An End-to-end Spiking Neural Network Platform for Edge Robotics: *From Event-Cameras to Central Pattern Generation*

Ashwin Lele Yan Fang Justin Ting Arijit Raychowdhury

***Abstract*—Learning to adapt one's gait with environmental changes plays an essential role in locomotion of legged robots which remains challenging for constrained computing resources and energy budget, as in the case of edge-robots. Recent advances in bio-inspired vision with dynamic vision sensors (DVS) and associated neuromorphic processing can provide promising solutions for end-to-end sensing, cognition and control tasks. However, such bio-mimetic closed-loop robotic systems based on event-based visual sensing and actuation in the form of spiking neural networks (SNN) have not been well explored. In this work, we program the weights of a bio-mimetic multi-gait central pattern generator (CPG) and couple it with DVS-based visual data processing to show a spike-only closed-loop robotic system for a prey-tracking scenario. We first propose a supervised learning rule based on stochastic weight updates to produce a multi-gait producing Spiking-CPG (SCPG) for hexapod robot locomotion. We then actuate the SCPG to seamlessly transition between the gaits for a nearest prey tracking task by incorporating SNN based visual processing for input event-data generated by the DVS. This for the first time, demonstrates the natural coupling of event data flow from event-camera through SNN and neuromorphic locomotion. Thus, we exploit bio-mimetic dynamics and energy advantages of spike-based processing for autonomous edge-robotics.**

I. Keywords:

Edge Intelligence, Spiking Neural Networks, Central Pattern Generation, Hexapods, Dynamic Vision Sensor (DVS) Cameras

II. Introduction

The central pattern generators (CPG) – neuronal circuits in the spinal cords form the basis of locomotion in many animals and cause the actuation of the muscles to generate rhythmic actions like breathing and walking [1]–[3]. These biological circuits cause contraction and relaxation of the muscles in a temporally correlated manner that actuates limbs in a specific order causing multiple gaits, which are observed in animal locomotion. Tasks like obstacle avoidance or prey approaching require locomotion in a particular direction at the desired speed. Thus, a seamless gait transition emerges naturally in such environmental situations. The information about the environment is gathered from sensory inputs like visual, olfactory, auditory or gyro systems and processed by the neural circuitry of different cortices in real-time. The CPG generating gait patterns can be modulated by these related cortex inputs and sensory feedback [3]. These complex biological closed-loop control systems that are composed of spiking neurons process the stream of event-based information in an energy-efficient way. Therefore, these systems serve as inspirations behind the models for the intelligent control of bio-mimetic edge-robots as shown in Fig. 1(a). This schematic shows sensing to actuation for a hexapod insect. The sensory information acquired is processed by the spiking neural circuitry in the brain to excite the motor neurons for locomotion. The rhythmic activity of motor neurons then causes rhythmic motion of muscles resulting in a gait.

A prototypical bio-mimetic electronic implementation of the spike-based locomotion platform is shown in Fig. 1(b). The legs of the hexapod robot are controlled by a network of spiking neurons. A typical network consists of six fully connected neurons where the spiking of a neuron causes the corresponding leg to move Fig. 1(c). Therefore, producing a gait corresponds to tuning the synaptic weights in the network to cause a specific sequence of spiking in the neurons. Execution of specific tasks requires transitioning between gaits to alter the speed and direction of motion. Thus, the objective is to produce multiple gaits on a single CPG each of which can be triggered independently with seamless transitions, depending upon the environment. Gonzalez et. al. used a set of simple linear equations to update the synaptic weights, thus demonstrating 3 different gaits [4] for forward motion. Multi-gait programming has been carried out in [5]. However, both these approaches use distinct synaptic matrices for different gaits requiring multiple CPGs for gait transitioning. Approaches like emulating the biological circuitry [6] or training gait patterns using remote supervision [7] have also been applied to SCPG learning. However, these methods are limited to learning a single gait and cannot effectively transition between gaits. [8] programs multiple gaits on a single CPG using a complex evolutionary learning method. Thus, a simple and generic weight adaptation algorithm for accommodating multiple gaits on a single SCPG is lacking and has been explored in this paper.

Sensory inputs describing the surrounding environment are necessary for the adaptation of gait for autonomous task completion. Neuromorphic Auditory Sensors (NAS) [9] and Dynamic Vision Sensors (DVS) [10] provide spike-based asynchronous inputs that are well-suited for spike-based locomotion in SCPG. NAS, when mounted on a robot for actuating locomotion, requires the generation of specific external sound

All authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, GA, USA.
E-mail: (alele9, yan.fang, jting31)@gatech.edu, arijit.raychowdhury@ece.gatech.edu

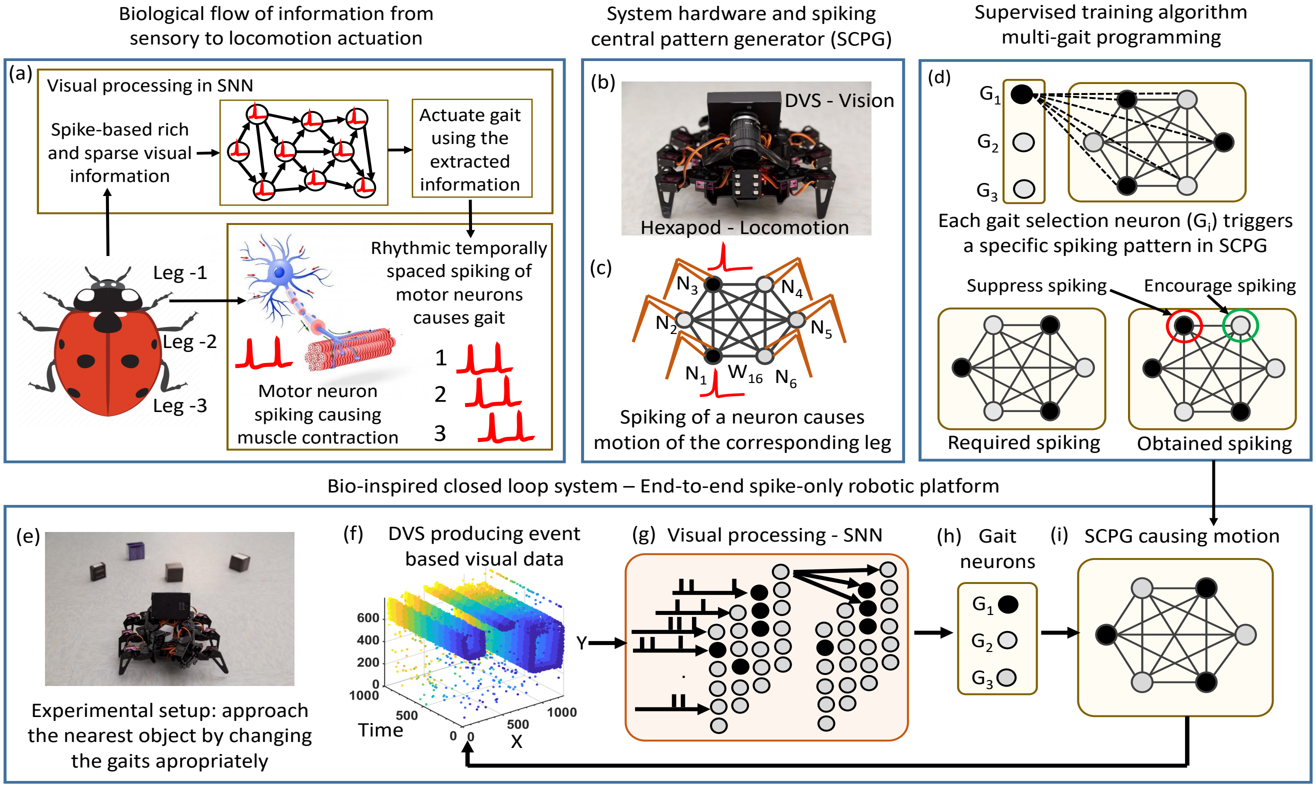

Biological flow of information from sensory to locomotion actuation
System hardware and spiking central pattern generator (SCPG)
Supervised training algorithm multi-gait programming
(a)
Visual processing in SNN
Spike-based rich and sparse visual information
Actuate gait using the extracted information
Leg -1
Leg -2
Leg -3
Rhythmic temporally spaced spiking of motor neurons causes gait
Motor neuron spiking causing muscle contraction
1
2
3
(b)
DVS - Vision
Hexapod - Locomotion
(c)
N_3
N_4
N_2
N_5
N_1
W_{16}
N_6
Spiking of a neuron causes motion of the corresponding leg
(d)
G_1
G_2
G_3
Each gait selection neuron (G_i) triggers a specific spiking pattern in SCPG
Suppress spiking
Encourage spiking
Required spiking
Obtained spiking
Bio-inspired closed loop system – End-to-end spike-only robotic platform
(e)
Experimental setup: approach the nearest object by changing the gaits apropriately
(f) DVS producing event based visual data
600
400
200
0
1000
500
0
Time
0
500
1000
X
Y
(g) Visual processing - SNN
(h) Gait neurons
G_1
G_2
G_3
(i) SCPG causing motion

