Robot Locomotion Control Using Central Pattern Generator with Non-linear Bio-mimetic Neurons

Vijay Shankaran Vivekanand
Electrical and Computer Engg.
University of Pittsburgh
Pittsburgh,USA
viv40@pitt.edu

Shahin Hashemkhani
Electrical and Computer Engg.
University of Pittsburgh
Pittsburgh, USA
shh199@pitt.edu

Shanmuga Venkatachalam
Electrical and Computer Engg.
University of Pittsburgh
Pittsburgh, USA
shanmugav@pitt.edu

Rajkumar Kubendran
ECE and Bioengineering
University of Pittsburgh
Pittsburgh, USA
rajkumar.ece@pitt.edu

2023 9th International Conference on Automation, Robotics and Applications (ICARA) | 978-1-6654-8921-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICARA56516.2023.10125666

***Abstract*— Central pattern generators (CPG) generate rhythmic gait patterns that can be tuned to exhibit various locomotion behaviors like walking, trotting, etc. CPGs inspired by biology have been implemented previously in robotics to generate periodic motion patterns. This paper aims to take the inspiration even further to present a novel methodology to control movement of a four-legged robot using a non-linear bio-mimetic neuron model. In contrast to using regular leaky integrate and fire (LIF) neurons to create coupled neural networks, our design uses non-linear neurons constituting a mixed-feedback (positive and negative) control system operating at multiple timescales (fast, slow and ultraslow ranging from sub-ms to seconds), to generate a variety of spike patterns that control the robotic limbs and hence its gait. The use of spikes as motor control signals allows for low memory usage and low latency operation of the robot. Unlike LIF neurons, the bio-mimetic neurons are also jitter tolerant making the CPG network more resilient and robust to perturbations in the input stimulus. As a proof of concept, we implemented our model on the Petoi Bittle bot, a quadruped pet dog robot and were able to reliably observe different modes of locomotion–walk, trot and jump. Four bio-mimetic neurons forming a CPG network to control the four limbs were implemented on Arduino microcontroller and compared to a similar CPG built using four LIF neurons. The differential equations for both neurons were solved real-time on Arduino and profiled for memory usage, latency and jitter tolerance. The CPG using bio-mimetic non-linear neurons used marginally higher memory (378 bytes, 18% higher than LIF neurons), incurred insignificant latency of 3.54ms compared to motor activation delay of 200ms, while providing upto 5-10x higher jitter tolerance.**

Keywords—central pattern generators, non-linear neurons, spike-based control, neuromorphic computing, neuromodulation

I. Introduction

Central pattern generators (CPG) using biological neurons are known to actuate movement in both invertebrates and vertebrate animals [1]. There has been extensive research on using CPGs as a method for controlling locomotion of robots [2]. Neuromodulation is considered intrinsic to CPG networks and required for its proper activation. However, simple neuron models such as the Rectified Linear Unit (ReLU) or the leaky integrate and fire (LIF) neuron used in modern ANNs and SNNs cannot exhibit neuromodulatory behavior at a nodal level [3].

Biological neurons exhibit rich non-linear dynamics both at the single neuron level and at the network level, which is enabled through neuromodulation, using multiple feedback paths (local and global) operating at multiple timescales [4]. Complex neuron models inspired by neurophysiology have been proposed before, like Hodgkin-Huxley model [5] and the Izhikevich model [6]. These models capture the biophysics of the single neuron accurately but are based on non-intuitive computationally expensive differential equations that cannot be tuned to modulate network level behavior. Neuromodulation of a single neuron can be efficiently implemented using a non-linear circuit model with mixed feedback paths (positive and negative), operating at different timescales – fast (sub-ms), slow (ms) and ultraslow (>100ms) [7].

