IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 14, 2017, accepted April 7, 2017, date of publication April 24, 2017, date of current version June 7, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2696985

A FPGA-Based Neuromorphic Locomotion System for Multi-Legged Robots

ERICK ISRAEL GUERRA-HERNANDEZ[1], ANDRES ESPINAL[2], PATRICIA BATRES-MENDOZA[1], CARLOS HUGO GARCIA-CAPULIN[1], RENE DE J. ROMERO-TRONCOSO[1], (Senior Member, IEEE), AND HORACIO ROSTRO-GONZALEZ[1]

[1]Department of Electronics Engineering, Universidad de Guanajuato, Guanajuato 36885, Mexico
[2]Department of Organizational Studies, Universidad de Guanajuato, Guanajuato 36885, Mexico

Corresponding author: Horacio Rostro-Gonzalez (hrostrog@ugto.mx)

This work was supported by the CONACYT Project Aplicación de la Neurociencia Computacional en el Desarrollo de Sistemas Robóticos Biológicamente Inspirados under Grant 269798.

ABSTRACT The paper develops a neuromorphic system on a Spartan 6 field programmable gate array (FPGA) board to generate locomotion patterns (gaits) for three different legged robots (biped, quadruped, and hexapod). The neuromorphic system consists of a reconfigurable FPGA-based architecture for a 3G artificial neural network (spiking neural network), which acts as a Central Pattern Generator (CPG). The locomotion patterns, are then generated by the CPG through a general neural architecture, which parameters are offline estimated by means of grammatical evolution and Victor-Purpura distance-based fitness function. The neuromorphic system is fully validated on real biped, quadruped, and hexapod robots.

INDEX TERMS FPGA, spiking neural networks, neuromorphic engineering, legged robots, grammatical evolution, Victor-Purpura distance.

I. INTRODUCTION

The neuromorphic engineering, an emerging research field, imitates neural architectures which are found in nervous systems of living beings by using hybrid architectures with digital and analog components. The specific neuromorphic architecture depends directly on the complexity of the neuron model and its connectivity within a neural network [1]. In this regard, in the late 1980s, the first designs of neuromorphic systems based on a VLSI (Very Large Scale Integration) analog architecture were introduced [2]. Some years later, the first Field Programmable Gate Array (FPGA) was designed by Xilinx. However, the use of FPGAs in the design of neuro-biological architectures started to be widely used ten years ago [3]–[7]. At present, neuromorphic systems are designed as hybrid architectures, which take advantage of the analog structure to design realistic neuron models and the digital one to handle the synaptic connections in neural networks with complex topologies [1].

Furthermore, robotic systems based on neuromorphic hardware have been recently studied [8]–[11]. In these papers, it has been demonstrated that the use of neuromorphic hardware is highly suitable to accomplish different robotic tasks. Specifically, locomotion in legged robots may be achieved by plausible neural mechanisms known as Central Pattern Generators (CPGs); these allow robots to successfully move through harsh environments. The basis of CPGs was settled in [12], where evidence points out that walking action is promoted by neural mechanisms in which neurons are inhibiting each other to control the bending and tension of muscles involved in such action. Biologically, CPGs contribute to unconscious movements such as digestion, breathing, locomotion (walking, swimming, etc.) among others by generating rhythmic patterns without endogenous stimulus [13]. CPG models have been implemented in software and hardware [1], [14], their abstraction degree may vary depending on the application as well as on the neuron model used to build them; most CPG implementations are built using neuron models with a low biological plausibility, e.g. oscillators [15]–[17].

Recently, CPGs are being modeled by using spiking neuron models with different leves of biological plausibility [18]–[23]. Such models, one of the main research areas in computational Neuroscience [24] are processing units of the third generation of artificial neural networks, which are known as Spiking Neural Networks (SNNs) [25]. Spiking neuron models have different abstraction degrees, from very abstracted ones such as integrate-and-fire neurons to highly detailed and complex models such as Hodkin-Huxley neurons [15]. Most of spiking neurons, imitate with certain plausibility degree, the electrochemical process that occurs in the brain by means of differential equations [26]. Focusing on the integrate-and-fire model, it describes the evolution over time of a neuron's membrane potential by integrating afferent information: action potentials from presynaptic neurons and injected currents from external sensory media; once the membrane potential reaches a threshold, the neuron fires an action potential [27], [28].

In this work, we combine all these elements with the purpose of develop and implement rhythmic locomotion control-oriented systems. To be specific, we design SNNs based on a discrete-time neuron model to automatically generate locomotion patterns such as those produced by CPGs and similarly to previous works presented in [21]–[23]. However, one of the main contributions of this paper compared to those presented in Rostro's and Espinal's works is in the design of the neural network, a hard task to accomplish when using CPG-based rhythmic locomotion [15]. Herein, we consider a grammar-based form of genetic programming, known as Grammatical Evolution (GE), for configuring the synaptic parameters (connections and weights) of the SNN. Compared to the Grammars proposed in [22] and [23], GE is a more compact grammar in terms of its implementation in software and provide us better results in terms of the neural network topology, i.e. less synaptic connections, which is highly desirable for hardware-based implementations. Besides, with this methodology we can explore several configurations instead of training explicitly predefined neural network architectures. Specifically, the generated SNNs can performed at least three different locomotion gaits per network (walk, trot and run), it is each configuration can reproduce the three different locomotion gaits without any external stimulus. Another contribution of this work, is the use of an another spike train (locomotion patterns expressed as neural code) metric known as the Victor-Purpura distance ($D^{spike}[q]$) [29], [30], which defines the distance between two spike trains in terms of the minimum cost of transforming one spike train into another.

Here, we also make a contribution in the design and implementation of digital architectures of CPG-based systems to be embedded in real robots so as to increase the computation speed and relative power consumption of the control system. In this regard, 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.

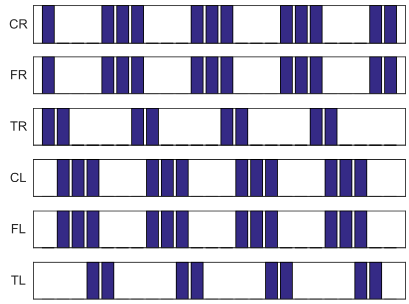


FIGURE 1. Gait (rhythmic signals) for walking locomotion in a biped robot.

Finally, the last contribution of this work is the successful validation of our approach on three robot platforms, i.e. biped, quadruped and hexapod.

