -49 mV is set on the first population to make the SCPG start running at the beginning of the simulation. With this stimulus, the SCPG is able to run infinitely while providing the same output spike pattern.

The current-based Leaky Integrate and Fire neuron model defined in Eq (1) has been used in this work, and its configuration parameters are presented in Table 1.

The SNN used to design each of the three locomotion sequences has basically the same architecture. The difference between them are the number of populations used, weights and delays for each of the sequences. It is necessary to mention that each of these sequences is implemented as a different SCPG.

In Fig. 2, we show the spiking neural network topology. Here, red neurons represent the direct stimulus towards the servomotors of the hexapod robot, and the neurons in blue and yellow are used to balance the generation of the correct patterns in each case. As can be seen in the figure, blue connections represent excitatory activity and, on the other hand, yellow connections represent inhibitory activity.

Table 1. Configuration parameters of the current-based LIF neuron model

Parameter	Value
cm	0.25
tau_m	20.0
tau_refrac	2.0
v_reset	-68.0
v_rest	-65.0
v_thresh	-50.0
tau_syn_E	5.0
tau_syn_I	5.0
i_offset	0.0

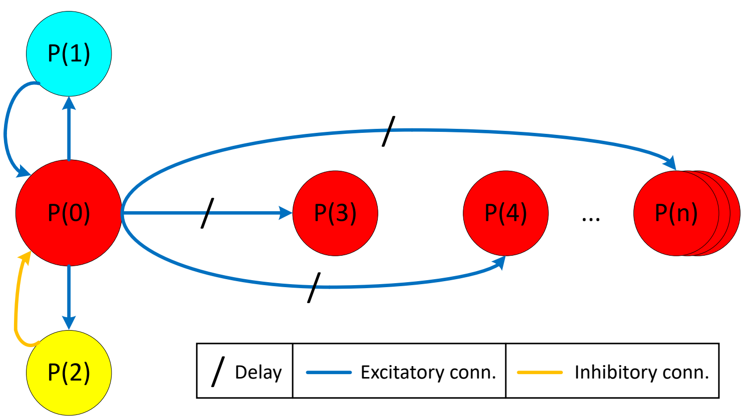

Fig. 2. SCPGs neural network architecture.

To create the complete design of a SCPG it was necessary to find the weight and delay values for each of the connections between populations of neurons. This was carried out using a exhaustive search method.

At the end of the design of the SCPGs, a matrix is obtained, where each element of a column represents a servomotor and each row is an instant of time. Then, the absence or existence of a spike in this matrix will be translated to the movement of a specific servomotor on the hexapod robot. Fig. 3 shows the spike trains (SCPG's output) representing the three different locomotion patterns that will perform the hexapod robot.

3.2 Implementation

The implementation of the SCPG in SpiNNaker was done using the PyNN toolchain. The results are the three types of spike patterns required for the locomotion. These sequences are shown in Fig. 3. In each figure, X-axis represents the simulation time and Y-axis shows the twelve neurons required for the locomotion of a hexapod robot; one for each servo-motor. The neural activity of each neuron is transduced into electrical signals for moving its associated servo-motor. Specifically, locomotion corresponds to horizontal (coxa) and vertical (femur) movements, thus six neurons controls the horizontal movement and the other six the vertical one.

The results were corroborated with the state of the art [9]. After verifying that the results were correct in simulation, we proceeded to implement a complete system to validate the implementation of the SCPGs, which is described in the following subsection.