Fig. 1. (a) Schematic of biological sensing to actuation in a hexapod insect. Information acquired by eyes in the form of neuronal spikes is processed by the SNN to trigger the motor neurons. The motor neurons trigger the muscle contraction causing motion. When the motor neurons driving legs 1,2,3 fire with fixed temporal differences as shown this gives a rhythmic motion of legs causing a gait (b) Hardware of the bio-inspired system proposed. The Dynamic Vision Sensor (DVS) mounted on the hexapod robot (c) Schematic of the fully connected spiking central pattern generator (SCPG). All neurons are connected to each other. The leg corresponding to a neuron moves when that neuron fires. (d) Multi-gait SCPG. Three gait selection neurons corresponding to the three gaits. Each G_i elicits a different spiking pattern corresponding to the motion of the legs required in that gait. The weights are updated using the difference between required and obtained spiking of CPG neurons which determines whether to excite or suppress the spiking of that neuron (e) Experimental setup for testing the end-to-end system. The hexapod has to autonomously identify the closest prey and approach it by changing the gaits. The motion of hexapod in the environment shown causes DVS to generate a flow of events in the frame as shown (f). The events are fed into a spiking neural network (g) detecting the edges of the objects and identifying the closest object. Visual data processing network triggers one of the three gait selection neurons to trigger the SCPG (h) Gait selection neurons excite the (i) SCPG which causes hexapod to move with that gait to generate visual data in form of events for the next step.

frequencies as inputs. This requires an external agent to steer the movement making it non-autonomous. On the other hand, DVS generates an asynchronous input event when the intensity of the pixel changes in the field [10]. This happens naturally during locomotion causing sensory input generation as a result of motion allowing potential autonomy in the actions. Considering the efficiency of energy consumption and data amount, we notice that the dynamic vision sensor (DVS) camera can be a promising candidate for our task of sensory input-driven autonomous locomotion.

In a DVS camera, all the pixels operate independently and therefore energy is spent in generating events only in the part of the image where the intensity has changed. This decoupling from a frame-based approach causes power levels as low as 10 mW. From the perspective of visual information processing, the DVS camera tremendously reduces the amount of data in a video stream and thus provides more energy efficiency. Another two merits of DVS are low latency and high dynamic range. DVS cameras have microsecond level latencies, which enable the visual processing for high-speed motion. Another advantage presented by these sensors is the high dynamic range of the order of 130 dB v.s. 60 dB of standard ones which is 7 orders of magnitude larger [11]. All these advantages coupled with matching event data for SNNs make DVS perfectly suited for our application. The locomotion platform with DVS mounted on it is shown in Fig. 1(b).

DVS camera-based end-to-end robotic tasks like obstacle avoidance [12]–[14] and object tracking [15]–[17] have been explored on different platforms and scenarios. A model-based event clustering method is implemented for a robotic goalie [17]. Deep learning has also been applied to the same problem [13]. Spiking implementation of such a visual processing system is explored by Blum et. al. showing target acquisition and obstacle avoidance in a wheeled robot [18]. A spiking neural network for looming object detection [19] has also been demonstrated recently. DVS-based vision coupled with spike-based path-planning with Pulse Frequency Modulation

(PFM) based actuation has also been shown [20]. However, all these approaches have shortcomings in either of the two critical aspects. First, the approaches involving non-spiking algorithms fail to take advantage of sparse asynchronous input data specifically suited for spiking neural networks and bio-plausibility. Secondly, the works involving spiking implementations of visual systems do not extend them to locomotion platforms [19], [21]. This means that the locomotion is not spike-based as in the SCPG and the flow of events terminates at the visual processing stage. Thus, a full exploration of the end-to-end robotic system allowing a flow of events throughout the network as shown in Fig. 1(e-i) has not been attempted.

In this work, we propose a supervision based weight adaptation methodology for programming multiple gaits in a single SCPG (Fig. 1(d)). We then demonstrate a closed-loop, fully spiking, a bio-plausible robotic system involving only SNN, processing of event data from DVS to actuation of a gait as shown in Fig. 1(e-i). Our simple edge detecting SNN separates multiple objects from each other (Fig. 1(e)), allowing online decision making in gait selection. The key contributions of this work are:

- A light-weight algorithm for programming spiking patterns in recurrently connected SCPG for multi-gait locomotion.
- Demonstration of the first closed-loop end-to-end robotic system with spiking neural network-driven processing of data from visual sensory data acquisition to locomotion.

The work opens up avenues for smart bio-mimetic robotics for advanced applications.

III. Locomotion via Multi-Gait Spiking-CPG

A. The Network Structure of the Central Pattern Generator

The SCPG network is shown in Fig. 2(a). It is composed of six fully connected neurons that generate gait pattern with spikes. Each neuron corresponds to one leg and spikes to activate the servo motor of the corresponding leg that finally actuates the leg. A sequence of spikes generated by the SCPG network can drive the locomotion of the hexapod robot (Fig. 2(b)). Such a sequence determines the gait pattern that controls the speed and direction of locomotion. The CPG neurons are driven by input neurons called gait selection neurons. We use three gait selection neurons to enable three different gaits. When a gait pattern is desired, one gait selection neuron stimulates a subset of CPG neurons to fire in a particular sequence and enable a designed locomotion gait. All the CPG neurons are connected to gait selection neurons with excitatory synapses. To emulate the dynamics of each neuron, we use a discretized leaky integrate and fire (LIF) model with membrane potential (V_j) expressed as [4].

$$V_j[t+1] = \frac{V_j[t]}{\alpha} + \sum_i W_{ij} S_i[t] \tag{1}$$

$$if\ V_j[t] > V_{th} \text{ then } S_j[t+1] = 1, V_j[t+1] = 0 \tag{2}$$

The leaking current is modelled with a decay factor α. When the membrane potential exceeds the spiking threshold V_{th}, a spike is fired and the membrane potential is instantly reset to the resting potential, which is zero (Equation (2)). A pre-synaptic spike, S_i results in the increment of the membrane potential of the post-synaptic neurons. The synaptic weights (W_{ij}) scale the inputs from the presynaptic neurons (i^{th} neuron) to the post-synaptic neuron (j^{th} neuron) as shown in equation (1). Here we do not introduce the synaptic dynamics and delay, but incorporate the effect in the dynamical equation of the neuron. Therefore, the membrane voltage increase caused by the pre-synaptic spike occurs in the immediate next cycle. These simple discretized dynamics are voltage-based and easy to emulate on a digital hardware platform. Such a method bypasses the computation of membrane current and reduces the power consumption as well as ensures low latency of computation. The CPG is implemented on a Raspberry Pi 3 B+ single-board computer, mounted on an Adeept Raspclaws Hexapod (Fig. 1(b)).

B. Weight Adaptation Mechanism for the SCPG

The CPG neurons need to fire spikes in a specific order to make the robot walk with a particular gait. Such a firing sequence is determined by the gait selection neurons (G_i) which trigger the network. Fig. 2(b) shows different spiking patterns that correspond to different gaits. E. g. when G_1 spikes to initiate forward motion using tripod gait, $N_{2,4,6}$ fire in the first time step and $N_{1,3,5}$ spike in the next time step. Thus, the training problem is to program the weight matrices so that when a gait selection neuron fires, the desired sets of CPG neurons fire in the next few time steps to generate the gait pattern.

We use a supervised weight update algorithm to program the synaptic weights in the network. The pseudo-code for the algorithm is described in algorithm 1 and illustrated in Fig. 3. To start, all the weights are initialized randomly. Fig. 3 shows a time instance when G_1 spikes which triggers the tripod gait that enables forward motion. The required spiking of the CPG neurons is shown in Fig. 3(a) with time steps shown on the left. In the beginning, the obtained and the required spiking patterns are not identical. For example, the obtained spiking at a particular time instance as the algorithm runs is shown in Fig. 3(b). The algorithm tries to make the required and obtained spiking patterns identical. If the obtained and required spike patterns of the SCPG are the same then the neuron is firing correctly, thus the weights need no further modification. However, for neurons spiking redundantly, the firing behaviour needs to be suppressed. This is marked by red circles in Fig. 3(b). Similarly, for a neuron not spiking when required, the spiking should be triggered as marked in the green circle. The total number of erroneous firings at every time step is called error as plotted in Fig. 3(d).