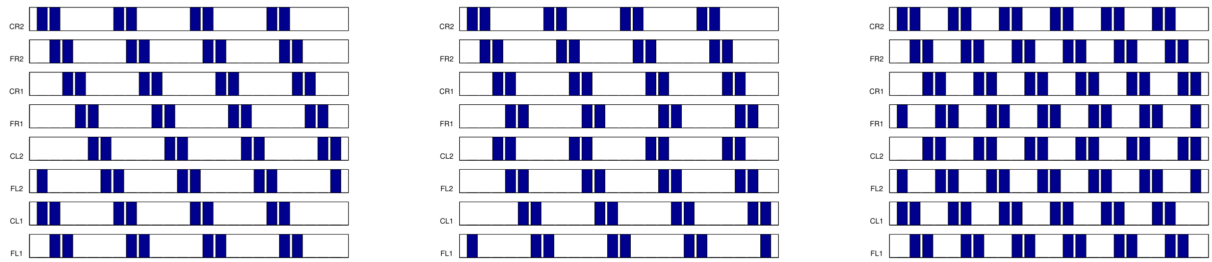
The structure of the paper is organized as follows: CPGs, the spiking neuron model, the method for defining the parameters of neural networks and the utilized hardware are described in Section II. Then, hardware implementations and validation on biped, quadruped and hexapod robots are presented in Section III. Finally, we conclude in Section IV.

II. MATERIALS & METHODS

This paper covers all aspects from the design of SNNs (or CPGs) to their hardware implementation in a FPGA and real-time validation on legged robots. In this regard, to automatically design CPGs, modelling and modulation aspects are defined by an Evolutionary approach, the Grammatical Evolution (GE). The CPG configuration generates a SNN that can reproduce a specific rhythmic locomotion pattern (gait). The gait transition is made by reseting the membrane potentials of all neurons and setting the initial state of the spike trains according to the gait to accomplish. Finally, the different configurations for the CPG are implemented on a SPARTAN 6 FPGA board and validated on three legged real robots (biped, quadruped and hexapod).

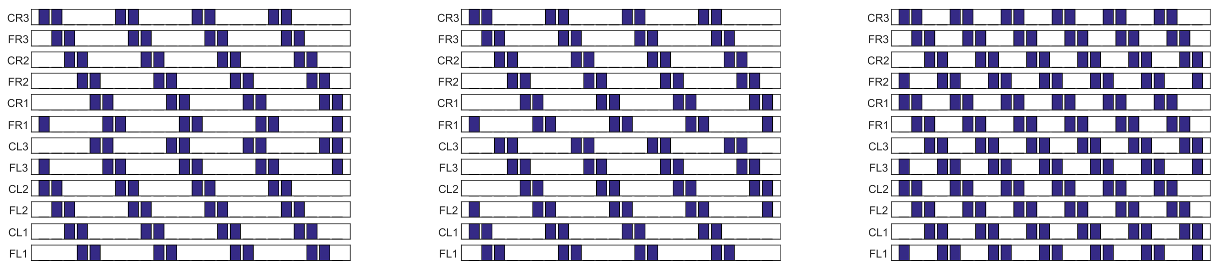
A. CENTRAL PATTERN GENERATOR

Central Pattern Generators (CPGs) are specialized neural systems involved in the locomotion of living beings through rhythmic patterns, which are endogenously generated. In robotics, CPGs have been mainly studied for locomotion in legged robots. Several interesting features have been reported about the advantages of using CPG-based locomotion robots instead other methods (finite-state machines, sine-generators, pre-recorded reference trajectories or heuristic control laws), such as: rhythmicity, stability, adaptability and variety [16]. Despite these features, there is still a lack of a well-established CPG design methodology [15]. In this regard, CPG design is mainly focus on three following aspects: modeling and analysis when the general architecture of the CPG and the type and topology of couplings are set,



(a) (b) (c)

FIGURE 2. Gaits (rhythmic signals) for locomotion in a quadruped robot. (a) Walking gait. (b) Jogging gait. (c) Running gait.



(a) (b) (c)

FIGURE 3. Gaits (rhythmic signals) for locomotion in a hexapod robot. (a) Walking gait. (b) Jogging gait. (c) Running gait.

modulation when its parameters are tuned to deal with the gait transition, and the implementation, which can be realized in software or hardware [16]. Locomotion skills performed by CPGs are commonly improved by incorporating external control signals, even when not sensory information is required by the CPG.

As was aforementioned, this paper deals with the locomotion of three different kind of legged robots (biped, quadruped and hexapod) by means of CPGs built as SNNs. In Figs. 1, 2 and 3, we show the rhythmic patterns used for this research. Figure 1 shows a walking locomotion gait for a biped robot, which was artificially designed. Figures 2 and 3 show biologically inspired locomotion patterns based on the study presented in [31]; these patterns have been previously used in [21]–[23] for hexapod and quadruped robot locomotion respectively. In these figures, the x-axis represents time of action (as reference) and the y-axis shows the labels of the corresponding servo-motors for each legged robot (see Figs. 4a, 4b and 4 for the correspondence of labels in locomotion patterns).

B. SPIKING NEURON MODEL

The CPG consists of a neural network with spiking neurons, which produces rhythmic patterns known as locomotion gaits (walk, trot and run). To do this, we used the explicit discrete-time spiking neuron model [32]. In such 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:

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

and,

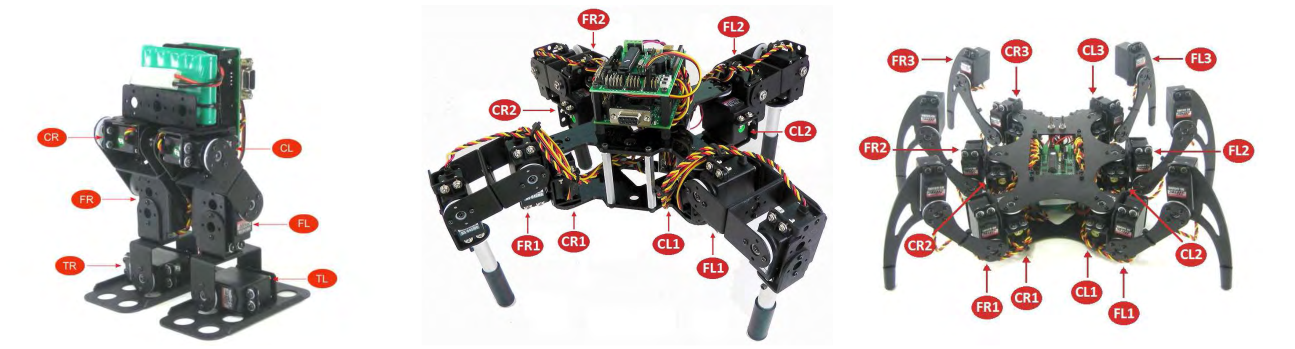
$$Z_i[k] = \begin{cases} 1 & \text{if } V \geq \theta \text{ (firing)} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where $\gamma \in [0, 1]$ is the leakage current. N is the number of neurons (number of servomotors to be controlled). W is the connectivity matrix. 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).

From Eq. (1), we estimate the membrane potential for a neural network with N neurons. In our case, N is equal to 6 for the biped robot (see Fig. 4a), 8 for the quadruped robot (see Fig. 4b) and 12 for the hexapod robot (see Fig. 4), since we have a one-to-one correspondence between neurons and servomotors (or Degrees Of Freedom). The neural network's topology is estimated by means of an evolutionary method and is described in the following section.

