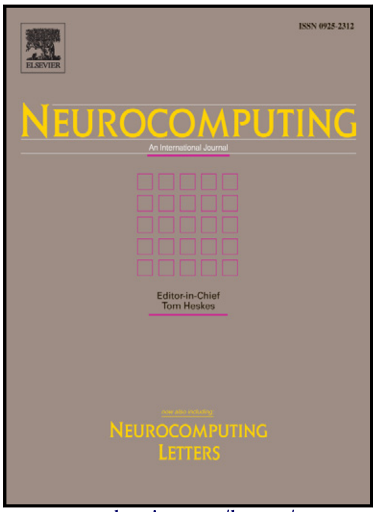
Author's Accepted Manuscript



www.elsevier.com/locate/neucom

A CPG system based on spiking neurons for hexapod robot locomotion

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

PII: S0925-2312(15)00878-4
DOI: http://dx.doi.org/10.1016/j.neucom.2015.03.090
Reference: NEUCOM15699

To appear in: *Neurocomputing*

Received date: 31 October 2014
Revised date: 28 January 2015
Accepted date: 9 March 2015

Cite this article as: 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,* http://dx.doi.org/10.1016/j.neucom.2015.03.090

A CPG system based on spiking neurons for hexapod robot locomotion

H. Rostro-Gonzalez[a], P. A. Cerna-Garcia[a], G. Trejo-Caballero[a,b], C. H. Garcia-Capulin[b], M. A. Ibarra-Manzano[a], J. G. Avina-Cervantes[a], C. Torres-Huitzil[c]

[a] *Department of Electronics, DICIS-University of Guanajuato, Mexico*
[b] *Department of Mechatronics, ITESI, Mexico*
[c] *Information Technology Laboratory, CINVESTAV, Mexico*

Abstract

In this paper, we propose a locomotion system based on a Central Pattern Generator (CPG) for a hexapod robot, suitable for embedded hardware implementation. The CPG system was built as a network of spiking neurons, which produce rhythmic signals for three different gaits (walk, jogging and run) in the hexapod robot. The spiking neuron model used in this work is a simplified form of the well-known generalised Integrate-and-Fire neuron model, which can be trained using the Simplex method. The use of spiking neurons makes the system highly suitable for digital hardware implementations that exploit the inherent parallelism to replicate the intrinsic, computationally efficient, distributed control mechanism of CPGs. The system has been implemented on a Spartan 6 FPGA board and fully validated on a hexapod robot. Experimental results show the effectiveness of the proposed approach, based on existing models and techniques, for hexapod rhythmic locomotion.

Keywords: Central Pattern Generators, Spiking Neuron Models, Legged Robot Locomotion, FPGA, Neuromorphic Engineering

1. Introduction

Legged robot locomotion mechanisms are often inspired by biological systems, which are very successful in moving through a wide area of abrupt and harsh environments. In most cases, these locomotion mechanisms are generated by neural circuits commonly called central pattern generators (CPG's) that when activated can produce rhythmic motor patterns such as walking, breathing, flying, and swimming in the absence of sensory or descending inputs that carry specific timing information Marder (2001). The principles of CPG systems emerged one century ago Brown (1914). In such work, Brown proposes the use of neurons which inhibit each other to achieve the control of the bending and tension in the leg muscles in order to perform the action of walking. From a mathematical viewpoint, CPG's have been modelled at different levels of abstraction, from simplified neuron models such as integrate-and-fire neurons to detailed biophysical models such as the Hodgkin-Huxley type Ijspeert (2008). These neuron models are best known as spiking neuron models and they are one of the main research areas in Computational Neuroscience Dayan (2005). There are other CPG's implementations based on oscillators, however in this work we focus on those based on biological systems.

Spiking neurons are considered the third generation of artificial neural networks, they are processing units described by differential equations that emulate the electrochemical process (in the most realistic case) that occurs in the brain Gertsner (2002). Computationally, the integrate-and-fire model and its variants have been widely used due to fact that they are simplified models which are able to reproduce most of the neural dynamics observed in the brain. The integrate-and-fire neuron model describes the state of a neuron in terms of its membrane potential, which is determined by the synaptic inputs and the injected current that the neuron receives. When the membrane potential reaches a threshold, an action potential (spike) is generated Burkitt (2006). Besides, software and hardware implementations of spiking neurons have been widely performed in recent years Brette (2007); Indiveri (2011).