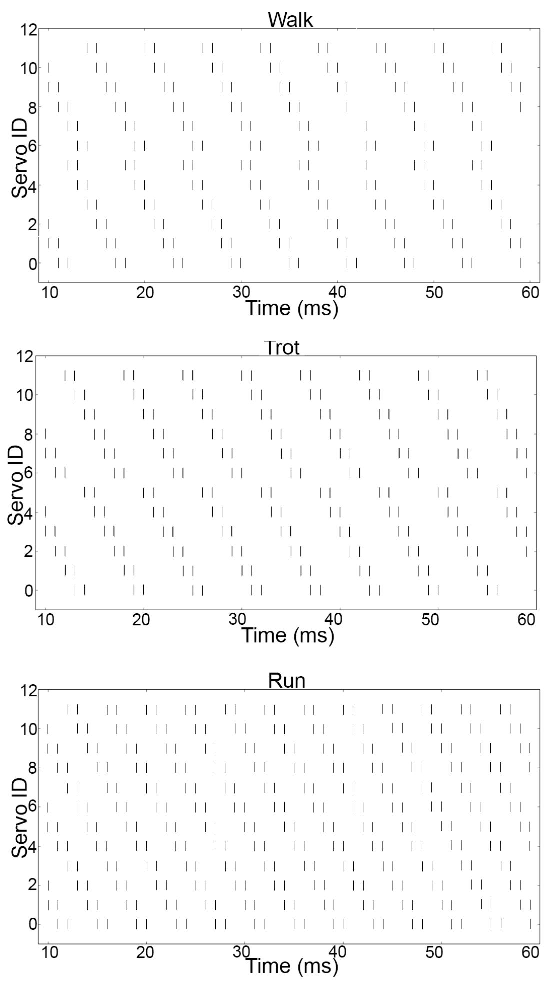

Fig. 3. Spike output patterns for each locomotion gait of the hexapod.

3.3 Validation

In Fig. 4, a block diagram of our complete validation system can be observed. It consists of a DVS camera, which receives events according to a change of intensity either positive or negative. The information is sent as an Address-Event-Representation (AER) format to an FPGA, which is used to synchronize and encode the incoming information in the specific format used by SpiNNaker (40 bits). The SpiNNaker is connected via Ethernet to a PC. For the mechanical validation of the SCPGs for this last, a hexapod robot was used, which consists of 12 servomotors: 6 to give movement to the extremities horizontally and 6 vertically. Each of the servomotors was connected to a PWM provided by an Arduino Mega board.

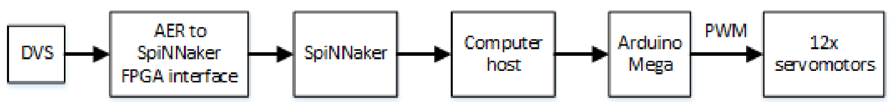

Fig. 4. Block diagram of the complete validation system.

The experiment consisted in connecting the complete system and, according to the frequency of events produced by the DVS, a specific locomotion sequence is executed. Different thresholds of event frequencies were set for this purpose. For example, if the DVS produces a few events, the robot would not perform any gait. However, if the DVS produces a moderate event frequency the hexapod would start walking, and so with trotting and running with higher event frequencies. All locomotion sequences were included in the same PyNN script. This scenario corroborates the operation of the complete system including both software and hardware.

4 Numerical results

In Fig. 5, we show the whole system running a locomotion pattern in real time. In such figure, we can easily identify the main elements of the system: a SpiNNaker board configured with a SCPG performing the trot pattern, which can be observed on the oscilloscope. Due to the fact that our oscilloscope can only register four analog signals simultaneously, we decided to register three signals at the same time, i.e. three for each of the movements: horizontal (coxa) and vertical (femur), making thus a total of 12 signals registered for each gait.

To show the effectiveness on replicating locomotion patterns such as those found in vertebrates we performed numerical simulations for the three different (walk, trot and run) gaits using the SpiNNaker board and they are presented in Fig. 6. In such figure, the gaits are presented as follows: in the left side we can find the signals corresponding to the horizontal movements and in the right side for the vertical movements, both for the right legs performing the three gaits. The hexapod robot has a symmetric design. For this reason we only present the results for one of the hexapod sides (the legs on the right).