C. ESTIMATION OF NEURAL NETWORK PARAMETERS

In this section, we describe a method for the automatic generation of CPGs based on spiking neurons. The proposed method replicates an input locomotion gait represented as a set of spike trains by estimating both, the topology and the synaptic weights of the SNNs. The method finds out for each spiking neuron into the CPG, where the signals (spike trains) from other spiking neurons (or itself) contribute to generate



(a) (b) (c)

FIGURE 4. One-to-one correspondence between neurons (in red) and servos for biped, quadruped and hexapod robots. Here, labels are as follow: C and F correspond to Coxa and Femur; L and R to Left and Right side and the number is the position of the servo in the robot. (a) Biped robot. (b) Quadruped robot. (c) Hexapod robot.

its target spike train; this is made by taking into account only the input locomotion gait without having yet a spiking neuron that reproduce them. Once, all the spike trains can be reproduced, all the configured neurons are integrated into the CPG to work as a whole.

The Algorithm 1 shows how the method works, it requires as input the gait to be produced and its output is a CPG which reproduces it. The design is made as follows: for each spiking neuron to replicate a specific target spike train into the gait, is configured both, PS is a set of the available spike trains which could contribute to the generation of the target spike train and S_{g_i} is the initial state of the spike train of the current spiking neuron to be designed (lines 2 to 7). After, the design routine **GE_Design**(S_{g_i}, PS) (line 9) is called; this is recalled until the generated spike train S_{g_i} is exactly equals to target spike train S_i (lines 8 to 10). For this work, we explore the usage of a different grammar-based form of Genetic Programming [33] instead of using Christiansen Grammar Evolution (CGE), the routine **GE_Design**(S_{g_i}, PS) is a stochastic optimization method known as Grammatical Evolution [34], this optimizer allows to define the presynaptic connections and weights of a spiking neuron by means of an indirect search; besides it avoids the necessity of an explicit learning process.

The method must generate words with the structure shown in Figure 5, where the i-th neuron id goes from 1 to N (neurons into the network) and the i-th weight can be a positive or negative integer value between 1 and 9.

To generate words as aforementioned by means of GE (with standard mapping process, Depth-First [35]), there are factors that need to be defined. For this work, GE was configured as follows:

- A Backus-Naur Form (BNF) grammar to derive words as in Fig. 5. The Grammar 1 shows the one used for this work.
- A search engine to evolve the solutions. For this work, we used Differential Evolution (DE) [36] with DE/Best/1 as mutation scheme (see [37] for implementation details), the use of GE with DE as search engine is known as Grammatical Differential Evolution [38].
- A Fitness function to evaluate the quality of the solutions. Here, we took as basis one of the fitness function proposed in [22] and [23] where it is configured to reproduce a set of different locomotion patterns (gaits) with the same spiking neural network topology. Instead of using SPIKE-distance, we explored the use of another one, the Victor-Purpura distance [29], [30] (see [39] for implementation details).

Algorithm 1 Method for the Automatic Design of CPGs Based on SN

Require: $G = \{S_1, \ldots, S_N\}$
Ensure: SCPG design
1: **for all** $i = 1$ **to** N **do**
2: Set the available signals for the i-th spiking neuron ($PS = \{G - S_i\}$).
3: **if** S_i contains time = 0 **then**
4: $S_{g_i} = \{0\}$
5: **else**
6: $S_{g_i} = \{\}$
7: **end if**
8: **while** $S_{g_i}! = S_i$ **do**
9: $Conn_i$ = **GE_Design**(S_{g_i}, PS)
10: **end while**
11: Add presynaptic connectivity for the i-th spiking neuron into the SCPG according to $Conn_i$
12: **end for**

$$\overbrace{id_{1-st}, weight_{1-st}}^{1-st\ configured\ synapse}\ \_ \ldots \_\ \overbrace{id_{m-th}, weight_{m-th}}^{m-th\ configured\ synapse}$$

FIGURE 5. Generic form of a derived word for presynaptic weighted connections of a postsynaptic neuron.

Due the GE's nature, a word with the structure as in Figure 5 may define several connections from the same presynaptic neuron; this means that a neuron id in the string

(a)

(b)

FIGURE 6. **Electronics used in this research. (a) FPGA board. (b) Servo controller board.**

$$
\begin{array}{rcl}
\langle\text{syanpses}\rangle & \models & \langle\text{synapse}\rangle \mid \langle\text{synapse}\rangle\_\langle\text{synapses}\rangle \\
\langle\text{synapse}\rangle & \models & \langle\text{neuronID}\rangle,\langle\text{sign}\rangle\langle\text{digit}\rangle \\
\langle\text{neuronID}\rangle & \models & 1 \mid ... \mid N \\
\langle\text{sign}\rangle & \models & + \mid - \\
\langle\text{digit}\rangle & \models & 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{array}
$$

Grammar 1. **Grammar to define presynaptic weighted connections of a postsynaptic neuron.**

may be repeated. When a neuron *id* is repeated in a generated string, their weights are added to obtain only one connection per neuron *id* into the word.

D. HARDWARE

To validate our method, we performed a reconfigurable low-level design, which allows us to implement efficiently the three different CPGs as an embedded system for the three robots. The design has been successfully implemented and synthesized in VHDL and mapped into a XEM6010 FPGA (see Fig. 6a) with a Spartan 6 XC6SLX45 chip from OpalKelly.[1] FPGA-based implementations are highly suitable for robotic applications for their high-speed, truly parallel and distributed processing, reconfigurability and potentially reduced power consumption. The FPGA-output is connected to a SSC32 (Fig. 6b), which is a 32 channels of 1uS resolution servo controller board with bidirectional communication. Thus, the FPGA sends the locomotion gaits to activate in synchronicity the servos through the SSC32. The XEM6010 FPGA provides a high level communication protocol to make easier the exchange of information between the FPGA and a host computer for debugging and validation purposes.

Also, we have performed real-time implementations on the three robots: a Brat Biped (Fig. 4a), a Symmetric Qudruped (Fig. 4b) and a Phoenix Hexapod (Fig. 4) from Lynxmotion. It is, the hardware design has been embedded to the robots and the FPGA controls their locomotion from the patterns generated by the CPGs. The CPGs have the ability to reproduce different locomotion gaits (walk for the three kind of robots and, trot and run for the quadruped and hexapod robots) as can be observed in the video provided with this paper.