Previous studies have demonstrated complex feedback-based control [8] with simple LIF neurons. Our efforts are more directed towards using the proposed non-linear bio-mimetic neuron as a basis for controlling the locomotion of a robot which has not been attempted before, to the best of our knowledge. Our work bio-mimetic CPGs is more relevant to robotics once we factor in the actual inspiration for using CPGs in the first place was to mimic efficient multimodal motor control seen in animals. CPGs found in biology invariably show rhythmic burst patterns which are not possible to implement with a LIF neuron that produces spikes only. The proposed non-linear neuron, on the other hand, is capable of spiking and bursting, with excitability tuning for neuromodulation at a network level.

In order to reap the benefits of having multiple intrinsic modes on the neuron and network level one must first establish that the non-linear neuron model is capable of generating basic motor behavior such as walking, jumping and trotting seamlessly. We believe that non-linear neurons are the way forward for complex motor control in advanced robotics and this work provides an initial proof-of-concept as shown in Figure 1.

This paper is organized as follows. Section II provides a brief background of the non-linear neuron model and spiking CPGs. Section III describes the implementation of the spiking CPGs on Arduino integrated with the Petoi bot. Section IV provides the benchmarking results of the CPG network using the non-linear neurons compared with that using LIF neurons, for different modes of locomotion. Section V summarizes our contributions.

II. Background

A. Non-linear Bio-mimetic Neuron Model

We base our neuron modelling on the circuit architecture proposed in [7]. This architecture makes use of several voltage controlled current sources inspired by the *Na+/K+* ion channels commonly found in neurophysiology [4]. They are connected in parallel with a capacitor and a resistor. These voltage-controlled current sources are modelled with IV characteristics shown in equations (1)-(3). These voltage-gated conductance channels channels provide positive and negative feedback in addition to the applied external stimulus current, I_{app} and passive leakage currents, thereby forming a mixed-feedback system. Moreover, each conductance element is tuned to operate at multiple distinct timescales in order to achieve modular control of the excitability properties of the entire neuron circuit.

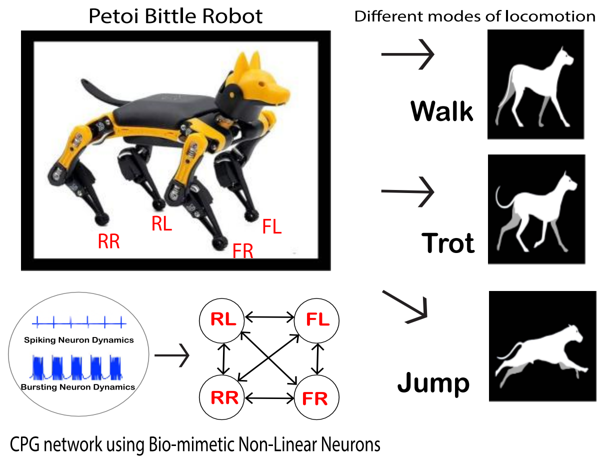

Fig. 1. Overview of system level working of robot locomotion control using a CPG conceived with non-linear neurons. Four of these non-linear neurons are fully connected to form a network where each neuron is responsible for locomotion of a limb. Here, FL is Front Left, FR is Front Right, RR is Rear Right and RL is Rear Left. The different modes of locomotion are achieved using different weight matrices to monitor synaptic connections and by carefully tuning the input stimulus I_{app}.

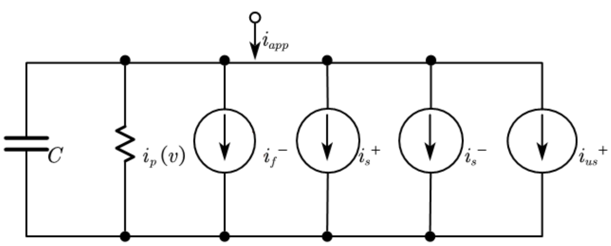

Fig. 2. Neuron cell circuit. Adopted from [7]. Passive RC network in parallel with four conductance elements operating at three timescales. The non-linear feedback currents are fast negative, slow positive, slow negative and ultra-slow positive. A current source I_{app} is added to provide external input stimulus.