At hardware level, neuromorphic engineering has emerged as a new research area, which emulates the electrophysiological behaviour of real neurons and conductances by using a hybrid architecture of digital and analog circuits. As was mentioned in Indiveri (2011), the term "neuromorphic" was coined by Carver Mead in the late eighties to refer to artificial neural systems whose architecture and design principles are based on those of biological nervous systems. Even, when the term neuromorphic is strictly focused on hybrid system, we consider our work as part of this area, since we combine digital and analog circuits to provide of locomotion to a hexapod robot.

From all these elements (legged robots, CPG systems, spiking neurons and hardware implementations), we have carried out this research in the endeavor of engineering rhythmic locomotion control-oriented systems. This research work does not attempt to propose a better algorithm or biological model but it tries to draw inspiration and ideas from what has been done, to step forward in the construction of fully practical digital hardware locomotion systems exhibiting similar properties and organization observed in biological systems. For the CPG-based system, this work proposes the use of spiking neurons as processing units, which from the estimation of their synaptic weights we have been able to reproduce different rhythms signals (gaits). A contribution of this research was the application of an alternative method, based on the simplex method, for parameter tuning of the discrete-time spiking neuron model, which is recognized an open issue in CPG-based rhythmic locomotion. Also, this work addresses the design and digital hardware implementation of CPG-based systems to be embedded in a real robot so as to increase the computation speed and relative power consumption of the control system. In this regard, Field Programmable Gates Arrays (FPGAs) appear to fit particularly well spiking-based neural processing thanks to their regular fine-grain parallel computational structure, reconfigurability and the availability of on-chip distributed memory. Moreover FPGA technology is always improving in logic density and speed, which constantly increases the complexity of the models that can be implemented on them by software-like techniques, thus facilitating fast prototyping. While, alternative parallel implementation media such as graphics processing units (GPUs) have been used to speed up computations by using threads at programming levels, major motivating factors for choosing FPGAs are the power-efficiency for embedded applications and the possibility to export an FPGA design to an application specific integrated circuit (ASIC) implementation.

The rest of the paper is organized as follows. Central Pattern Generators, the Spiking Neuron Model and the Parameter Estimation Method are described in Section 2. Then, the design and implementation of FPGA-based architectures for different network topologies are presented (Section 3). In Section 3, we also present the CPG configurations, the system running on the hexapod robot and the obtained numerical results. Finally, we conclude in Section 4.

2. Methods

2.1. Spiking Neuron Model

The Spiking Neuron (SN) model is a discrete-time derivation of the best-known and widely used generalised Integrate-and-Fire model (gIF). This model is called BMS and was initially introduced in Soula (2006). In the BMS model the membrane potential V_i and the firing state Z_i of the ith neuron at time k are given by the following equations:

$$V_i[k] = \gamma V_i[k-1](1 - Z_i[k-1]) + \sum_{j=1}^{N} W_{ij} Z_j[k-1] + I_i^{ext} \tag{1}$$

and,

$$Z_i[k] = \begin{cases} 1 & \text{if } V \geq \theta \text{ (firing)} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\gamma \in [0, 1[$ defines the leak rate. N is the number of neurons. W is the matrix of synaptic weights. Finally $I^{(ext)}$ represents an external stimulus. Hence, when $V_i[k]$ reaches a given threshold θ, then a spike occurs in $Z_i[k]$ Eq. (2) and the neuron i is reset by the term $(1 - Z_i[k])$ in Eq. (1).

Also, Eq. (1) can be written in the following form:

$$V_i[k] = \sum_{j=1}^{N} W_{ij} \sum_{\tau=0}^{\tau_{ik}} \gamma^{\tau} Z_j[k - \tau - 1] + I_{ik\tau} \tag{3}$$

writing $I_{ik\tau} = \sum_{\tau=0}^{\tau_{ik}} \gamma^{\tau} I_i^{(ext)}[k - \tau]$ with:

$$\tau_{ik} = k - \text{arg min}_{l>0}\{Z_i[l-1] = 1\}$$

(see Rostro (2012) for more details about this derivation).