[1] http://www.opalkelly.com/

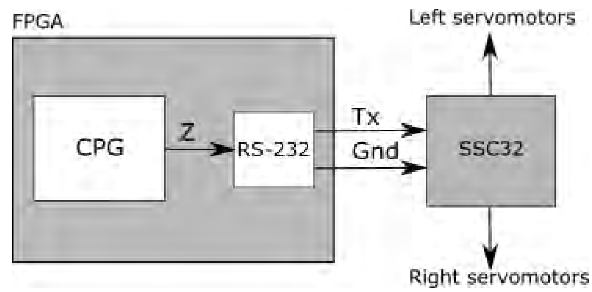


FIGURE 7. **System block diagram.**

CR3	forward	forward	back	back
FR3	up	down	down	up
CR2	back	back	forward	forward
FR2	down	up	up	down
CR1	forward	forward	back	back
FR1	up	down	down	up
CL3	back	back	forward	forward
FL3	down	up	up	down
CL2	forward	forward	back	back
FL2	up	down	down	up
CL1	back	back	forward	forward
FL1	down	up	up	down

FIGURE 8. **Running gait interpreted as instructions for locomotion of hexapod robot.**

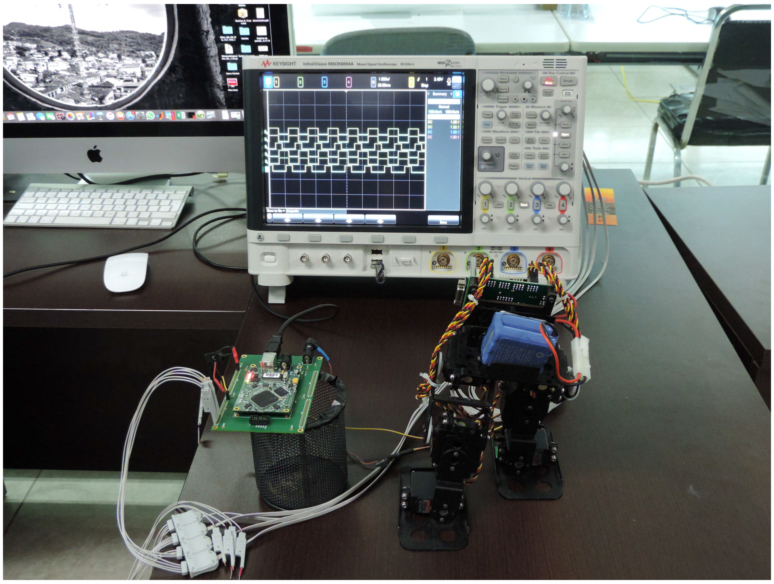


FIGURE 9. Biped robot setup.

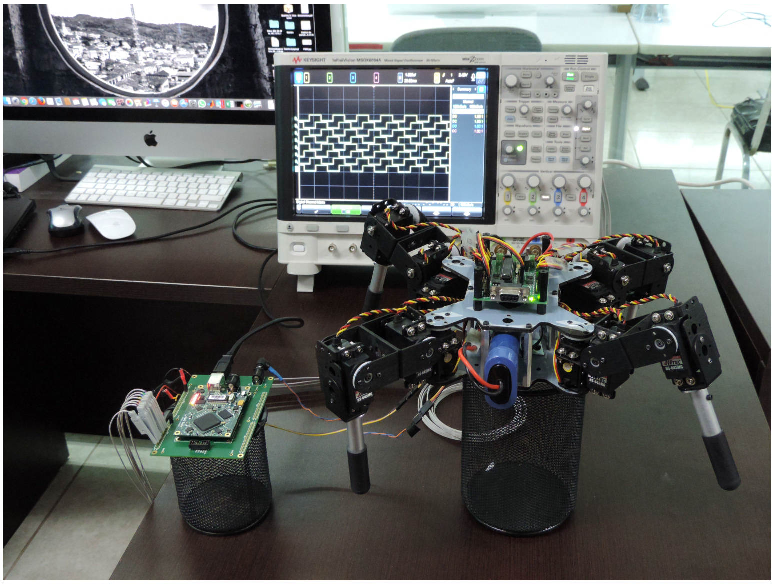


FIGURE 10. Quadruped robot setup.

These robots have a three degree of freedom (DOFs) 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 robots have six, twelve and eighteen DOFs respectively, however we only considered eight for the quadruped robot and twelve for the hexapod robot due to the fact that the movement of these robots is focused in the coxa and femur sections. According to the servomotor data sheet, its control signal consist of positive going pulses raising from 0.5 to 2.5 mS (milliseconds) long, repeated 50 times per second (every 20 ms). The servo position shift in proportion to the pulse-width, i.e., every 100ms equivalent to 9 degrees. In Fig. 7 we show the complete system block diagram.

TABLE 1. Connectivity matrix for gait locomotion in a biped robot.

	CR	FR	TR	CL	FL	TL
CR	0	0	9	-9	0	0
FR	-9	9	-7	7	0	0
TR	-6	9	-7	7	0	0
CL	0	0	-10	0	7	-1
FL	9	0	0	-11	0	9
TL	6	0	0	-1	-3	4

Fig. 8 shows the first four repetitions of the rhythmic signal for the hexapod robot performing the running gait and how

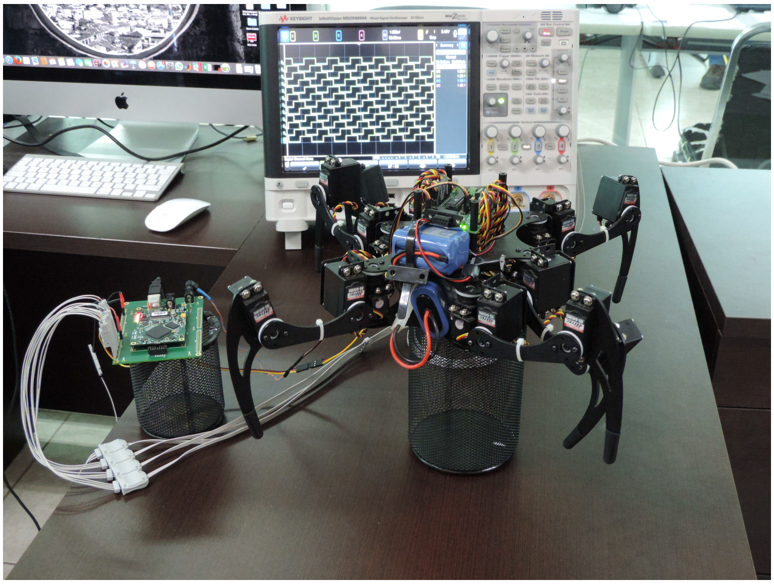


FIGURE 11. Hexapod robot setup.

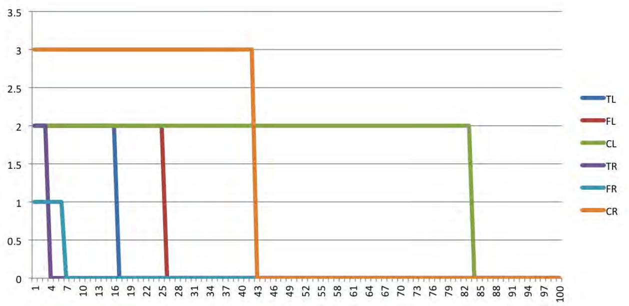


FIGURE 12. CPG design evolution process for the biped robot locomotion.