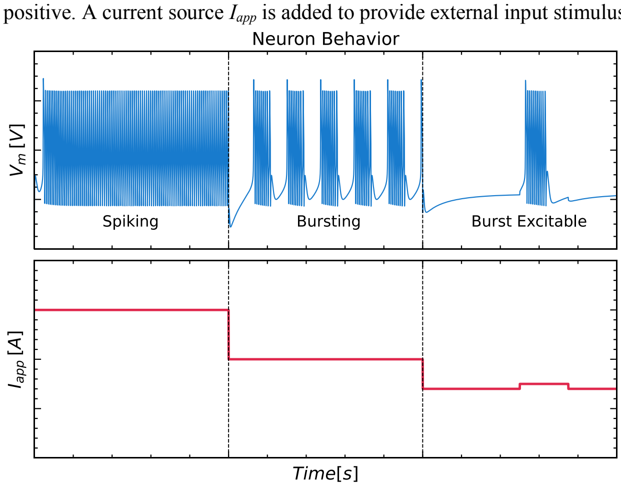

Fig. 3. Variation of neuron behavior, from spiking to bursting to burst excitable modes with respect to changes in I_{app}. This paper uses the first spiking mode to generate patterns used to control a quadruped robot.

Our work implements a neuron model with 4 conductance elements, forming 2 positive, 2 negative feedback loops and operates at 3 different timescales – fast, slow and ultraslow, as shown in Fig. 2. The fast negative conductance (i_f^-) operates in sub-ms range, the slow positive (i_s^+) and slow negative (i_s^-) conductances operate in ms and ultra-slow positive conductance (i_{us}^+) operates in the range of seconds. Each conductance element incorporates a hyperbolic tangent ($tanh$) transfer function, which combined together enables a complex non-linear activation function for the neuron. The non-linearity is particularly important at the fast and slow timescales, which can be seen as 'N shaped' IV curve for the corresponding conductance channel. This 4th order system can be viewed as a superposition of 2 second order systems with fast-slow and slow-ultra slow timescales. By ensuring that the timescales are separate enough it is guaranteed that the circuit has a stable spiking behavior when the applied current is between a defined range. This behavior is explained by limit cycle oscillations between high and low voltage ranges. The amplitude range of the spikes generated from the system is controlled by two parameters δ_f^- and α_f^-. The spiking frequency is controlled mainly by α_s^+ but at the same time it can be modulated using the applied current. Non-linear differential equation describing the neuron model,

$$C_m \frac{dV_m}{dt} = I_{app} - I_p(V_m) - \sum I_x^{\pm} \quad (1)$$

$$where,\ I_x^{\pm} = \pm\alpha_x^{\pm}\, tanh(V_x - \delta_x^{\pm})\ and\ I_p(V_m) = V_m \quad (2)$$

Differential equation to set delay of different timescales,

$$\tau_x \frac{dV_x}{dt} = V_m - V_x \ where\ \tau_f \ll \tau_s \ll \tau_{us} \quad (3)$$

Fig. 3 shows the software simulation of the neuron model provided by the equations (1)-(3), where the neuron output can be tuned using the input I_{app}, to exhibit spiking, bursting and burst excitable modes.

B. Spiking CPGs for Locomotion Control

Periodic activities like breathing and walking require some level of co-ordination not controlled by the brain. This temporal co-ordination occurs at the lower end of the nervous system hierarchy in biology, since it is highly time critical. Spiking Neural Networks (SNN) fit this bill perfectly by offering extremely high levels of energy efficiency on devices with limited compute and memory.