2.2. Parameter estimation for gait generation

In Rostro (2012), it has been demonstrated that from Eq. (1) and Eq. (2) we can derivate a Linear Programming System, which allow us to estimate the parameters (synaptic weights) to generate any desired spike train (neural dynamics). To be more precise, in this work, we performed a reverse engineering process of desired rhythmic spiking dynamics (gaits), which are known but not the values of $V_i[k]$. This fact allows us to establish the next expression:

$$Z_i[k] = 0 \Leftrightarrow V_i[k] < \theta \text{ and } Z_i[k] = 1 \Leftrightarrow V_i[k] \geq \theta,$$

where $\theta = 1$ for simplification. This last expression can now be written as an inequality as follows:

$$e_{ik} = (2\, Z_i[k] - 1)\,(V_i[k] - 1) \geq 0.$$

Substituting Eq. (3) in the last expression we get a Linear Programming System Bixby (1992) and its equation reads:

$$\mathbf{e}_i = \mathbf{A}_i\,\mathbf{w}_i + \mathbf{b}_i \geq 0 \tag{4}$$

where:

$$\begin{aligned}
\mathbf{A}_i &= \begin{pmatrix} \cdots & \cdots & \cdots \\ \cdots & (2\, Z_i[k] - 1)\, \sum_{\tau=0}^{\tau_{jk}} \gamma^\tau Z_j[k-\tau-1] & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix} \\
\mathbf{b}_i &= (\ldots \quad (2\, Z_i[k] - 1)\,(I_{ik\tau} - 1) \quad \ldots)^t \\
\mathbf{w}_i &= (\ldots \quad W_{ij} \quad \ldots)^t \\
\mathbf{e}_i &= (\ldots \quad (2\, Z_i[k] - 1)\,(V_i[k] - 1) \quad \ldots)^t
\end{aligned}$$

The synaptic weights now can be estimated from the set of the linear inequalities (one for each neuron). Such weights will further define the network topology for the different gaits.

The solution to the LP problem defined in Eq. (4) has been performed offline in software by using the Simplex method and the resulting synaptic weights are presented in Section 3.1. The parameter estimation algorithm has been coded in C++ language and we used the GLPK[1] library to solve the simplex method.

2.3. Central Pattern Generator

A Central Pattern Generator (CPG) is a neural network that generate rhythmic signals without the intervention of sensory information coming from the outside. These systems are responsible for the execution of diverse activities in living organisms, such as respiration, digestion, heart beat, among

[1] http://www.gnu.org/software/glpk

others Ijspeert (2008). CPG's core neurons mutually inhibit their activation creating a cyclical pattern.

Although not sensory information is required in CPG systems, the utilisation of external control signals allows to switch between the start, the stop and the modulation of the operating frequency of the system so as to produce adaptive and richer locomotion skills.

Each CPG system is built for a specific task, thus, if we want to execute a different one, we need to change the structure of the CPG. In our case, we designed a CPG system to perform the locomotion of a hexapod robot. Such design is based on a spiking neural network, whose parameters (synaptic weights) are estimated by using the method described in Section 2.2. Here, each leg of the hexapod robot is controlled by two neurons (one for the coxa and one for the femur), which are connected by a different topology depending on the desired gait (see Figs. 2, 3 and 4 for the different phase relationship and activity of neurons for each desired gait). Figure 1 shows twelve red points over the hexapod robot, where each of them represents a neuron and they are positioned on the joints that they will control.

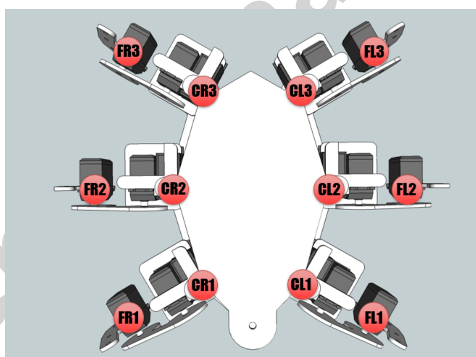


Figure 1: Localizations of neurons in the robot. Here, CL and CR correspond to Coxa Left a Right respectively. On contrary, FL and FR correspond to Femur Left and Right respectively.

Figs. 2, 3 and 4 show the pattern generated for the 12 neurons (6 for the coxa and the same number for the femur) to control each servo in the robot and generate a coordinated locomotion in the robot (see Ijspeert (2008); Marder (2001); Grabowska (2012) for more details about the generation of these locomotion patterns).