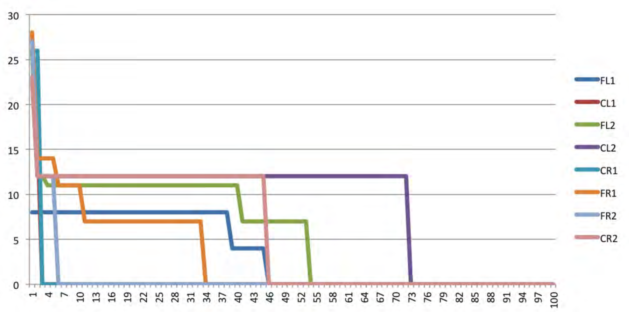


FIGURE 13. CPG design evolution process for the quadruped robot locomotion.

these patterns are translated into movements to the robot joints. Black squares indicate that the value of Z (see Eq. (2)), for that joint is 1, at some point in the time, and 0 in the case of a grey square. This fact makes the spiking neuron model highly suitable for digital hardware implementations such as FPGAs, since it is not necessary to store the values of V, but only its firing state represented by Z.

TABLE 2. Connectivity matrix for gaits locomotion in a quadruped robot.

	CR2	FR2	CR1	FR1	CL2	FL2	CL1	FL1
CR2	0	8	0	0	0	1	-3	-2
FR2	0	0	3	0	0	0	0	0
CR1	0	0	0	6	0	0	0	0
FR1	0	0	-5	0	5	0	1	0
CL2	0	0	0	0	0	1	0	0
FL2	0	0	0	0	-5	0	8	0
CL1	0	0	0	0	0	0	0	1
FL1	9	0	2	0	0	0	-9	0

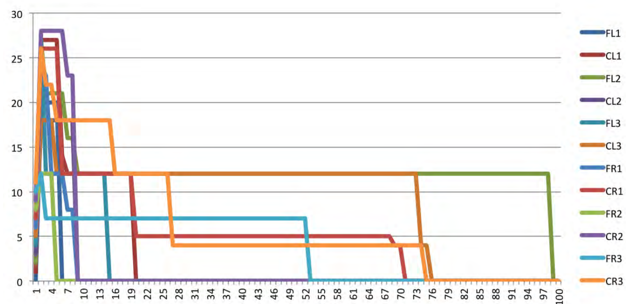


FIGURE 14. CPG design evolution process for the hexpad robot locomotion.

In Fig. 9, Fig. 10 and Fig. 11 we present the systems setup for each robot, where we also show a real time simulation on an oscilloscope for the different gaits configuration.

III. EXPERIMENTS & RESULTS

In this paper, we presented a central pattern generator based on spiking neurons, which with the same configuration is able to generate three different patterns for a legged robot. The CPG's parameters depend on the number of DOFs (neurons) to control and they are estimated

TABLE 3. Connectivity matrix for gaits locomotion in a hexapod robot.

	CR3	FR3	CR2	FR2	CR1	FR1	CL3	FL3	CL2	FL2	CL1	FL1
CR3	-1	5	0	0	0	0	0	0	0	0	0	0
FR3	0	0	5	0	0	0	0	0	0	0	0	0
CR2	0	0	0	2	0	0	0	-1	0	0	0	0
FR2	0	0	0	0	9	0	0	0	0	0	0	0
CR1	0	0	0	0	0	3	0	0	0	0	0	0
FR1	0	0	-9	0	-9	0	0	6	0	5	7	0
CL3	0	0	0	0	0	0	-8	9	0	0	0	0
FL3	0	0	0	0	0	0	0	0	7	0	0	0
CL2	0	0	0	0	0	0	0	0	0	7	0	0
FL2	0	0	0	0	0	0	0	0	0	0	7	0
CL1	0	0	0	0	0	0	0	0	0	0	0	5
FL1	0	2	0	0	0	0	4	0	0	0	-8	4

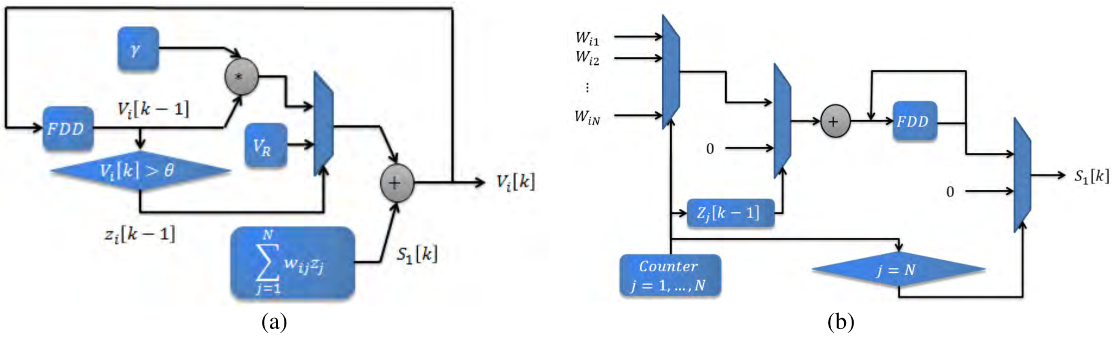


FIGURE 15. Block diagram of Neuron module's VHDL implementation. (a) Overall block diagram of the Neuron module. (b) Block diagram responsible for the sum of weights from the matrix W_{ij}.

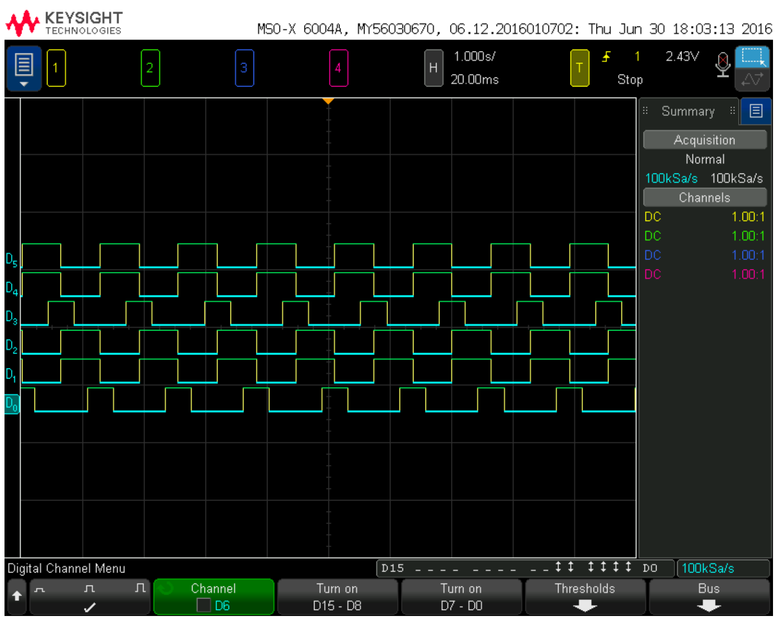


FIGURE 16. Real time waves for the Biped Robot. (https://www.dropbox.com/s/nenljiuxsr2i0zs/DSCN4267.MOV?dl=0).

from the evolution of the indirect representations of the connectivity patterns by a GE of the desired dynamics. To validate, our method we have performed a low-level hardware-based design and we embedded it to three different robot platforms.