In [8] the method proposed consists of controlling a hexapod with a spiking central pattern generator with feedback from cameras and gyroscopes and uses a reinforcement learning scheme to train the weights. A fully connected spiking central pattern generator based on leaky integrate and fire neurons are used to learn timings for 'walking'. When the neuron fires, it sets off a series of events which includes lifting, rotating and landing the leg. The fully trained SCPG will integrate the neurons with external stimulus till they fire and send post synaptic currents to all the neurons which have an excitatory synapse with the firing neuron. This sets off a chain reaction causing all the neurons to have an increase in their respective membrane voltages. The synaptic weights between these neurons are trained to ensure that the other neurons fire only in the order that they are meant to.

Reference [8] also talks about preserving balance by taking snapshots of the gyroscope output while moving the legs as a part of the training process in order to avoid moving more than3 legs at a particular time. Another, important problem that needs to be addressed is that the system might realize that not moving the limbs at all is also a stable point as per the system restrictions, but this is not intended and moving less than 1 limb for a given time stamp is also met with a penalty in the cost function. This entire setup ensures that the system does not go off balance by moving more than 3 limbs at any time whilst also moving forward.

III. Implementation

Our first task was to faithfully reproduce the spiking mode of the neuron presented in [7] on a software testbench. This was done using Euler's method to approximate the solution of the differential equation (1). We have used a parameterized sparse firing neuron model for this work since the physical movement of the robot leg i.e. motor activation takes a significantly longer time (~200ms). The weight matrices for different modes of operation were derived from Linear Programming as shown in [9].

Once we had the neuron spiking as expected, next task was to tune the different parameters namely α_f^-, α_s^+, α_s^-, α_{us}^+, δ_f^-, δ_s^+, δ_s^- and δ_{us}^+ to obtain the optimal value for sparse spiking since we will eventually connect multiple neurons with one another and having a sparse spike pattern allows for more synaptic currents to be added without making the system spike too many times. This was done by sweeping through a range of values for all eight of these parameters and looking at the corresponding spike patterns. The optimal values which ensured that the neuron remained in the spiking mode whilst also giving sparse activity are summarized in Table I.

TABLE I. Optimal Values of parameters for Nonlinear Neuron to Produce Sparse Spiking Activity

Parameter	Value	Parameter	Value
α_f^-	-2	δ_f^-	0
α_s^+	2	δ_s^+	0
α_s^-	-1.5	δ_s^-	-0.88
α_{us}^+	2	δ_{us}^+	0

Now, the next step was to use this sparse firing neuron and create synaptic connections between four of these neurons, one for each leg of the robot. These synaptic weights for these connections should be tunable to increase the connectivity as per our requirement as well as be able to completely cut off connectivity to make the neurons behave as if it were never connected in the first place.

To achieve this, we introduced a NxN weight matrix (here N is the total number of neurons with all possible connections made between any two neurons), a scale factor to tune the effectiveness of the synapses as a whole and synaptic currents as per the membrane voltages of the other neurons. This new synaptic current from all neighboring neurons was added to the primary neuron alongside the preexisting currents as shown in (2) $\{I_x$, I_p and $I_{app}\}$ this was then integrated to calculate the new membrane voltage.

In order to get different modes of locomotion the weight matrix was tuned with the help of Linear Programming [9] and by calibrating the physical motion required by the leg to actually perform a given mode of locomotion (jump,walk or trot) we were able to come up with different weight matrices. Fig. 4 shows the raster plot of the spikes generated by the CPG network with 4 neurons which act as motor activation events to enable different modes of locomotion.

The next challenging task was to detect the spiking of a non-linear neuron and convert it into event frames where motor commands are to be sent to the bot. This was done by solving the differential equations for each time step real-time on the Arduino microcontroller (NY Board V1_0). The spikes generated by the 4 non-linear neurons produced a rhythmic pattern over an interval of time which were used as the motor activation events for actuating each individual limb of the robot.