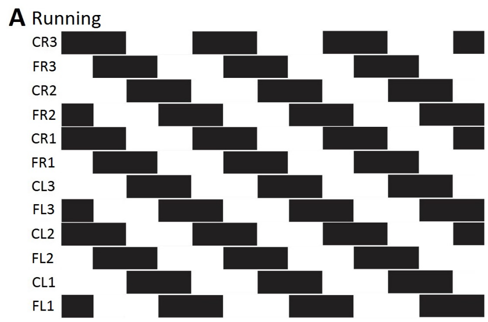


Figure 2: Rhythmic signals generated artificially for a running gait.

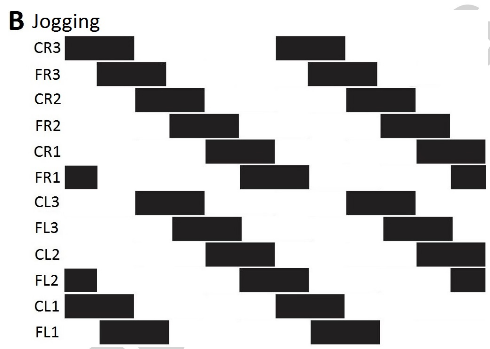


Figure 3: Rhythmic signals generated artificially for a jogging gait.

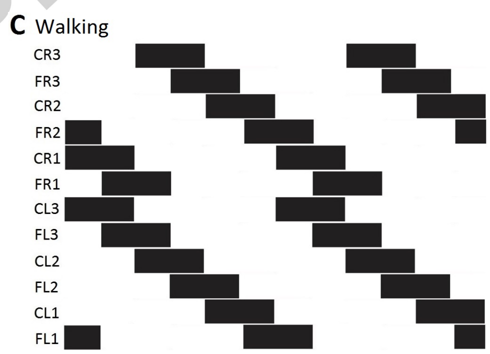


Figure 4: Rhythmic signals generated artificially for a walking gait.

2.4. Hardware

One of the major achievements of this research is the use of hardware platforms to fully implement and validate our CPG system on a real robot. In this regard, we have successfully implemented and synthesized a VHDL design of the CPG system (for three different configurations) on a FPGA. Some of the main advantages of FPGA technology in robotics-related applications are high-speed, truly parallel and distributed processing of control computations, small silicon area and potentially reduced power consumption.

In this work, we used an OpalKelly[2] FPGA prototyping board (Fig. 5) centered around a Spartan 6 XC6SLX45 chip. One of the main advantages of OpalKelly boards is their high level communication protocol facilities making easier the exchange of information between the FPGA and a host computer for debugging and validation purposes.

Furthermore, as we can observe in Eq. (2) the neural dynamics can be represented as a binary code, thus, Z contains the firing states of the whole network. This fact makes the spiking neuron model highly suitable for digital hardware implementations such as FPGA's, since is not necessary to store the values of V.

Figure 5: Spartan 6 FPGA board (XEM6310 Opal kelly)

Also, the FPGA architecture has been tested on a Phoenix Hexapod Robot from Lynxmotion[3] (Fig. 6). This robot has a three degree of freedom (DOF) leg design, each driven by a Hitec HS-645 servo. The leg design corresponds to the three main sections: coxa, femur and tibia. In total, the robot has eighteen DOF, however we only considered twelve due to the fact that the movement of the robot is concentrated on the coxa and the femur.

[2]http://www.opalkelly.com/
[3]http://www.lynxmotion.com/

The electronics consists of a BotBoarduino and a SSC-32 (servo controller) to program and control the robot respectively. Here, we substituted the BotBoarduino with a Spartan 6 FPGA in order to improve the computational capabilities of this robot.

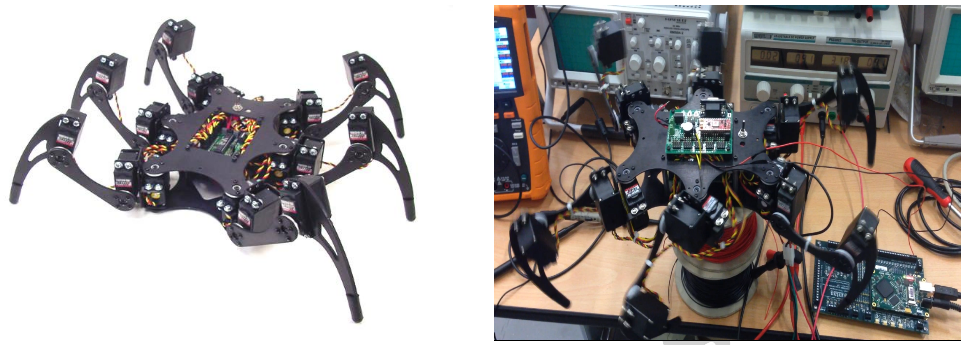


Figure 6: In the left side we show the hexapod robot platform from Lynxmotion and on the right side we show the robot setup.