The weight updates required for learning the correct spike patterns are shown in Fig. 3(c). To suppress redundant spiking, the incoming weights to that neuron are decreased. However, not all incoming weights are altered and only the synapses

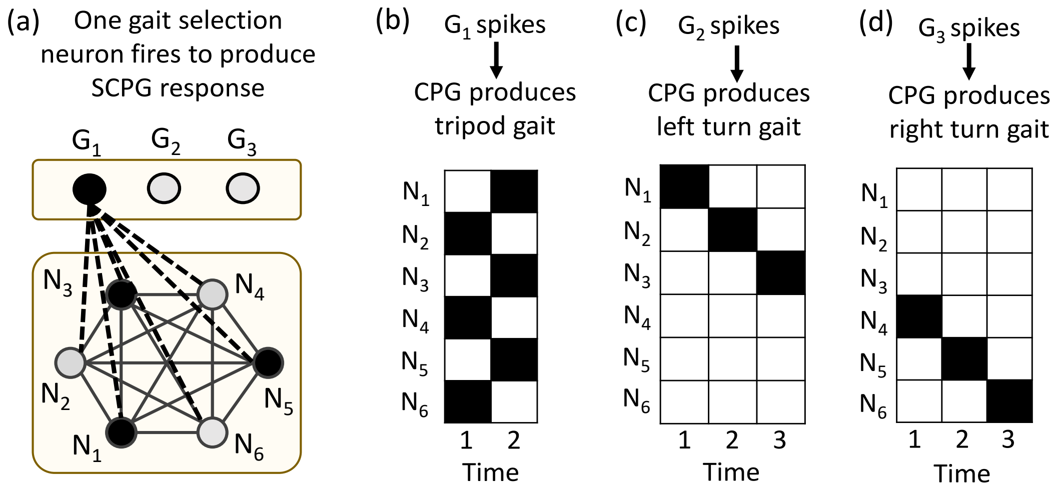


Fig. 2. (a) Schematic for SCPG and gait selection neurons. Firing of each gait selection neuron is expected to cause a specific sequence of firing in the SCPG neurons. Spiking of CPG for (b) tripod (forward), left turn and right turn gaits occurs for spiking of G_1,G_2,G_3 respectively. Alternate neurons fire in tripod gait. For turning left, neurons of the right side of the hexapod fire and vice-versa

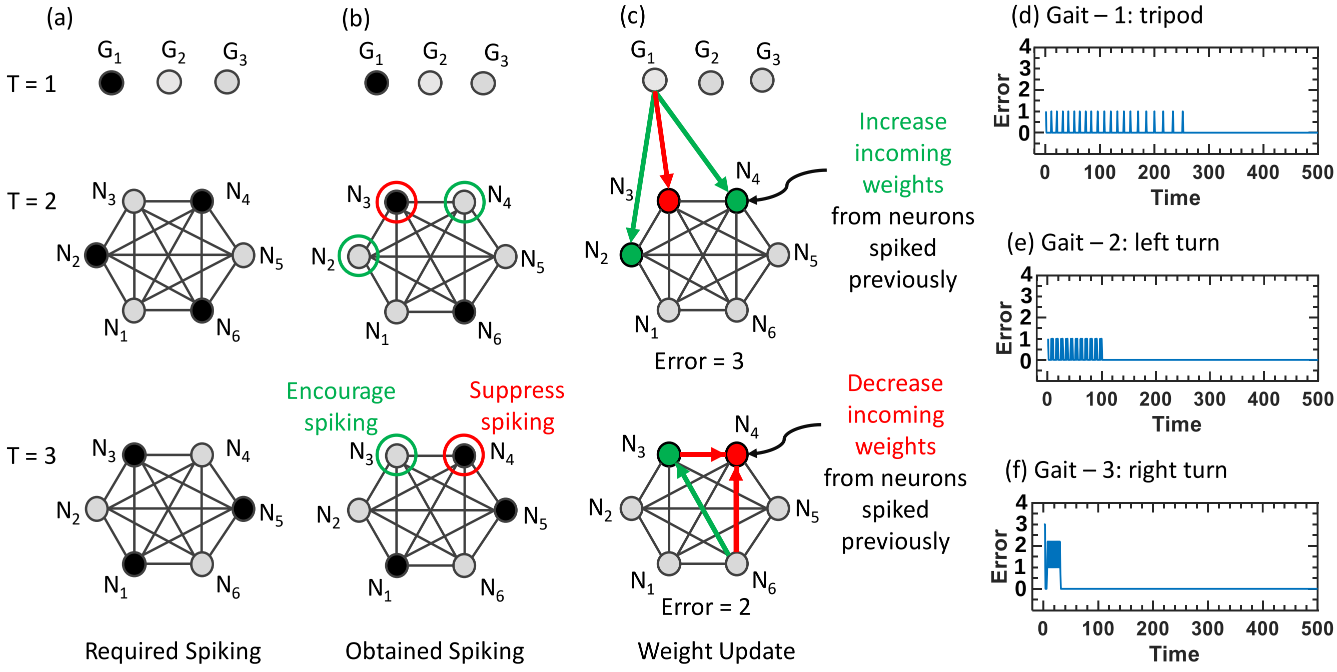


Fig. 3. Programming the tripod gait on SCPG. Spiking of gait selection neuron G_1 should invoke a spike pattern causing tripod gait for forward motion. G_1 fires at T = 1 and the required SCPG neuron firing pattern for the next two time instances as shown in part (a). (b) The obtained spiking sequence for neurons in these two time instances. To align with the required spiking pattern, the neurons spiking unnecessarily (red) should be suppressed and not spiking as required are to be excited (c) This is done using the potentiating (depressing) synaptic weights from the neuron last spiked as shown. The change in synaptic weight is stochastic at every step. The total number of neurons showing deviation from required spiking is shown as the error. (d-f) Total error for each gait as the algorithm progresses. The error starts high, oscillates and goes down to zero showing the convergence of the algorithm

connected to the neurons that have spiked previously are updated as shown. This procedure of stochastic updates is repeated for every gait. Fig. 3(d-f) shows the total error observed in the spiking process at each time step for each gait. For all gaits, with the random initialization, the error oscillates and finally converges to zero indicating correct programming of the gait pattern in the SCPG. Thus, our method allows programming of multiple complex recurrent connections on a single compute fabric with local error minimization using supervision. Stochasticity ensures that the weight updates do not get stuck in local minima.

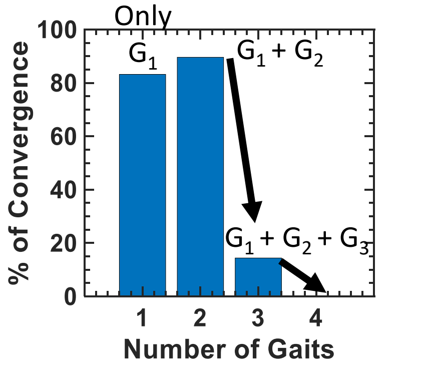
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3097675, IEEE Transactions on Cognitive and Developmental Systems

Algorithm 1 Supervised Weight Adaptation for the Spiking Central Pattern Generator

```
Initialize weights randomly $W_{in}, W_{CPG}$
Initialize CPG neuron membrane potentials, $V_{CPG}[0] = 0$

for $gait = 1$ to 3 do
    Initialize Spikes $G_i = 1, S_{CPG} = 0$
    for $time = 1$ to $T_{gait}$ do
        for $neuron = 1$ to 6 do
            $I = W_{in}S_{in} + W_{CPG}S_{CPG}$
            $V_{neuron}[t] = V_{neuron}[t-1]/\alpha + I$
            $S_{CPG-prev}[t] = S_{CPG}$
            if ($V_{neuron}[t] > V_{Thresh}$) then
                $S_{CPG} = 1$, $V_{neuron}[t] = 0$
            end if
            $Error = S_{CPG}$ - $S_{CPG-requied}$
            $\Delta W_{in} = Error \times random()$
            $\Delta W_{CPG} = Error \times S_{CPG-prev} \times random()$
        end for
    end for
end for
```

C. Convergence Analysis for Multi-Gait SCPG



Maximum 3 gaits can be programmed using the algorithm

Fig. 4. Percentage of instances converging to correct gait versus the number of gaits being programmed to the SCPG. Up to two gaits can be programmed with high accuracy which drops when three gaits are being programmed simultaneously

In this section, we analyze the capacity of the algorithm to store multiple gaits on a single CPG. Due to the stochasticity of the weight adaptation algorithm, not all simulation trials achieve convergence. Also, as the number of gaits increases, the percentage of correct convergences is expected to drop. Thus, we choose a different number of gaits to be programmed and run 100 iterations for each. Then we identify the number of instances of all gaits being correctly learnt to obtain the percentage of cases where the system convergences correctly. Fig. 4 shows the percentage of successes for a varying number of total gaits that need to be learnt. We observe that a maximum of two gaits can be learnt with a high convergence rate of ($> 80\%$). The convergence rate steeply drops when 3 gaits are to be simultaneously learnt to ($\sim 20\%$). More than three unique gaits cannot be simultaneously learnt in the current algorithm.