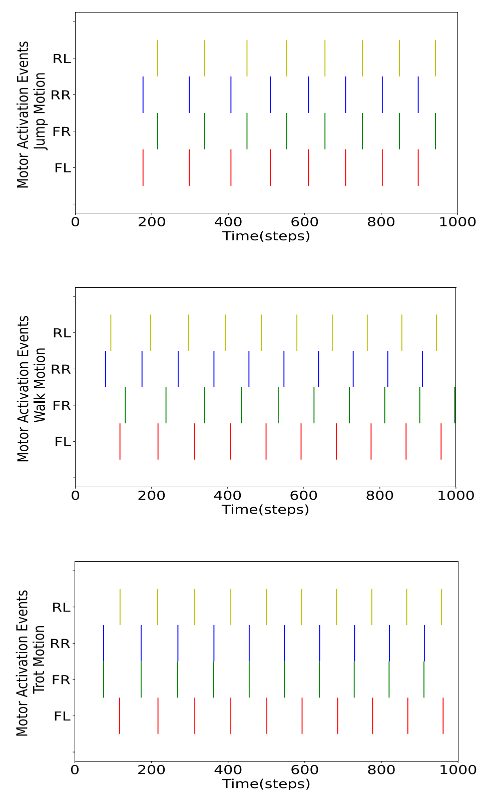

Fig. 4. Raster plot of motor event triggers for different modes of operation.

A. Tuning the Network

To tune the model to be in the spike mode of operation we first gradually increase the applied current in the first neuron whilst keeping the other parameters like alpha and delta same. This gradual increase in I_{app} causes the first neuron to cause spikes periodically.We consider this as our base I_{app},this ensures that any isolated neuron/the primary neuron in any network is always causing spikes. Now, to tune the neurons with strong synaptic connections to the primary neuron we start with base $I_{app}/2$ and tune further to produce desired frequency of spikes on