3. Results

We presented a Central Pattern Generator based on a network of spiking neurons connected by different topologies. These topologies depend directly on the parameters of the network and they are estimated from a reverse engineering process of the desired dynamics (Section 2.2) as those shown in Figs. 2, 3 and 4. From these parameters, the next step is to simulate the network for the different gaits. Then, the network for the three topologies is coded in VHDL for a FPGA-based implementation of the CPG. Finally the FPGA architectures are tested on a hexapod robot.

3.1. Gait topologies

In Eq. (3) we considered $N = 12$, this value corresponds to the number of neurons required to control the servos in the robot. Since, sensory information is not necessary for the locomotion of the robot, the external current could be set to a constant value, however for simplicity we set it to zero. Finally, Z is the spiking activity (rhythmic signal) from which the reverse engineering process is applied, in our case Z corresponds to the rhythmic patterns shown in Figs. 2, 3 and 4.

From the reverse engineering process described in Section 2.2, we obtained the following synaptic weights matrices, where each of them represents a different configurations for the three gait patterns (walking, jogging and running).

$$w_w = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2 & 0 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \end{vmatrix} \quad (5)$$

$$w_j = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 & 2 & -2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \end{vmatrix} \quad (6)$$

$$
w_r = \begin{vmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0
\end{vmatrix} \tag{7}
$$

where w_w, w_j and w_r correspond to the walking, jogging and running configurations respectively.

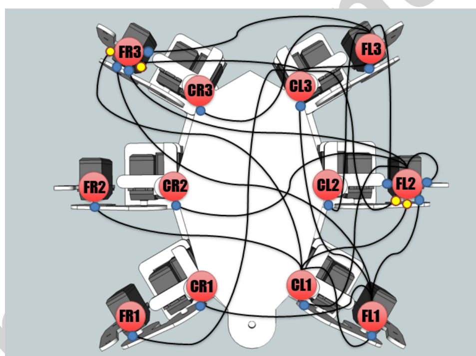


Figure 7: A spiking neural network for walking. The small circles in yellow and blue represent inhibitory and excitatory synaptic connections respectively.

3.2. Hardware implementations

In hardware-based systems design the choice of the appropriate hardware for the implementation is a critical issue, this is mainly due to the limitations of the hardware indeed and the development cost. In this research, due to the fact that each neuron operates as a separate processing unit, we chose a fully parallel implementation taking advantage of the fine-grain parallelism available in FPGAs. In this particular case, a Spartan 6 Opal Kelly board was employed. Using VHDL code, we developed a module called Neuron.

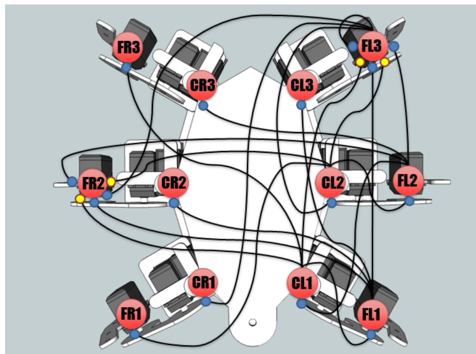


Figure 8: A spiking neural network for jogging. The small circles in yellow and blue represent inhibitory and excitatory synaptic connections respectively.

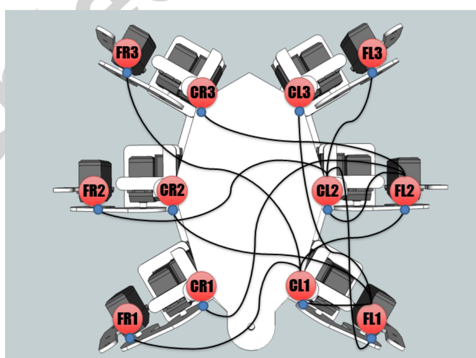


Figure 9: A spiking neural network for running. The small circles in blue represent excitatory synaptic connections.