A. GAIT TOPOLOGIES

The configuration of the GE-based estimation method for generation CPGs was as follow. For each legged robot the CPG generates its locomotion gaits given an initial state of the desired gait. The DE algorithms ran for all experiments with a

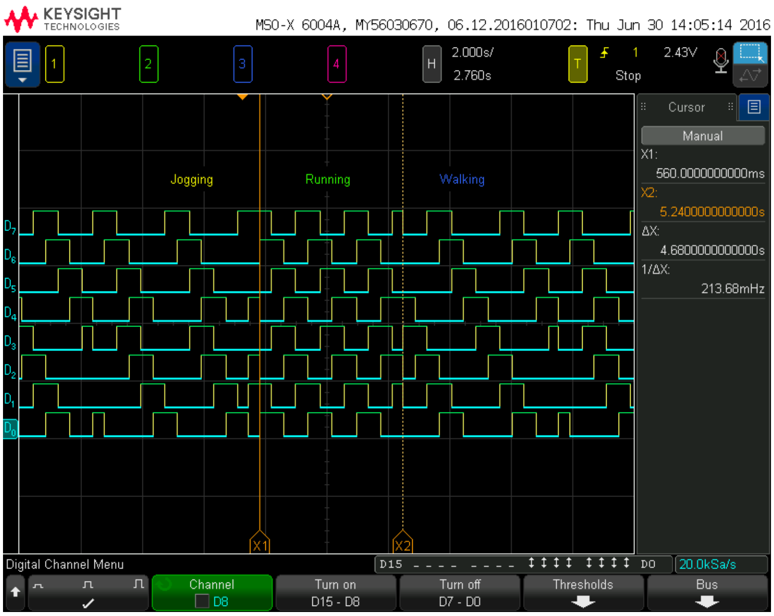


FIGURE 17. **Real time waves for the Quadruped Robot. (https://www.dropbox.com/s/724lo099grk68gv/DSCN4263.MOV?dl=0).**

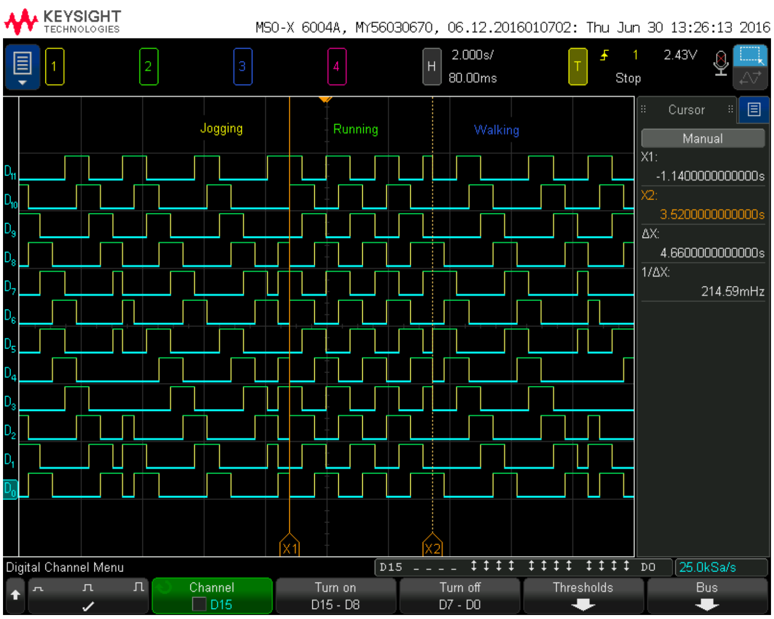


FIGURE 18. **Real time waves for the Hexapod Robot. (https://www.dropbox.com/s/5h8gd73xeg9grfm/DSCN4258.MOV?dl=0).**

population of 5 individuals formed as a real vector of length of 75 ($0 \leq x_i \leq 255$, $i = \{1, \ldots, 75\}$), a crossover rate of $Cr = 0.9$, a differentiation factor $F = 0.8$ and 500 function calls; the DE's parameters were empirically established. The Victor-Purpura distance's penalty factor q was set to 1, due that this value is the minimum integer that can be assigned for not being the spike count metric (D^{count}) with $q = 0$.

Figure 12 shows the evolution of the CPG estimation process for the biped robot, the topology and synaptic configuration are shown in table 1. Figure 13 shows the evolution of the CPG estimation process for the quardruped robot, the topology and synaptic configuration are shown in table 2. And Figure 14 shows the evolution of the CPG estimation process for the hexapod robot, the topology and synaptic configuration are shown in table 3.

B. HARDWARE IMPLEMENTATIONS

We have designed a fast, compact and reconfigurable FPGA-based architecture to map the central pattern generators, which provides of locomotion mechanisms to multi-legged robots. We proved the speed of our implementations by running real-time locomotion experiments on the

TABLE 4. Hardware utilization for different gaits patterns.

Gait pattern	LUTs	Flip-flops	DSP48A1	(%) Utilization		Power (mW)	
				LUTs	Flip-flop	Dynamic	Quiescent
Biped Walk	626/27288	383/54576	0/58	2.29	0.70	1	38
Quadruped Walk	857/27288	492/54576	0/58	2.29	0.90	1	38
Quadruped Jog	857/27288	494/54576	0/58	3.14	0.91	1	38
Quadruped Run	988/27288	496/54576	0/58	3.62	0.91	1	38
Hexapod Walk	836/27288	492/54576	0/58	3.06	0.90	1	38
Hexapod Jog	849/27288	492/54576	0/58	3.11	0.90	1	38
Hexapod Run	938/27288	488/54576	0/58	3.44	0.89	1	38

TABLE 5. Power consumption.

Hardware	Current(mA)	Power(mW)
FPGA	115	828
SSC32	25	180
Servomotors (Biped)	1000	7200
Servomotors (Quadruped)	2000	14400
Servomotors (Hexapod)	3000	21600

three robots. In table 4 we show that our architecture is very compact, since we only consumed around the 3% of the total of available resources in the chip. It is true that this percentage corresponds specifically to implementations mapped on a Spartan 6 FPGA board and can be different for another device. Finally, we created three topologies (for biped, quadruped and hexapod) for the CPGs to test the reconfigurability of our design.

Specifically, we created a module called Neuron in order to generate the activity of the neuron model described by Eq. (1). This module is shown in Figure 15.