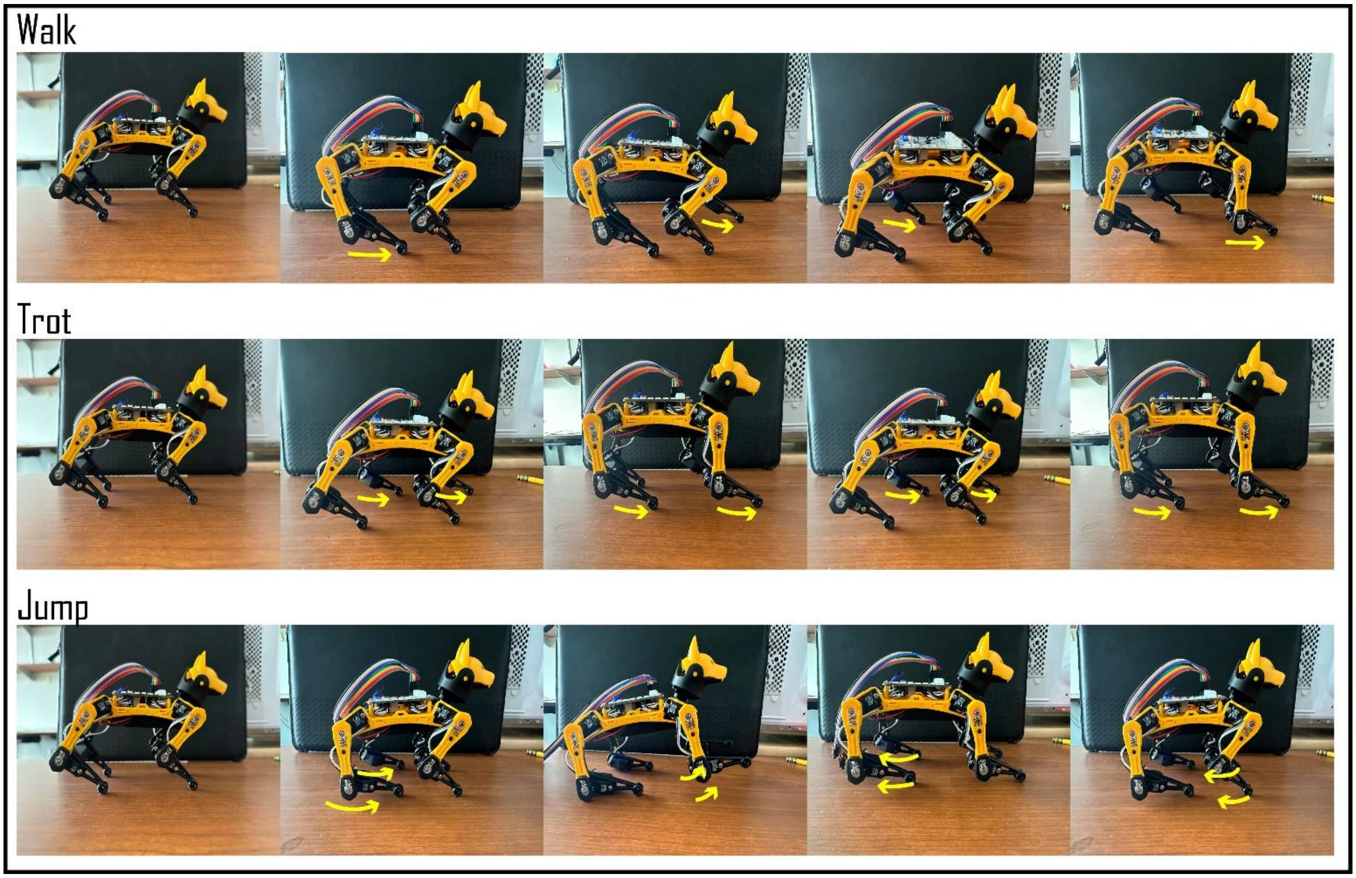

Fig. 5. Different modes of locomotion working on a Petoi Bittle robot with yellow arrows indicating the motion of limbs with respect to the previous(left) picture. These different modes were generated using different weight matrices and input stimulus whilst keeping the neuron parameters constant. Demo video with all three modes can be found at: https://youtu.be/_QFXf07VsEM/

all neurons. After this initial set of adjustment each neuron in the network is expected to fire in a particular order determined by the weight matrix. If any neuron does not fire, then we gradually increase the weight of that specific neuron till it exhibits expected spiking pattern.

IV. Results

The CPG system implemented on the Petoi bot does real-time emulation of a 4-neuron network with fully connected pre-trained synaptic weights between them. This rhythmic spiking due to external stimulus is then discretized from spike timings to events and these events are used to drive the motors in a fixed pattern determined by the weight matrix. Changing the weight matrix causes the robot to change the locomotion pattern. Fig. 5 shows the stop motion frames of the different modes of locomotion – walk, trot and jump, with the arrows showing which limb moved between consecutive frames. Demo video with all three modes can be found at: https://youtu.be/_QFXf07VsEM/

A. Benchmarking

For the evaluation of our proposed work, it is essential to compare the best available existing method to our method. We have compared our work with the widely used and accepted Leaky Integrate and Fire (LIF) Neuron running a similar network configuration to control robot locomotion and compared three parameters: memory used, average latency per time step and jitter tolerance to perturbations in input stimulus.

TABLE II. Comparitive Benchmarking of Central Pattern Generators Using Proposed Bio-Mimetic Neurons and LIF Neurons

Method / Motion Type		**Memory Used (in bytes and % total memory on Arduino)**	**Latency per time step (in ms) – Average over 500 steps**	**Jitter Tolerance to Input Stimulus Variation**
Proposed Non-linear Bio-mimetic Neuron	Jump	1133 (55.3%)	3.58	5%
	Walk	1137 (55.5%)	3.54	5%
	Trot	1137 (55.5%)	3.54	5%
Leaky Integrate and Fire Neuron	Jump	759 (37%)	0.98	<1%
	Walk	759 (37%)	0.98	<1%
	Trot	759 (37%)	0.98	<1%

Benchmarking results are summarized in Table 2. As expected because of its simpler dynamics, the LIF neuron takes 18% less memory and on average is 72% faster albeit this delay is in milliseconds which gives a net advantage of just over 2.5ms, is insignificant in the pipeline where the slowest task is the motor activation that takes around 200ms. Also this memory efficiency and speed gain for LIF neurons does not translate to jitter tolerance in the input which means that when this network is connected to a bigger network and has to work based on output of other similar networks, the LIF neuron will fail when the exact input is not matched whereas the proposed model can tolerate an input stimulus (I_{app}) jitter of 5% whilst still producing the expected output.