IV. DVS Data Acquisition and Visual Information Processing

In this section, we describe our method of using SNN to process the event stream of visual data generated from a DVS camera. We mount a CelexV DVS camera on the hexapod robot for event-flow generation [10] (Fig. 1(b)). This DVS camera has a dense frame resolution of 800 × 1280 pixels, streaming event data through MIPI interface. Cypress CX3 board within the camera converts MIPI to USB 3.0 for acquisition in the laptop over USB interface. The events are processed using packages provided in C++. The event stream is fed into the SNN network which processes it to identify the presence of the prey. The timestamps of the events are used to encode voltage leakage in emulating LIF behaviour. As the hexapod robot walks, visual events are generated with changing intensities of the pixels in the frame. Fig. 5(a) shows the block diagram of the spiking neural network that detects the presence of objects for locomotion control. The event accumulated frame at every stage is shown in Fig. 5(b-e).

The DVS generated events are acquired in the DVS layer. This noisy input is shown in Fig. 5(b). The input is filtered using nearest neighbour filtering keeping only spatially and temporally close events. This results in only the sharp edges of the detected objects surviving as shown in Fig. 5(c). The frame is now divided into multiple windows of size 100 × 100 (Fig. 5(c)) to identify the window in which the object is present. The next two layers (layer 2,3) identify the vertical and horizontal edges of the object, thus localizing the object within the frame. Layer - 3 connects to the gait selection layer driving locomotion. Fig. 5(d) shows the detected edges at the end of layer -3. These edges indicate the presence of the object in the upper half of the frame. Thus, the objects are still far from the robot and the robot need to move forward. This is done by triggering G_1 to spike which generates a tripod motion in the CPG (Fig. 5(f)). Details of the synaptic connections in each layer are explained in the next section using the miniaturized model of visual processing.

A. Noise Removal via Filtering

The first layer in visual processing filters the noise in DVS data. When the DVS moves, edges of the objects create events in continuous regions as observable in Fig. 5(b). The events in these continuous regions and close in time correspond to the edges while other sparse events are to be filtered out. The nearest neighbour filter implementing this is shown in Fig. 6(a). Each neuron spiking in the DVS layer is connected to nine neurons in the filtering layer with synaptic weights shown by the nearest neighbour filter. Thus, with every incoming event from DVS, nine neurons in the filtering layer have an increase in the membrane potential. The neuron

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3097675, IEEE Transactions on Cognitive and Developmental Systems

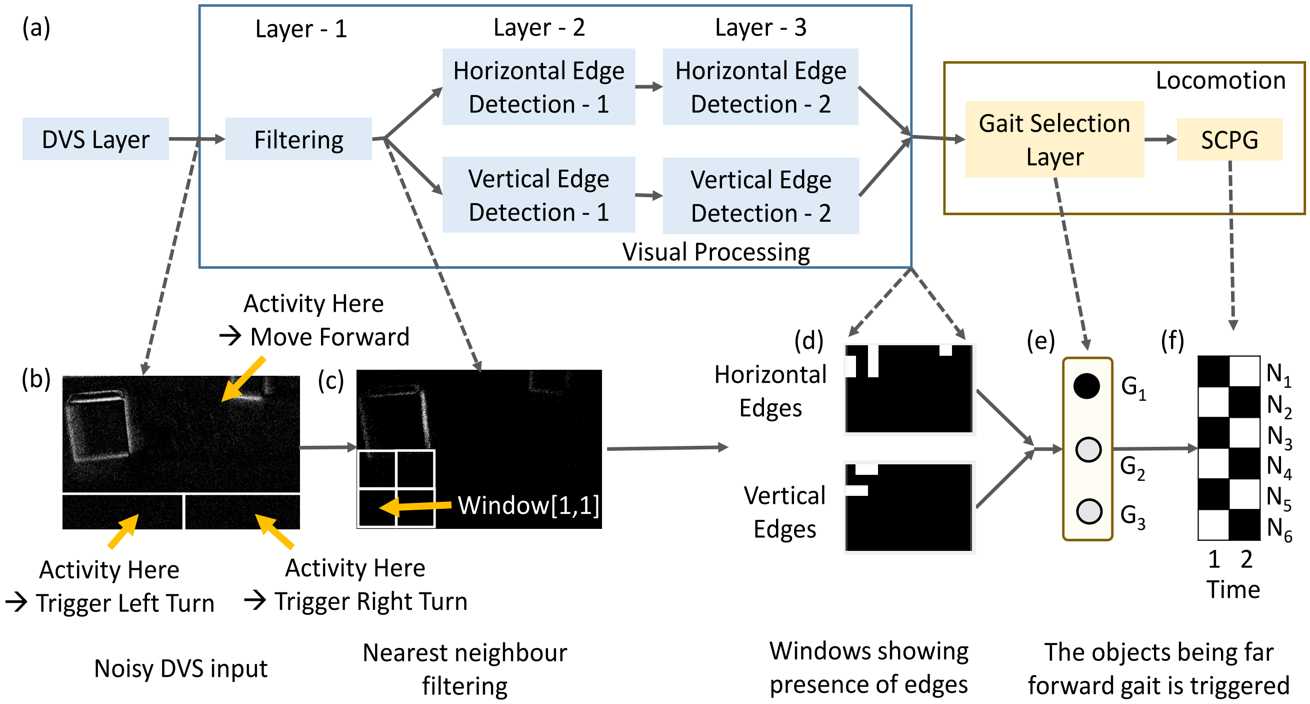

(a)
Layer - 1
Layer - 2
Layer - 3
Horizontal Edge Detection - 1
Horizontal Edge Detection - 2
DVS Layer
Filtering
Vertical Edge Detection - 1
Vertical Edge Detection - 2
Visual Processing
Locomotion
Gait Selection Layer
SCPG
Activity Here → Move Forward
(b)
(c)
Window[1,1]
Activity Here → Trigger Left Turn
Activity Here → Trigger Right Turn
(d)
Horizontal Edges
Vertical Edges
(e)
G_1
G_2
G_3
(f)
N_1
N_2
N_3
N_4
N_5
N_6
1 2
Time
Noisy DVS input
Nearest neighbour filtering
Windows showing presence of edges
The objects being far forward gait is triggered

Fig. 5. (a) Block diagram of the SNN for visual processing. DVS layer acquires the data from the SNN. This noisy data is filtered using nearest neighbour filtering in layer - 1. Filtered data is passed to two layers of neurons detecting horizontal and vertical edge as shown. Detection takes place in two stages in layer - 2 and 3. Depending upon where in the frame the object is detected, different gait selection neuron is triggered to actuate locomotion using SCPG (b) Data representation at different stages of processing. Noisy event information is acquired from the DVS as shown (c) Filtered events. Filtering reduces the noise pixels significantly and only continuous object edges remain. (d) Edges are detected in the upper region of the frame (e) This triggers the forward gait selection neuron G_1 resulting in (f) tripod gait of SCPG

exactly opposite to the DVS layer neuron has the highest rise in the potential followed by the nearest neighbours and next-nearest neighbours. If the same or neighbouring neuron in the DVS layer fires immediately after this, it boosts the potential of the neuron in the filtering layer further causing it to fire. If no neuron in the neighbourhood fires, the potential in the filtering layer leaks out according to the leaky behaviour of the LIF neuron. Thus, only temporally and spatially close spiking activity results in a spike in the filtering layer. The other spikes get filtered out by the leaky behaviour of spiking neurons. The synaptic weights are chosen such that a single spike does not cause spikes in the filtering layer. This results in only continuous edges surviving as seen in Fig. 5(b). Thus, we exploit the leakiness of LIF neurons to cancel out high noise in DVS events. This layer feeds the edge detection layer.