This module performs all the necessaries operations to obtain the voltage in a neuron. The block diagrams of this module are shown in Figures 10 and 11.

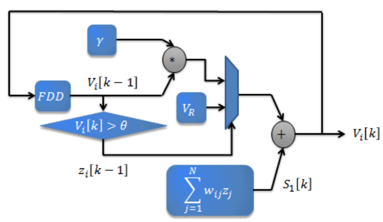


Figure 10: Overall block diagram of the Neuron module

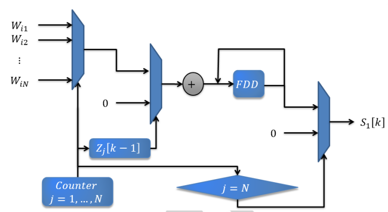


Figure 11: Block diagram responsible for the sum of weights from the matrix W_{ij}

3.2.1. Precision analysis

In Rostro (2011) a precision analysis for the discrete-time spiking neuron model has been performed. This, allows us to identify the best fixed-point representation (or word length) for the digital hardware implementation and consequently have a robust implementation in terms of the exact reproduction of neural dynamics. In such analysis, three different dynamical regimes (neural death, periodic and chaotic dynamics) for a network with discrete-time spiking neurons have been simulated in a FPGA for different word length (from 6 to 32 bits). For each regime, a plot such as that shown in Fig. 12 has been obtained. In this regard, Fig. 12 (a green square for an exact reproduction and a red square for a wrong estimation) shows the precision analysis results for periodic dynamics, such as those generated in the CPG systems.

The 2D plot in Fig. 12 clearly shows that even for a fixed point representation of 6 bits is possible to reproduce any periodic dynamic. This is due to

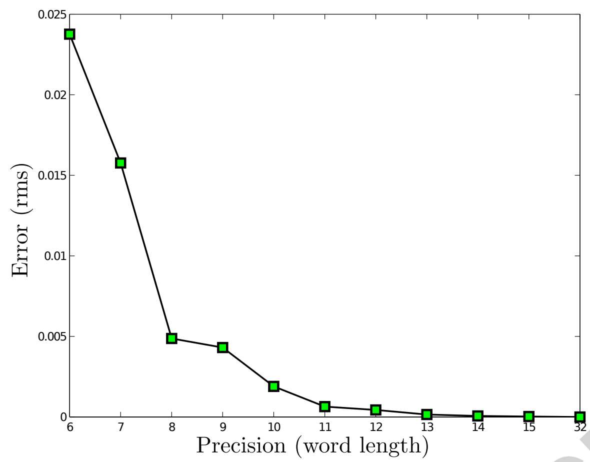


Figure 12: A precision analysis for a discrete-time spiking neuron model, where the small green squares indicate the exact reproduction in hardware of a periodic dynamic for the given word length.

the fact that in periodic dynamics the membrane potential is constantly reset to the resting potential (zero in our case), for such reason the cumulative error is also reset. However, since the network topologies for the different gaits are not too complex (12 neurons and around 20 synaptic connections in the worst case), we decide to use the most common representation, 16 bits (8 bits for the integer part and the same number for the fractional one).

3.2.2. Numerical simulations

The FPGA-based neuron architecture shown in Figures 10 and 11 has been successfully tested in VHDL and implemented on the FPGA for the three gaits and the results are shown in Figs. 13, 14 and 15.

3.2.3. FPGA implementation results

In this section, we present Table 1 with the place and route results of the Central Pattern Generator on a Spartan 6 XC6SLX45 device. As we can observe, it is possible to scale up the system to a most complex one, i.e. increasing the number of spiking neurons and synapses or incorporating sensory processing, since the area consumption is very slow.