1) FPGA IMPLEMENTATION RESULTS

The FPGA-based neuron architectures shown in Figures 15a and 15b have been implemented on the FPGA for the three robots, we show the real time waves for the gait patterns in Figs. 16, 17 and 18. The real time waves were obtained with a Keysight oscilloscope MSO-X 6004A with 16 digital channels. In these figures, we have also included WEB links to some videos showing the performance of our robots. In such videos, we can observe the transition among the different gaits.

In Table 4, hardware utilization for the different gaits patterns mapped on a Spartan 6 XC6SLX45 device is presented. The whole system represents only a 3% of resources consumption, which allows us to scale up the system to a more complex one, i.e. a more biologically plausible neuron model, increasing the size of the network or incorporating sensory processing. In addition, Table 5 presents the power consumption for each part of the system, which demonstrates the low power consumption of our system.

IV. CONCLUSION

CPGs are usually based on coupled oscillators, which have shown a good performance, however in this paper we have presented a biologically-inspired approach and its hardware implementation in a FPGA. The hardware architecture is a SNN that act as CPGs configured to achieve several gaits for different robots (one artificially-inspired for biped robot and three biologically-inspired for quadruped and hexapod robot). The CPGs designed for locomotion systems have been implemented on Biped, Quadruped and Hexapod robots, thus the methodology is susceptible to be applied to any legged robot when biological or artificial rhythmic patterns are available, both cases are used in this paper.

The spiking neuron model used in this work consumes less hardware resources and power than those based on coupled oscillators. So give us the option to re-design new digital CPG systems. In addition, even with a low cost Spartan 6 device, we may be abled to generate more complex neural networks taking advantage of its capabilities.

Our results demonstrate that from the Grammatical Evolution (GE) we are able to tune the parameters of a spiking neural network (Central Pattern Generator) to determine the configurations for different robots. Even when GE is a simpler grammar-based Genetic Programming algorithm than CGE, it achieves simpler (with fewer connections) CPGs that generate several rhythmic patterns than those obtained by CGE for hexapod and quadruped robots reported in [23] and [22], respectively. Besides, Victor-Purpura distance was successfully applied to drive the search process as SPIKE-distance was in previous works.

Although we have designed, implemented and tested locomotion systems for different legged-robots, it is necessary to emphasize how difficult is to obtain biologically-inspired or design artificial rhythmic patterns for locomotion either for other or the same kind of legged-robots. Moreover, several aspect must be explored for getting more robust locomotion systems; such as: implementing strategies to smooth transition between gaits instead of resetting and re-initializing the SNN, incorporating sensors (cameras, infrared, etc.) to make responsive locomotion systems by allowing the modulation of CPG through external stimuli and testing our locomotion system over other platforms besides of FPGAs.

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 the Grammatical Evolution (GE) and Victor-Purpura distance to estimate any desired gait or neural dynamic, (3) an FPGA-based implementation of the CPG system and (4) an implementation on a different legged robot for a fully embedded implementation.

REFERENCES

[1] G. Indiveri *et al.*, "Neuromorphic silicon neuron circuits," *Frontiers Neurosci.*, vol. 5, p. 73, May 2011.

[2] C.-A. Mead, *Analog VLSI and Neural Systems*. Reading, MA, USA: Addison-Wesley, 1989.

[3] A. Cassidy, S. Denham, P. Kanold, and A. Andreou, "FPGa based silicon spiking neural array," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Nov. 2007, pp. 75–78.

[4] A. Cassidy and A.-G. Andreou, "Dynamical digital silicon neurons," in *Proc. IEEE Biomed. Circuits Syst. Conf.*, Nov. 2008, pp. 289–292.

[5] K.-L. Rice, M.-A. Bhuiyan, T.-M. Taha, C.-N. Vutsinas, and M.-C. Smith, "FPGA implementation of Izhikevich spiking neural networks for character recognition," in *Proc. Int. Conf. Reconfigurable Comput. (FPGAs)*, 2009, pp. 451–456.

[6] A. Cassidy, A. G. Andreou, and J. Georgiou, "Design of a one million neuron single FPGA neuromorphic system for real-time multimodal scene analysis," in *Proc. 45th Annu. Conf. Inf. Sci. Syst. (CISS)*, Mar. 2011, pp. 1–6.

[7] R.-M. Wang, G. Cohen, K.-M. Stiefel, T.-J. Hamilton, J.-C. Tapson, and A. van Schaik, "An FPGA implementation of a polychronous spiking neural network with delay adaptation," *Frontiers Neurosci.*, vol. 7, p. 14, Feb. 2013.

[8] J.-L. Krichmar and H. Wagatsuma, *Neuromorphic and Brain-Based Robots*. Cambridge, U.K.: Cambridge Univ. Press, 2011.

[9] C. Bartolozzi *et al.*, "Embedded neuromorphic vision for humanoid robots," in *Proc. CVPR WORKSHOPS*, 2011, pp. 129–135.

[10] L.-D. Bucci, T.-S. Chou, and J.-L. Krichmar, "Sensory decoding in a tactile, interactive neurobot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2014, pp. 1909–1914.

[11] S. Menon, S. Fok, A. Neckar, O. Khatib, and K. Boahen, "Controlling articulated robots in task-space with spiking silicon neurons," in *Proc. 5th IEEE RAS/EMBS Int. Conf. Biomed. Robot. Biomechatron.*, Aug. 2014, pp. 181–186.

[12] T.-G. Brown, "On the nature of the fundamental activity of the nervous centres; together with an analysis of the conditioning of rhythmic activity in progression, and a theory of the evolution of function in the nervous system," *J. Physiol.*, vol. 48, no. 1, p. 18, 1914.

[13] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current Biol.*, vol. 11, no. 23, pp. 986–996, 2001.

[14] R. Brette *et al.*, "Simulation of networks of spiking neurons: A review of tools and strategies," *J. Comput. Neurosci.*, vol. 23, no. 3, pp. 349–398, Jul. 2007.

[15] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Netw.*, vol. 21, no. 4, pp. 642–653, 2008.

[16] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on CPG-inspired control models and system implementation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 441–456, Mar. 2014.

[17] J.-H. Barron-Zambrano and C. Torres-Huitzil, *CPG Implementations for Robot Locomotion: Analysis and Design*. Amsterdam, The Netherlands: INTECH Open access Publisher, 2012.

[18] M. A. Lewis, F. Tenore, and R. Etienne-Cummings, "CPG design using inhibitory networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr. 2005, pp. 3682–3687.

[19] A. Russell, G. Orchard, and R. Etienne-Cummings, "Configuring of spiking central pattern generator networks for bipedal walking using genetic algorithms," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2007, pp. 1525–1528.

[20] A. Russell *et al.*, "Optimization methods for spiking neurons and networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1950–1962, Dec. 2010.

[21] H. Rostro-Gonzalez *et al.*, "A CPG system based on spiking neurons for hexapod robot locomotion," *Neurocomputing*, vol. 170, pp. 47–54, Dec. 2015.