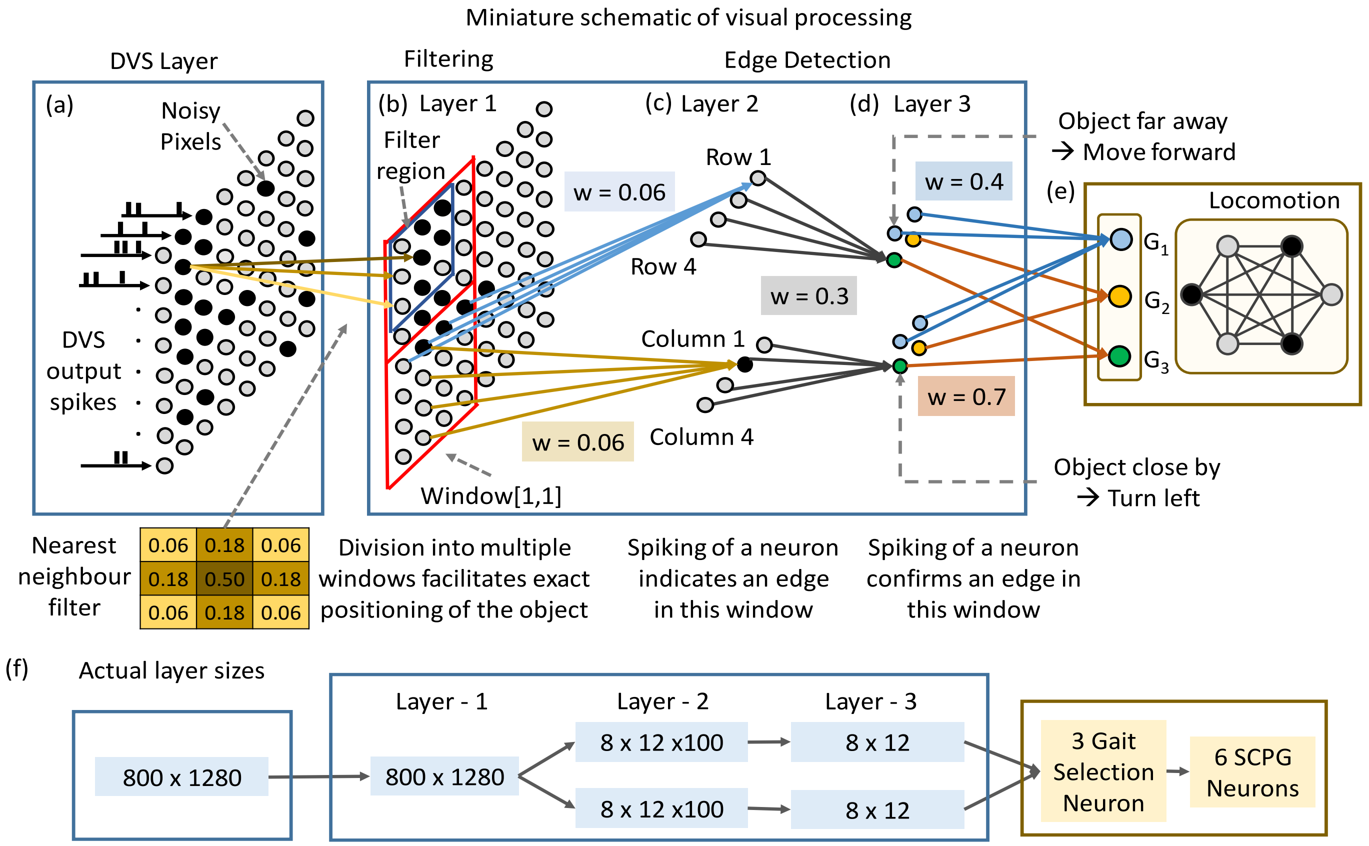
B. Edge Detection for Object Identification

For edge detection, the neurons in the noise filtered layer are segregated in windows as shown in Fig. 6(b). The mini model shows 8 × 8 frame with 4 windows of dimensions 4 × 4. Two such windows are shown bound by the red rectangle in Fig. 6(b).

The edge detection is a two-step process where layer - 2 indicates the presence of edge and layer - 3 confirms it. For vertical edge detection, neurons in a vertical column of filtering layer are connected to a neuron in layer - 2 for each window. Column - 1 to column - 4 each correspond to one column in window [1,1] as shown in Fig. 6(c). Significant spiking activity in a column indicates the presence of an edge in it, making the column$_i$ neuron in layer - 2 to spike. Column neurons in layer - 2 feed the next layer. All neurons corresponding to one window are connected to one neuron in layer -3. Significant spiking activity of layer - 2 neurons triggers the layer - 3 neurons confirming the presence of a vertical edge in that window. Similarly, an entire parallel branch for horizontal edge detection is present in layer - 2 and layer - 3. The synapses connecting the filtering layer to layer - 2 have the uniform synaptic weights making layer - 2 a leaky summation unit. The connection between layer - 2 and layer - 3 are also the same with the only change being the magnitude of the weight.

The synaptic weights are chosen such that only when a critical number of neurons are spiking, the neuron in the next layer is fired. This is done using 1 volt as the spiking threshold. Thus, the weights connecting the DVS layer to the filtering layer have weights 0.5,0.18,0.06 as shown in Fig. 6(a). This makes sure that only when the neurons in a continuous spatial region (8 × 8) are spiking closely, the filtering neuron spikes. Similarly, the weights of connection between layer - 2 and layer - 3 are chosen to be 0.06, so that at least 16 neurons

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3097675, IEEE Transactions on Cognitive and Developmental Systems



0.06	0.18	0.06
0.18	0.50	0.18
0.06	0.18	0.06

Fig. 6. Miniaturized schematic of SNN processing the visual data. The noisy output of the DVS in the form of a stream of events is generated when the robot walks (a) 8 × 8 pixels receive this data from DVS (b) Nearest neighbour filtering filters out pixels that are not accompanied by their neighbours. As observable, only continuous edges survive. The frame is divided into multiple windows (c) Window-wise identification of vertical and horizontal edges. Parallel networks run for horizontal and vertical edges. All neurons in a row (column) within a window are connected to one neuron in layer - 2. The firing of layer - 2 neurons indicates the presence of an edge. (d) All neurons corresponding to a window are connected to a single edge detection neuron per window. The spiking of these neurons confirms the presence of a vertical (horizontal) edge. These neurons drive the gait selection neurons for locomotion (e) Colour-coded connections of layer - 3 neurons with gait selection neurons. If objects are in the upper half, the robot needs to walks forward. Therefore, the neurons of layer - 3 in the upper region connect to G_1. Similarly, for approaching the object detected in the lower region, these neurons are connected to the gait selection neurons G_2, G_3 (e) SCPG for locomotion (f) Actual layer sizes of the SNN for DVS. The synaptic weight at each layer is shown by w with matching colour

in the column need to spike for it to be detected as a vertical edge. Layer - 2 and Layer - 3 are connected by the synapses of weight 0.3 requiring at least 4 neurons to spike for the presence of an edge to be confirmed in layer - 3.

Fig. 6 miniaturizes the setup. The real sizes of the layers in the system are shown in Fig. 6(f). 800 × 1280 neurons filtering layer feeds layer - 2. Layer - 1 consists of 8 × 12 windows each having 100 columns (rows) corresponding to the window size of 100 × 100. The rightmost 80 neurons in every row are ignored while division into windows. 100 neurons in each column within a window feed the corresponding layer - 2 neurons. Thus, 100 columns in every window make the size of layer - 2 to be 8 × 12 × 100. Each neuron in layer - 3 corresponds to a single window. Spiking of that neuron suggests the presence of a horizontal (vertical) edge in that window.

C. Gait Selection

The presence of an object in the upper windows indicates that the object is far from the robot. In this case, the robot does not have to align with the object immediately and thus chooses to move forward. On the other hand, when a neuron corresponding to a window belonging to the lower region of the frame spikes in layer - 3, this indicates the presence of an object in close vicinity. This requires the triggering of a turning gait for aligning the robot with the object. This is implemented by connecting the neurons corresponding to the lower region to the left (G_2) and right (G_3) turn triggering neurons as shown by colour-coding. The neurons in the upper window connect to the forward gait selection neuron (G_1).

The weights connecting to $G_{2,3}$ are kept higher ($w = 0.7$) compared to that of G_1 ($w = 0.4$) to give preference to closer objects. This makes the robot to get steered towards the nearest prey when there are multiple objects in the field of view. The rate-coded winner-take-all mechanism is applied in gait selection neurons. This means that the gait selection neuron spiking with the highest frequency is preferred and that gait is activated. Locomotion in the environment generates a new set of events for every step and the flow of events through the SNN continues.

V. Results

The system is deployed in an indoor lab environment and tested in a simulated predatory scenario shown in Fig. 1(e). The hexapod robot patrols in the environment with multiple objects and approaches the nearest object as the target. Visual processing of the DVS data is carried out on laptop and is communicated to the SCPG on the robot through wifi. During the process, the robot adapts its gait based on different circumstances by transitioning between three gaits.

A. Predator-Prey Scenario with a Single Prey

We test the system in our first task, namely approaching a single target object (prey) placed in the front of the robot. The robot needs to approach and align itself directly in front of it. A demonstration video is available online (video - 1). In this demo, DVS data is processed and actuation signals are sent to the hexapod in an end-to-end closed-loop brain-inspired system. This acts as a proof-of-concept design that shows the feasibility of engineering bio-mimetic spiking systems. Screenshots of the video along with the visual information processed in the background is shown in Fig. 7. Part (a) of the figure shows the block diagram of the end-to-end system. In the first step, the object is far from the robot. The event stream generated by the DVS in one step of the hexapod robot is shown in Fig. 7(b). After the events pass through the filtering stage, the object boundaries become clearly discrete as in part (c). The activity in the upper region of the frame indicates the object is far which is inferred by the edge detection layers. Part (d) shows the 96 neurons corresponding to layer - 3 driving the gait triggering neurons coloured in part (a). Neurons 1 to 60 drive the forward gait selection neuron, 61 to 78 drive right and 79 to 96 drive the left turning gait as shown. The activity is mainly observed forward motion triggering neurons as the object is far away. This activation of tripod gait results in the forward motion of hexapod as shown in part (e). For illustration, we have shown edge detection only for the vertical edges (Fig. 7). The system also includes horizontal edge detection and the spiking patterns show similar behaviour.