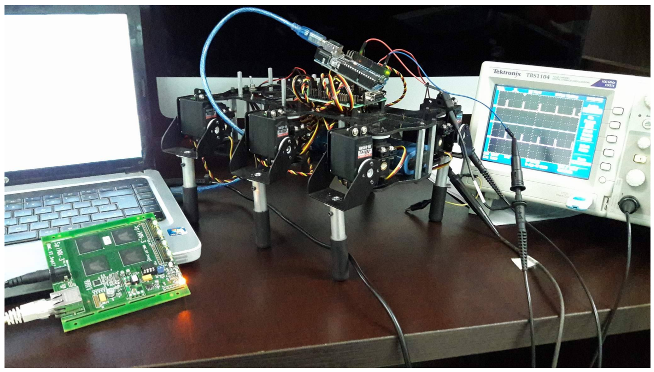

Fig. 5. System configuration

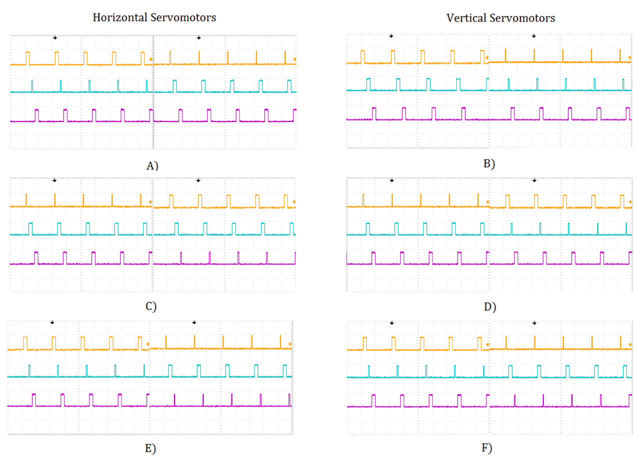

Fig. 6. Real time simulation

5 Stand-alone cognitive locomotion system

The work presented in this paper is part of a more ambitious project in which the non-wheeled locomotion robot is commanded by a SNN in real-time using neuromorphic sensors and cognition to react to different stimulus. For example, to make the robot to follow an object that produces higher event activity on the

DVS sensor it can easily be discriminated not only the frequency of the events, but also by the position on the visual field (left, center or right) to command a pattern that makes the robot turn left or right [22].

The hexapod robot used in the previous section will host an FPGA-based platform that will command the actuators of the robot according to SpiNNaker output stream of events. The circuit on the FPGA will be collecting AER events from the DVS, converting them to 40-bit format in one part. Furthermore, it will host a second circuit that receives 40-bit output from SpiNNaker and it will convert them into the according duty-cycles for the 12 Servo-motors of the hexapod to reproduce the right pattern. By the use of both hardware reconfigurable systems (FPGA and SpiNNaker) in the same robotic platform in a stand-alone capability, any research work could be implemented by properly dividing the functionality between both of them. DVS postprocessing [23] can be done in the FPGA in order to make lighter the algorithms running in SpiNNaker.

6 Conclusions

In this work, the authors have presented the design, implementation and validation of Spiking Central Pattern Generators for three locomotion gaits (walk, trot and run). The design was carried out in PyNN by using the current-based Leaky Integrate-and-Fire neuron model. Both implementation and validation were performed on a SpiNNaker board, which controls the locomotion of a real hexapod robot (See Fig. 5) through the generation of periodic spike trains (gaits) sent to the servomotors of the robot. Also, a DVS sensor can be incorporated to provide of sensory feedback and motor response through the locomotion of the hexapod robot. The results obtained are satisfactory comparing it with previous works, due to the fact that the implementation was done in a massively-parallel multicore computing system, improving power consumption and temporal efficiency.

This opens the way to consider future work such as including the complete system within an hexapod, quadruped or bipedal robot, where the robot is completely stand-alone based on the perceptions obtained through neuromorphic sensors (e.g. DVS and Neuromorphic Cochlea) and processing this information in real-time.

Acknowledgements. This work is partially supported by the Spanish government grant (with support from the European Regional Development Fund) COFNET (TEC2016-77785-P). Also, this work has been supported by the Mexican government through the CONACYT project Aplicacion de la Neurociencia Computacional en el Desarrollo de Sistemas Roboticos Biologicamente Inspirados (269798). The work of Juan P. Dominguez-Morales was supported by a Formación de Personal Universitario Scholarship from the Spanish Ministry of Education, Culture and Sport.