Usually, CPG-based robotic locomotion control is programmed in software and running on a CPU/microcontroller. On the other hand, considering that the physics of silicon is analogous to the biophysics of the nervous

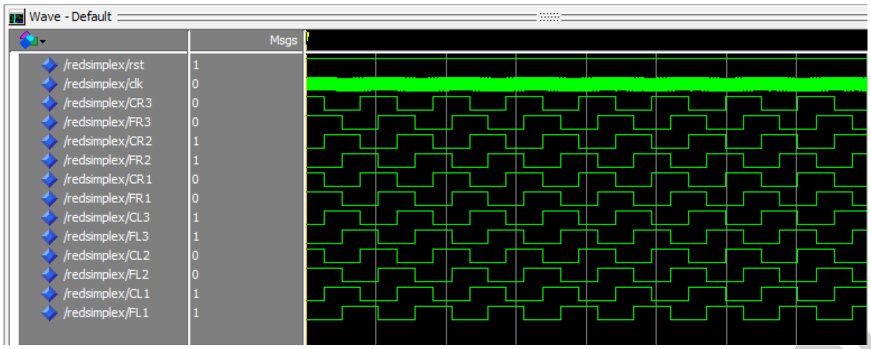


Figure 13: Pattern A (Fig. 2) running on the FPGA.

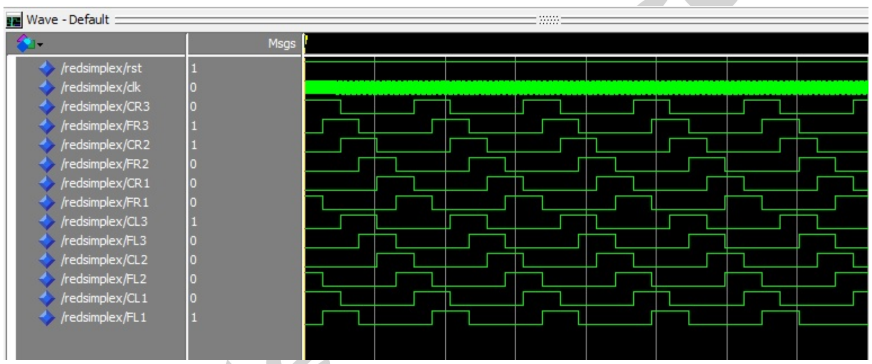


Figure 14: Pattern B (Fig. 3) running on the FPGA.

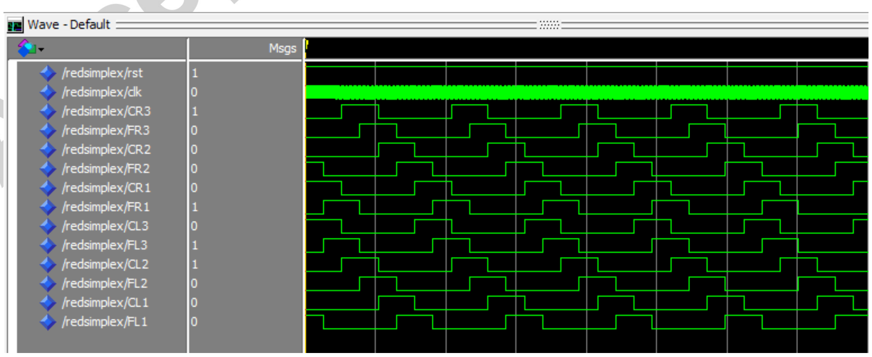


Figure 15: Pattern C (Fig. 4) running on the FPGA.

Module	Registers	LUTs	%
Neuron	38/54,576	68/27,288	0.3
RS-232 COM	194/54,576	247/27,288	0.9
CPG (12 neurons)	255/54,576	549/27,288	2.0
System	449/54,576	796/27,288	2.9

Table 1: Device utilization for a Xilinx Spartan 6 (from OpalKelly)

system, silicon-based integrated circuit technology is used to construct CPG chips. A CPG chip is usually compact and small, and consumes less power but lacks flexibility.

Few efforts have been devoted to fully practical implementations of CPGs in pure digital hardware, such as FPGAs, in recent years. In a related work, Barron (2013) explored the FPGA-based implementation of van de Pol oscillators coupled with a CPG controller to generate adaptive gait patterns for hexapod robots. This implementation when was placed and routed to a Spartan 6 SLX45CSG324 device reports the following resource utilization: 6,476 flip-flops and 6,621 LUTs. Clearly, the proposed implementation in this work consumes less hardware resources thanks to the spiking neurons that are less hardware-greedy than van de Pol oscillators. In fact, our implementation can even be more compact, since we are considering 16 bits, instead of 6 for the fixed-point representation.

4. Conclusion

Our results demonstrate that from a reverse engineering process of a spiking neural network, it is possible to determine the configurations for different gaits in hexapod robot locomotion.