In the second step, the robot has approached the object and moved close to it. The same set of figures for this configuration indicate the DVS outputs and the corresponding filtered boundaries. As an edge lies in the lower right corner, this triggers the right turn (G_2) so that the robot can align with the object. This is observed in part (h) where spiking is observed in the neurons corresponding to the right turn. As the weights to (G_2) are higher, as explained in the previous section, a right turn is triggered. The hexapod now turns right to align itself with the object. This demonstration validates the working principle of the proposed end-to-end SNN architecture.

B. Predator-Prey Scenario with Multiple Preys

We repeat the experiment in a more complex scenario where multiple objects (preys) are placed in front of the robot. The task becomes more difficult as the robot is now required to consider only the closest object. video - 2 shows the demonstration. Screenshots of the video are shown in Fig 8. The objects are named as shown. The initial position of objects and the robot is shown in Fig. 8(a).

The robot starts moving forward and takes three forward steps until object - 1 is close to it (Fig. 8(b)). Now object 1 being in its proximity, it senses the activity in the lower right corner of the frame. This makes it change the gait to turn towards the right for two consecutive steps. The robot can be seen to have aligned itself with object 1 (Fig. 8(c)). Now object 1 is removed to test if the robot can decide between other objects in front of it. Object 2 and object 4 are in its field of view at a distance where object 2 is closer. The robot takes 2 steps forward to be in a position in Fig. 8(d). With object 2 lying on its right, it generates activity in the lower right region of the DVS frame. Thus, the robot now turns right to follow the object - 2. This step aligns it with object - 2 as required (Fig. 8(e)). The trajectory of the robot is shown in Fig. 8(e). This experiment confirms the capability of the platform to identify the nearest object and trigger appropriate gait depending upon that. The demonstration confirms the potential in fully-spiking systems for connecting spiking interfaces like DVS and CPG.

C. Energy Estimation for End-to-end System

We provide an approximate estimation of the energy consumed by different parts of the system. Spiking neural networks both in CPG and visual processing can be implemented on an SNN ASIC platform. We calculate the energy consumption of the proposed system assuming it is implemented on Loihi, the neuromorphic platform by Intel. Each spiking consumes 1.7 nJ of energy [22]. We record the number of events generated in taking 1 step forward. The results show that the filtering layer generated $\sim$1.5 million events, consuming 2.55 mJ energy. Gait generation in CPG makes 7 neurons spike consuming only 11.9 nJ. This results in $\sim$ 2.55 mJ of energy consumption. The overall sub-mJ energy consumption highlights the potential of such platforms for battery-powered edge applications, where event-based processing and spiking neural network-based decision making and actuation can reduce the overall system power.

VI. Discussion and Benchmarking

We compare our work with previous efforts in building autonomous robots using spiking CPGs in Table - 1. [4]–[8] are focused on designing training algorithms for SCPG of hexapod robots. The trained locomotion is inflexible and leaving the agents difficult to be steered in dynamically

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCDS.2021.3097675, IEEE Transactions on Cognitive and Developmental Systems

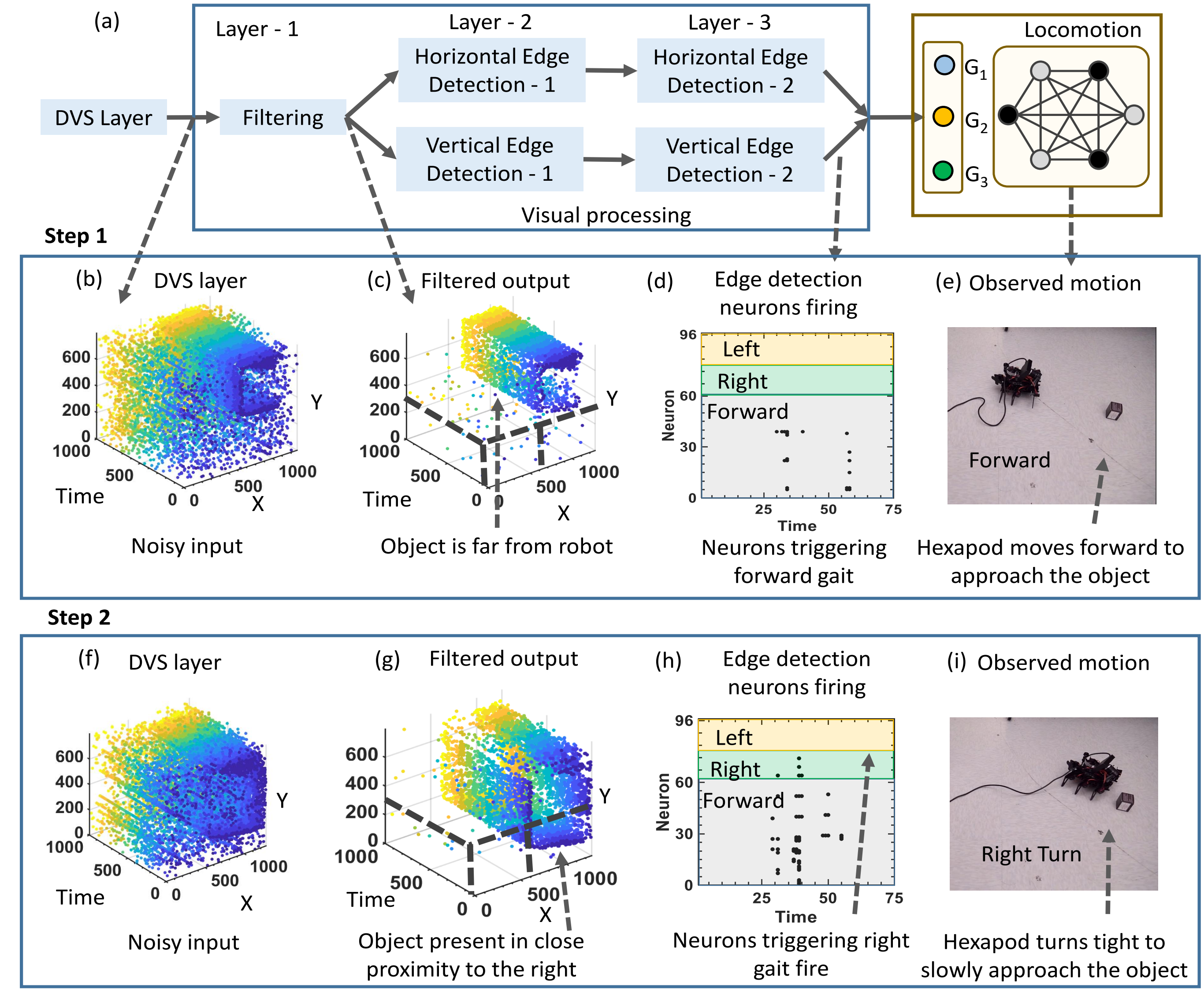

(a)
Layer - 1
Layer - 2
Layer - 3
Locomotion
DVS Layer
Filtering
Horizontal Edge Detection - 1
Horizontal Edge Detection - 2
Vertical Edge Detection - 1
Vertical Edge Detection - 2
Visual processing
G1
G2
G3
Step 1
(b) DVS layer
Noisy input
(c) Filtered output
Object is far from robot
(d) Edge detection neurons firing
Left
Right
Forward
Neuron
Time
Neurons triggering forward gait
(e) Observed motion
Forward
Hexapod moves forward to approach the object
Step 2
(f) DVS layer
Noisy input
(g) Filtered output
Object present in close proximity to the right
(h) Edge detection neurons firing
Left
Right
Forward
Neuron
Time
Neurons triggering right gait fire
(i) Observed motion
Right Turn
Hexapod turns tight to slowly approach the object
Time
X
Y

Fig. 7. (a) Block diagram of the system. (b) Event stream acquired by the DVS during one step of the hexapod walk (c) Contiguous edges can be seen to survive occupying the upper region in the frame after filtering (d) Raster plot for neuronal firing in layer - 3. Colour coding shows the set of neurons driving the gait selection neurons shown in (a). The object being far results in triggering of forward gait (e) Hexapod approaches the object (f) Noisy event stream generated during the previous step (g) Filtered event stream shows the presence of an edge in the lower right corner of the frame (h) Triggering of edge detection neurons driving the right turning gait (G_2) (i) Robot aligns itself with the object