V. CONCLUSION

We have presented a novel method for controlling robotic locomotion with a Central Pattern Generator built using spiking behavior of non-linear bio-mimetic neurons, that can exhibit neuromodulatory behavior at nodal and network scales. The method involves the use of the non-linear neuron model presented in [7] to conceive a tightly coupled 4 neuron network which acts as the CPG for controlling the motor behavior of Petoi, a quadruped robot.

We have tuned the CPG network design for multiple modes of operation (Walk, Trot, Jump) which can be adjusted and observed real-time on-line by simply swapping a pre-trained weight matrix. The on-chip resources required to build this system have been highly optimized to realize real-time execution of the CPG network which requires the pre-trained weight matrices, and delay between action events to be stored in local memory on board for implementing the different modes.

The CPG using bio-mimetic non-linear neurons used marginally higher memory (378 bytes, 18% higher than LIF neurons), incurred insignificant latency of 3.54ms compared to motor activation delay of 200ms, while providing upto 5-10x higher jitter tolerance.

Our plan for future work is to use the proposed neuron models to operate in the bursting mode and generate a bursting CPG network that control the robot motion. Bursts in CPGs is highly common in biology leading to more granular control and robust loss-resilient operation, when compared to single spikes which can be lost or removed affecting the overall rhythm. Bursting CPGs will potentially enable the use of bio-mimetic control in advanced robots and for seamless human-machine interaction.

REFERENCES

[1] Dickinson, Patsy S. "Neuromodulation of central pattern generators in invertebrates and vertebrates." *Current opinion in neurobiology* 16, no. 6, pp. 604-614, 2006.

[2] Ijspeert, Auke Jan. "Central pattern generators for locomotion control in animals and robots: a review." *Neural networks* 21, no. 4, pp. 642-653, 2008.

[3] Kubendran, R., Wan, W., Joshi, S., Wong, H.S.P. and Cauwenberghs, G., "A 1.52 pJ/Spike Reconfigurable Multimodal Integrate-and-Fire Neuron Array Transceiver." In *International Conference on Neuromorphic Systems 2020*, pp. 1-4. 2020.

[4] Marder, Eve. "Neuromodulation of neuronal circuits: back to the future." *Neuron* 76, no. 1, pp. 1-11, 2012.

[5] Hodgkin, A.L., and Huxley, A.F., "A quantitative description of membrane current and its application to conduction and excitation in nerve." *The Journal of physiology,* Vol.117, no. 4, p. 500, 1952.

[6] Izhikevich, E.M., "Simple model of spiking neurons." *IEEE Transactions on* neural *networks* 14, no. 6, pp. 1569-1572, 2003.

[7] Ribar, R., and Sepulchre R., "Neuromodulation of Neuromorphic Circuits," in IEEE *Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 8, pp. 3028-3040, Aug. 2019, doi: 10.1109/TCSI.2019.2907113.

[8] A. S. Lele, Y. Fang, J. Ting and A. Raychowdhury, "Learning to Walk: Bio-Mimetic Hexapod Locomotion via Reinforcement-Based Spiking Central Pattern Generation," in *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 4, pp. 536-545, Dec. 2020, doi: 10.1109/JETCAS.2020.3033135.

[9] Rostro-Gonzalez, H et al. “Parameter estimation in spiking neural networks: a reverse-engineering approach.” *Journal of neural engineering,* vol. 9,2 (2012): 026024. doi:10.1088/1741-2560/9/2/026024.