It is true, that the observation that the brain operates on analog principles of the physics of neural computation that are fundamentally different from digital principles in traditional computing, however in hardware design there is always a manner to reverse a digital signal into an analog one. Also, at this moment there is not an analog or hybrid system that emulates a CPG system running on a robot.

In this work we have successfully contributed in the effort of engineering rhythmic locomotion control systems by integrating four areas: (1) a CPG system based on spiking neurons, (2) a method based on reverse engineering to estimate any desired gait or neural dynamic, (3) an FPGA-based implementation of the CPG system and (4) an implementation running on a hexapod robot for a fully embedded implementation.

Also, the results of this implementation in a FPGA give us the option to design new digital CPG systems. In addition, even with a low cost Spartan 6 device, we can generate more complex neural networks taking advantage of its capabilities.

Acknowledgments

This work has been partially funded by the CONACYT Projects 229839 (*Apoyo al Fortalecimiento y Desarrollo de la Infraestructura Científica y Tecnológica 2014*) and 99912, the SEP-PRODEP project (Apoyo a la incorporacin de NPTC) and the DAIP Project 444 (*Convocatoria Institucional de Investigación Científica 2014*).

References

H. Soula, G. Beslon, O. Mazet, *Spontaneous dynamics of asymmetric random recurrent spiking neural networks*. Neural Computation **18** (2006) 60–79.

H. Rostro-Gonzalez, B. Cessac and T. Vieville, *Parameters estimation in spiking neural networks : a reverse-engineering approach* Journal of Neural Engineering **9** (2), 026024, (2012) pp. 1-14.

R. B. Darst, *Introduction to Linear Programming Applications and Extensions*. Marcel Dekker Ltd, New-York (1990).

R. E. Bixby, *Implementing the Simplex Method: The Initial Basis*. Journal on Computing **4** (3) (1992).

A. J. Ijspeert, *Central pattern generators for locomotion control in animals and robots: A review*. Neural Networks, **21** (2008) 642–653.

H. Rostro-Gonzalez, B. Cessac, B. Girau and C. Torres-Huitzil, *he role of the asymptotic dynamics in the design of FPGA-based implementations of gIF-type neuron models*. Journal of Physiology-Paris, **1-3** (2011) 91–97.

E. Marder and D. Bucher *Central pattern generators and the control of rhythmic movements*. Current Biology, **11** (23), (2001) 986-996.

T. G. Brown, *On the nature of the fundamental activity of the nervous center; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system.* Journal of physiology, **18**, (1914) 18–46.

P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. 1st edn., The MIT Press (2005).

W. Gerstner and W. Kistler, *Spiking Neuron Models*. 1st edn., Cambridge University Press (2002).

AN. Burkitt, *A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input.* Biological Cybernetics **95** (1) (2006) 1–19.

AN. Burkitt, *A review of the integrate-and-fire neuron model: II. Inhomogeneous synaptic input and network properties.* Biological Cybernetics **95** (2) (2006) 97–112.

R. Brette, M. Rudolph, T. Carnevale, M. Hines, D. Beeman, J. M. Bower, M. Diesmann, A. Morrison, P. H. Goodman, F. C. Harris, Jr., M. Zirpe, T. Natschlager, D. Pecevski, B. Ermentrout, M. Djurfeldt, A. Lansner, O. Rochel, T. Vieville, E. Muller, A. P. Davison, S. El Boustani and A. Destexhe, *Simulation of networks of spiking neurons: a review of tools and strategies.* Journal of Computational Neuroscience **23** (3) (2007) 349–98.

G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S-C. Liu, P. Dudek, P. Hafliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang and K. Boahen *Neuromorphic silicon neuron circuits.* Frontiers in Neuroscience. **5**(73), doi: 10.3389/fnins.2011.00073 (2011).

J. H. Barron-Zambrano and C. Torres-Huitzil, *FPGA implementation of a configurable neuromorphic CPG-based locomotion controller*. Neural Networks. **45** 50–61 DOI=10.1016/j.neunet.2013.04.005 http://dx.doi.org/10.1016/j.neunet.2013.04.005 (2013).

M. Grabowska, E. Godlewska, J. Schmidt and S. Daun-Gruhn, *Quadrupedal gaits in hexapod animals inter-leg coordination in free-walking adult stick insects*. The Journal of Experimental Biology. **215** 4255–4266 (2012).