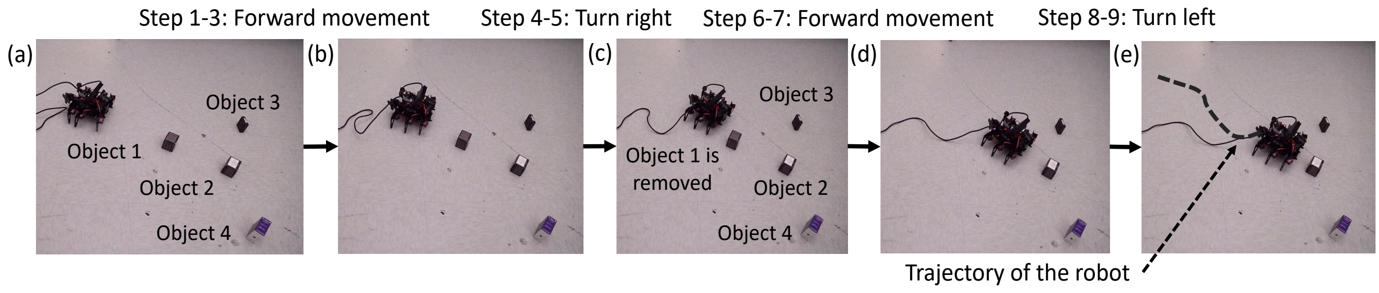


Fig. 8. Step wise evolution of the robotic locomotion (a) Initial setup (b) Robot takes 3 forward steps to approach the nearest object (object 1) (c) Right turn gait is activated and robot turns right to align itself with object 1 (d) Object 1 is removed and robot approaches the next nearest object (object 2) taking 2 forward steps (e) Spiking activity if edge detection neurons triggers left turning gait to align the robot with object 2

changing targets and environments. The gait transition is also not possible online because no sensory inputs are integrated into the control. [23] and [24] use non-bio-inspired sensory input closing the loop allowing autonomous learning. [21], [19], [20] take additional step towards bio-mimicking by using event-based sensors. However, the locomotion platforms are not event-based. [25] closes the fully SNN control loop at the cost of external agent-based steering, restricting the autonomy. Our work adds full bio-plausibility by marrying bio-inspired sensor and locomotion platforms allowing end-to-end spike-based autonomous processing. This, to the best of our knowledge, is the first such closed-loop and end-to-end system that integrates event-sensing with SNN based processing and event-actuation with central pattern generation for gait transitioning locomotion system. Our system is not trained in an end-to-end network but the inference runs from data acquisition to gait actuation using vision processing SNN and SCPG.

The first contribution of our work lies in programming multiple gaits on a single SCPG with simple stochastic local weight updates for error minimization. We begin with a biologically known tripod gait for forward motion, similar to previous approaches and turning gaits for maximum rotation. We then program the required activation sequences on a single SCPG fabric. The weight updates happen, in absence of any complex error-backpropagation [7] mechanism or optimization [8] making the updates quick and hardware-friendly. It is then transferred to the locomotion platform for actuation where the gait selection neurons are driven by fixed synapses from the visual processing network to steer the robot. This seamless integration of the learning modality with evolution-inspired hard-coded connections between the visual response and the CPG network, provides an efficient neuro-inspired system design. Hard coding the connection between the visual response and the actuation is in accordance with the observation that many instinctive tasks observed in insects are shaped by evolution and do not need an explicit learning response on individual organisms. [28], [29].

Programming ‘n’ gaits with ‘n’ gait selection neurons results in the task of programming $15 + 6 \times n$ synaptic weights. Our network is constrained by the small size to allow on-board computation with limited resources. Our experimental results with the stochastic update algorithm allow simultaneously programming three gaits (‘n’ = 3) similar to [8]. The theoretical analysis of the maximum number of possible gaits that can be programmed on a single network of a given dimension is beyond the scope of this paper.

The second main contribution lies in developing not only the cognition or control platforms but the integration of them to enable close-loop spike-based autonomy. This enables the event stream generated by the DVS to enter the SNN pipeline and get processed using a binary event-based representation, without changing the paradigm of binary computation. The system demonstrates a particular task, namely, tracking a prey. Although the current setup is limited to cuboidal prey, the setup can be extended to more complex shapes and cluttered scenarios. For example, tuned Gabor filter [30] can detect inclined surfaces. Rich literature on training methods like SLAYER [31], E-prop [32], Tempotron [30], SPA [33] can extend the applicability to more advanced image processing. Our work focuses on building the end-to-end platform making it suitable for low-power neuro-inspired hardware for edge-robotics [34]–[37]. Evidences of such hardware designs from the industry are Loihi from Intel [22] and Truenorth from IBM [38].

The hardware implementation of the platform opens other interesting opportunities in low-power computing. Asynchronous sensory data can be processed asynchronously to spend power only when objects in the field of view move. All-to-all connections in visual information processing network suit the typical crossbar platforms proposed for non-volatile memory based implementation of synapses [39]. Various complex bio-inspired robotics tasks like Odometry and Simultaneous Localization and Mapping (SLAM) have been demonstrated for model-based algorithms [40]. Energy-constrained robots with ultra-low power ASICs for reinforcement learning [36] and swarm acceleration [37] have also come up recently . Extension of these algorithms to spiking neural networks with correspondingly optimized hardware and motion platform will bring exciting opportunities in this domain.

VII. Conclusion

We develop an end-to-end neuromorphic system that takes event-based visual data from a dynamic visual sensor and generates adaptive gait patterns for the locomotion of a hexapod robot in a predator-prey tracking scenario. The proposed method is fully bio-inspired carrying out the sensing to actuation in event-based processing. For learning various gaits, we propose a supervision-based weight adaptation algorithm to learn multiple gaits in a single CPG. Furthermore, the programmed SCPG is coupled with DVS to form a closed-loop control system that processes binary events to mimic predatory behaviour when the hexapod is approaching prey. Benefiting from the event-driven sensing and data processing, the proposed method shows high energy efficiency if this is implemented on well-known neuromorphic hardware (Intel’s Loihi platform). We demonstrate the feasibility of end-to-end neuromorphic systems for resource-constrained edge robotics.

Acknowledgement

This work was supported by CBRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

References

[1] T. G. Brown, “The intrinsic factors in the act of progression in the mammal,” *Proceedings of the Royal Society of London. Series B, containing papers of a biological character*, vol. 84, no. 572, pp. 308–319, 1911.

[2] A. I. Selverston, “Invertebrate central pattern generator circuits,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 365, no. 1551, pp. 2329–2345, 2010.

[3] S. L. Hooper, “Central pattern generators,” *e LS*, 2001.

TABLE I
SCPG LOCOMOTION SYSTEMS - COMPARISON WITH PRIOR WORK

Reference	**Locomotion Actuator**	**Actuator training**	**Sensory Input**	**Task**
[4]	Hexapod SCPG	Linear Equation Solving	-	Single Gait
[8]	Hexapod SCPG	Christiansen Grammar Evolution	-	Single Gait
[6]	Hexapod SCPG	Emulating Connectome Structure	-	Single Gait
[7]	Quadrapod SCPG	Remote supervision method [26]	-	Single Gait
[5]	Hexapod SCPG	Manual Design	-	Multiple Gaits
[23]	Hexapod SCPG	Reward STDP	Olfactory + Optical Camera	Gait Adaptation in Obstacle Crossing
[24]	Hexapod SCPG	Stochastic Reward	Gyrosensor + Optical Camera	Single Gait
[21]	Wheeled Robot	-	DVS	Obstacle Avoidance
[19]	UAV	-	DVS	Obstacle Avoidance
[20]	Hexapod PFM Actuation	-	DVS	Object Approaching
[25]	Hexapod SCPG	Neuron Parameter Tuning [27]	NAS	Remotely Controlled Gait Transition
This Work	Hexapod SCPG	Supervised	DVS	Gait Transitioned Object Approaching

[4] H. Rostro-Gonzalez, P. A. Cerna-Garcia, G. Trejo-Caballero, C. H. Garcia-Capulin, M. A. Ibarra-Manzano, J. G. Avina-Cervantes, and C. Torres-Huitzil, "A cpg system based on spiking neurons for hexapod robot locomotion," *Neurocomputing*, vol. 170, pp. 47–54, 2015.

[5] D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Peña, A. Jimenez-Fernandez, and A. Linares-Barranco, "Neuropod: a real-time neuromorphic spiking cpg applied to robotics," *Neurocomputing*, 2019.

[6] I. Polykretis and K. P. Michmizos, "An astrocyte-modulated neuromorphic central pattern generator for hexapod robot locomotion on intel's loihi," *arXiv preprint arXiv:2006.04765*, 2020.

[7] E. Aljalbout, F. Walter, F. Röhrbein, and A. Knoll, "Task-independent spiking central pattern generator: A learning-based approach," *Neural Processing Letters*, pp. 1–14, 2020.

[8] A. Espinal, H. Rostro-Gonzalez, M. Carpio, E. I. Guerra-Hernandez, M. Ornelas-Rodriguez, and M. Sotelo-Figueroa, "Design of spiking central pattern generators for multiple locomotion gaits in hexapod robots by christiansen grammar evolution," *Frontiers in neurorobotics*, vol. 10, p. 6, 2016.

[9] A. Jiménez-Fernández, E. Cerezuela-Escudero, L. Miró-Amarante, M. J. Domínguez-Morales, F. de Asís Gómez-Rodríguez, A. Linares-Barranco, and G. Jiménez-Moreno, "A binaural neuromorphic auditory sensor for fpga: A spike signal processing approach," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 4, pp. 804–818, 2016.