References

1. Wu, Q., Liu, C., Zhang, J., Chen, Q.: Survey of locomotion control of legged robots inspired by biological concept. Science in China Series F: Information Sciences **52**(10) (2009) 1715–1729
2. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. Neural networks **21**(4) (2008) 642–653
3. Arena, P.: The central pattern generator: A paradigm for artificial locomotion. Soft Computing **4**(4) (2000) 251–266
4. MacKay-Lyons, M.: Central pattern generation of locomotion: a review of the evidence. Physical Therapy **82**(1) (2002) 69–83
5. Yu, J., Tan, M., Chen, J., Zhang, J.: A survey on cpg-inspired control models and system implementation. IEEE Transactions on neural networks and learning systems **25**(3) (2014) 441–456
6. Lewis, M.A., Tenore, F., Etienne-Cummings, R.: Cpg design using inhibitory networks. In: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on, IEEE (2005) 3682–3687
7. Russell, A., Orchard, G., Etienne-Cummings, R.: Configuring of spiking central pattern generator networks for bipedal walking using genetic algorthms. In: Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on, IEEE (2007) 1525–1528
8. Espinal, A., Rostro-Gonzalez, H., Carpio, M., Guerra-Hernandez, E., Ornelas-Rodriguez, M., Puga-Soberanes, H., Sotelo-Figueroa, M., Melin, P.: Quadrupedal robot locomotion: A biologically inspired approach and its hardware implementation. Computational Intelligence and Neuroscience **2016** (2016)
9. Rostro-Gonzalez, H., Cerna-Garcia, P.A., Trejo-Caballero, G., Garcia-Capulin, C.H., Ibarra-Manzano, M.A., Avina-Cervantes, J.G., Torres-Huitzil, C.: A cpg system based on spiking neurons for hexapod robot locomotion. Neurocomputing **170** (2015) 47–54
10. Espinal, A., Rostro-Gonzalez, H., Carpio, M., Guerra-Hernandez, E.I., Ornelas-Rodriguez, M., Sotelo-Figueroa, M.: Design of spiking central pattern generators for multiple locomotion gaits in hexapod robots by christiansen grammar evolution. Frontiers in Neurorobotics **10** (2016)
11. Maass, W.: Networks of spiking neurons: the third generation of neural network models. Neural networks **10**(9) (1997) 16591671
12. Grabowska, M., Godlewska, E., Schmidt, J., , Daun-Gruhn, S.: Quadrupedal gaits in hexapod animals inter-leg coordination in free-walking adult stick insects. Journal of Experimental Biology **215**(24) (42554266) 60–79
13. Soula, H., Beslon, G., , Mazet, O.: Spontaneous dynamics of asym- metric random recurrent spiking neural networks. Neural Computation **18**(1) (2006) 60–79
14. Abbott, L.F.: Lapicques introduction of the integrate-and-fire model neuron (1907). Brain research bulletin **50**(5) (1999) 303–304
15. Davison, A., Brderle, D., Eppler, J., Kremkow, J., Muller, E., Pecevski, D., Perrinet, L., Yger, P.: Pynn: a common interface for neuronal network simulators. Frontiers in Neuroinformatics **2** (2009) 11
16. Painkras, E., Plana, L.A., Garside, J., Temple, S., Galluppi, F., Patterson, C., Lester, D.R., Brown, A.D., Furber, S.B.: Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation. IEEE Journal of Solid-State Circuits **48**(8) (2013) 1943–1953

17. Advanced Processor Technologies Research Group: Spinnaker home page. http://apt.cs.manchester.ac.uk/projects/SpiNNaker. Accessed: 2016-01-22.
18. Dominguez-Morales, J.P., Jimenez-Fernandez, A., Rios-Navarro, A., Cerezuela-Escudero, E., Gutierrez-Galan, D., Dominguez-Morales, M.J., Jimenez-Moreno, G.: Multilayer spiking neural network for audio samples classification using spinnaker. In: International Conference on Artificial Neural Networks, Springer (2016) 45–53
19. Rios-Navarro, A., Dominguez-Morales, J.P., Tapiador-Morales, R., Dominguez-Morales, M., Jimenez-Fernandez, A., Linares-Barranco, A.: A sensor fusion horse gait classification by a spiking neural network on spinnaker. In: International Conference on Artificial Neural Networks, Springer (2016) 36–44
20. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128x128 120 db 15 us latency asynchronous temporal contrast vision sensor. IEEE journal of solid-state circuits **43**(2) (2008) 566–576
21. Serrano-Gotarredona, R., Oster, M., Lichtsteiner, P., Linares-Barranco, A., Paz-Vicente, R., Gómez-Rodríguez, F., Camuñas-Mesa, L., Berner, R., Rivas-Pérez, M., Delbruck, T., et al.: Caviar: A 45k neuron, 5m synapse, 12g connects/s aer hardware sensory–processing–learning–actuating system for high-speed visual object recognition and tracking. IEEE Transactions on Neural Networks **20**(9) (2009) 1417–1438
22. Jiménez-Fernandez, A., Fuentes-del Bosh, J.L., Paz-Vicente, R., Linares-Barranco, A., Jiménez, G.: Neuro-inspired system for real-time vision sensor tilt correction. In: Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, IEEE (2010) 1394–1397
23. Linares-Barranco, A., Gómez-Rodríguez, F., Villanueva, V., Longinotti, L., Delbrück, T.: A usb3. 0 fpga event-based filtering and tracking framework for dynamic vision sensors. In: Circuits and Systems (ISCAS), 2015 IEEE International Symposium on, IEEE (2015) 2417–2420