[22] A. Espinal *et al.*, "Quadrupedal robot locomotion: A biologically inspired approach and its hardware implementation," *Comput. Intell. Neurosci.*, vol. 2016, p. 13, May 2016.

[23] A. Espinal, H. Rostro-Gonzalez, M. Carpio, E.-I. Guerra-Hernandez, M. Ornelas-Rodriguez, and M.-A. Sotelo-Figueroa, "Design of spiking central pattern generators for multiple locomotion gaits in hexapod robots by christiansen grammar evolution," *Frontiers Neurorobot.*, vol. 10, no. 6, pp. 1–13, Jul. 2016.

[24] P. Dayan and L.-F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modelling of Neural Systems*. Cambridge, MA, USA: MIT Press, 2005.

[25] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, 1997.

[26] W. Gerstner and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[27] A.-N. Burkitt, "A review of the integrate-and-fire neuron model: I. homogeneous synaptic input," *Biological*, vol. 95, no. 1, pp. 1–19, 2006.

[28] A.-N. Burkitt, "A review of the integrate-and-fire neuron model: II. Inhomogeneous synaptic input and network properties," *Biological*, vol. 95, no. 2, pp. 97–112, 2006.

[29] J. D. Victor and K. P. Purpura, "Nature and precision of temporal coding in visual cortex: A metric-space analysis," *J. Neurophysiol.*, vol. 76, no. 2, pp. 1310–1326, 1996.

[30] J. D. Victor and K. P. Purpura, "Metric-space analysis of spike trains: Theory, algorithms and application," *Netw. Comput. Neural Syst.*, vol. 8, no. 2, pp. 127–164, 1997.

[31] M. Grabowska, E. Godlewska, J. Schmidt, and S. Daun-Gruhn, "Quadrupedal gaits in hexapod animals-inter-leg coordination in free-walking adult stick insects," *J. Experim. Biol.*, vol. 215, no. 24, pp. 4255–4266, 2012.

[32] H. Soula, G. Beslon, and O. Mazet, "Spontaneous dynamics of asymmetric random recurrent spiking neural networks," *Neural Comput.*, vol. 18, no. 1, pp. 60–79, Jan. 2006.

[33] J.-R. Koza, *Genetic Programming: On The Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992, vol. 1.

[34] C. Ryan, J.-J. Collins, and M. O'Neill, "Grammatical evolution: Evolving programs for an arbitrary language," in *Proc. 1st Eur. Workshop Genet. Program.*, 1998, pp. 83–95.

[35] D. Fagan, "Analysing the genotype-phenotype map in grammatical evolution," Ph.D. dissertation, School Comput. Sci. Inf., Univ. College Dublin, Dublin, Ireland, 2013.

[36] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[37] V. Feoktistov, *Differential Evolution*. New York, NY, USA: Springer, 2006.

[38] M. O'Neill and A. Brabazon, "Grammatical differential evolution," in *Proc. Int. Conf. Artif. Intell. (ICAI)*, Las Vegas, NV, USA, 2006, pp. 231–236.

[39] D. Aronov, "Fast algorithm for the metric-space analysis of simultaneous responses of multiple single neurons," *J. Neurosci. Methods*, vol. 124, no. 2, pp. 175–179, 2003.

ERICK ISRAEL GUERRA-HERNANDEZ received the B.S. degree in electronics and the M.S. degree (Hons.) in electronics from the Benemérita Universidad Autónoma de Puebla, Mexico, in 2003 and 2007, respectively. He is currently pursuing the Ph.D. degree with the Universidad de Guanajuato, Mexico. His current research interests include digital designs in FPGAs, bio-inspired systems, robotics, and computational vision.

ANDRES ESPINAL received the B.S. degree in computational systems engineer and the master's degree in computer science from the Tecnologico Nacional de México–Instituto Tecnológico de León, in 2009 and 2011, respectively, and the Ph.D. degree in computer science from the Tecnológico Nacional de México–Instituto Tecnológico de Tijuana. He is currently a full-time Professor with the Department of Organizational Studies, Universidad de Guanajuato. He has authored several journals and proceedings papers. His research lines cover: evolutionary algorithms, artificial neural networks, computer vision, digital image processing, and bio inspired algorithms.

PATRICIA BATRES-MENDOZA received the B.E. and M.S. degrees in software engineering from Instituto Tecnológico de Orizaba in 2001 and 2003, respectively. She is currently pursuing the Ph.D. degree with the Universidad de Guanajuato, Mexico. His current research interests include the brain signal processing, quaternion, and computational neuroscience.

CARLOS HUGO GARCIA-CAPULIN received the B.S. degree in electronics engineering from the Instituto Tecnológico de Ciudad Madero in 1998, the master's degree in electrical engineering from the Universidad de Guanajuato in 2004, and the Ph.D. degree in optics from the Centro de Investigaciones en Óptica, A. C., in 2014. He is currently a full-time Professor with the Department of Electronics, Universidad de Guanajuato. He has authored several journals and conferences proceedings papers. His research lines cover: computational intelligence, evolutionary algorithms, bio-inspired computing and robotics, reconfigurable electronics, and parallel computing.

RENE DE J. ROMERO-TRONCOSO (M'07–SM'12) received the Ph.D. degree in mechatronics from the Autonomous University of Queretaro, Queretaro, Mexico, in 2004. He is a National Researcher level 3 with the Mexican Council of Science and Technology, CONACYT. He is currently a Head Professor with the Universidad de Guanajuato and an Invited Researcher with the Autonomous University of Queretaro, Mexico. He has been an Advisor of over 200 theses, an author of two books on digital systems (in Spanish), and a co-author of over 150 technical papers published in international journals and conferences. His fields of interest include hardware signal processing and mechatronics. He was a recipient of the 2004 Asociación Mexicana de Directivos de la Investigación Aplicada y el Desarrollo Tecnológico Nacional Award on Innovation for his work in applied mechatronics, and the 2005 IEEE ReConFig Award for his work in digital systems. He is part of the editorial board of the *Hindawi's International Journal of Manufacturing Engineering*.

HORACIO ROSTRO-GONZALEZ received the B.S. degree in electronic engineering from Instituto Tecnológico de Celaya, Mexico, in 2003, the M.E. degree (Hons.) in electrical engineering from the Universidad de Guanajuato, Mexico, in 2006, and the D.Sc. degree in computational neuroscience from the University of Nice Sophia Antipolis, France, in 2011. From 2011 to 2012, he was a Research Fellow with the University of Cyprus, where he was involved in the FP7 EU SCANDLE Project. Since 2012, he has been a Full Professor with the Department of Electronics, Universidad de Guanajuato, Mexico. He has authored around 40 journals and conferences proceeding papers. His current research interests include reconfigurable electronic, neuromorphic engineering, computational neuroscience, bio-inspired algorithms, parallel computing, and signal processing.