[10] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, *et al.*, "Event-based vision: A survey," *arXiv preprint arXiv:1904.08405*, 2019.

[11] E. Mueggler, H. Rebecq, G. Gallego, T. Delbruck, and D. Scaramuzza, "The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 142–149, 2017.

[12] X. Clady, C. Clercq, S.-H. Ieng, F. Houseini, M. Randazzo, L. Natale, C. Bartolozzi, and R. B. Benosman, "Asynchronous visual event-based time-to-contact," *Frontiers in neuroscience*, vol. 8, p. 9, 2014.

[13] N. J. Sanket, C. M. Parameshwara, C. D. Singh, A. V. Kuruttukulam, C. Fermüller, D. Scaramuzza, and Y. Aloimonos, "Evdodge: Embodied ai for high-speed dodging on a quadrotor using event cameras," *arXiv preprint arXiv:1906.02919*, 2019.

[14] S. Weiss, M. Achtelik, L. Kneip, D. Scaramuzza, and R. Siegwart, "Intuitive 3d maps for mav terrain exploration and obstacle avoidance," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 473–493, 2011.

[15] T. Delbruck and P. Lichtsteiner, "Fast sensory motor control based on event-based hybrid neuromorphic-procedural system," in *2007 IEEE international symposium on circuits and systems*, pp. 845–848, IEEE, 2007.

[16] D. Drazen, P. Lichtsteiner, P. Häfliger, T. Delbrück, and A. Jensen, "Toward real-time particle tracking using an event-based dynamic vision sensor," *Experiments in Fluids*, vol. 51, no. 5, p. 1465, 2011.

[17] T. Delbruck and M. Lang, "Robotic goalie with 3 ms reaction time at 4% cpu load using event-based dynamic vision sensor," *Frontiers in neuroscience*, vol. 7, p. 223, 2013.

[18] M. B. Milde, H. Blum, A. Dietmüller, D. Sumislawska, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "Obstacle avoidance and target acquisition for robot navigation using a mixed signal analog/digital neuromorphic processing system," *Frontiers in neurorobotics*, vol. 11, p. 28, 2017.

[19] L. Salt, G. Indiveri, and Y. Sandamirskaya, "Obstacle avoidance with lgmd neuron: towards a neuromorphic uav implementation," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, IEEE, 2017.

[20] F. Perez-Peña, A. Morgado-Estevez, A. Linares-Barranco, A. Jimenez-Fernandez, F. Gomez-Rodriguez, G. Jimenez-Moreno, and J. Lopez-Coronado, "Neuro-inspired spike-based motion: from dynamic vision sensor to robot motor open-loop control through spike-vite," *Sensors*, vol. 13, no. 11, pp. 15805–15832, 2013.

[21] H. Blum, A. Dietmüller, M. Milde, J. Conradt, G. Indiveri, and Y. Sandamirskaya, "A neuromorphic controller for a robotic vehicle equipped with a dynamic vision sensor," *Robotics Science and Systems, RSS 2017*, 2017.

[22] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[23] E. Arena, P. Arena, R. Strauss, and L. Patané, "Motor-skill learning in an insect inspired neuro-computational control system," *Frontiers in Neurorobotics*, vol. 11, p. 12, 2017.

[24] A. S. Lele, Y. Fang, J. Ting, and A. Raychowdhury, "Learning to walk: Spike based reinforcement learning for hexapod robot central pattern

generation," in *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 208–212, IEEE, 2020.

[25] D. Gutierrez-Galan, J. P. Dominguez-Morales, F. Perez-Pena, A. Jimenez-Fernandez, and A. Linares-Barranco, "Live demonstration: Neuromorphic robotics, from audio to locomotion through spiking cpg on spinnaker," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–1, IEEE, 2019.

[26] F. Ponulak, "Resume-new supervised learning method for spiking neural networks. institute of control and information engineering, poznoń university of technology," tech. rep., Tech. rep, 2005.

[27] B. Cuevas-Arteaga, J. P. Dominguez-Morales, H. Rostro-Gonzalez, A. Espinal, A. F. Jimenez-Fernandez, F. Gomez-Rodriguez, and A. Linares-Barranco, "A spinnaker application: design, implementation and validation of scpgs," in *International Work-Conference on Artificial Neural Networks*, pp. 548–559, Springer, 2017.

[28] R. Kanzaki, "Behavioral and neural basis of instinctive behavior in insects: Odor-source searching strategies without memory and learning," *Robotics and Autonomous Systems*, vol. 18, no. 1-2, pp. 33–43, 1996.

[29] A. M. Zador, "A critique of pure learning and what artificial neural networks can learn from animal brains," *Nature communications*, vol. 10, no. 1, pp. 1–7, 2019.

[30] R. Xiao, H. Tang, Y. Ma, R. Yan, and G. Orchard, "An event-driven categorization model for aer image sensors using multispike encoding and learning," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3649–3657, 2019.

[31] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," *arXiv preprint arXiv:1810.08646*, 2018.

[32] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature communications*, vol. 11, no. 1, pp. 1–15, 2020.

[33] Q. Liu, H. Ruan, D. Xing, H. Tang, and G. Pan, "Effective aer object classification using segmented probability-maximization learning in spiking neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 1308–1315, 2020.

[34] J.-H. Yoon and A. Raychowdhury, "31.1 a 65nm 8.79 tops/w 23.82 mw mixed-signal oscillator-based neuroslam accelerator for applications in edge robotics," in *2020 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 478–480, IEEE, 2020.

[35] J.-H. Yoon and A. Raychowdhury, "Neuroslam: A 65-nm 7.25-to-8.79-tops/w mixed-signal oscillator-based slam accelerator for edge robotics," *IEEE Journal of Solid-State Circuits*, vol. 56, no. 1, pp. 66–78, 2021.

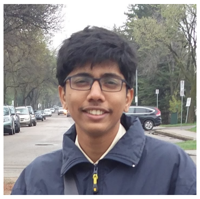
[36] A. Amravati, S. B. Nasir, S. Thangadurai, I. Yoon, and A. Raychowdhury, "A 55nm time-domain mixed-signal neuromorphic accelerator with stochastic synapses and embedded reinforcement learning for autonomous micro-robots," in *2018 IEEE International Solid - State Circuits Conference - (ISSCC)*, pp. 124–126, 2018.

[37] N. Cao, M. Chang, and A. Raychowdhury, "14.1 a 65nm 1.1-to-9.1tops/w hybrid-digital-mixed-signal computing platform for accelerating model-based and model-free swarm robotics," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, pp. 222–224, 2019.

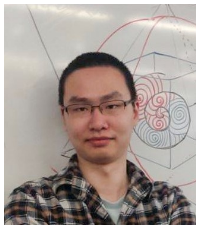
[38] M. V. DeBole, B. Taba, A. Amir, F. Akopyan, A. Andreopoulos, W. P. Risk, J. Kusnitz, C. O. Otero, T. K. Nayak, R. Appuswamy, *et al.*, "Truenorth: Accelerating from zero to 64 million neurons in 10 years," *Computer*, vol. 52, no. 5, pp. 20–29, 2019.

[39] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature materials*, vol. 18, no. 4, pp. 309–323, 2019.

[40] J. Yoon and A. Raychowdhury, "31.1 a 65nm 8.79tops/w 23.82mw mixed-signal oscillator-based neuroslam accelerator for applications in edge robotics," in *2020 IEEE International Solid- State Circuits Conference - (ISSCC)*, pp. 478–480, 2020.



Ashwin Lele received his B. Tech and M. Tech degrees from Indian Institute of Technology Bombay, Mumbai, India in 2019. He is currently a PhD student at the School of Electrical and Computer Engineering, Georgia Institute of Technology. He previously interned at the University of Alberta in 2017 and Intel India 2018. His research interest lies in brain-inspired spiking neural networks and low power computing for edge robotics.



Yan Fang received his M.S. and Ph.D. degree in Electrical and Computer Engineering from University of Pittsburgh. He is currently a postdoctoral researcher at the School of Electrical and Computer Engineering, Georgia Institute of Technology. His research focuses on brain-inspired computing systems based on emerging nanodevices, smart materials that compute, and applications in machine intelligence. He is also interested in dynamical systems, computational neuroscience and robotics.

Justin Ting an undergraduate student at Georgia Tech's School of Electrical and Computer Engineering and is also pursuing a minor in Computer Science. He joined Georgia Tech's Integrated Circuits and Systems Research Lab (ICSRL) during the Summer of 2017, where he tackled on-chip neural networks for robots, a project that he demonstrated at ISSCC 2018. He interned twice at Intel in 2018 and 2019, first as a Validation Intern, and second as an Advanced Analytics Intern. Justin continues to work with ICSRL on algorithms for bio-inspired hardware, and aspires for a PhD after he graduates. When Justin is not engineering, he likes to play piano, dance Chicago Footwork, compose music, or read.

Arijit Raychowdhury (Senior Member, IEEE) is currently a Professor in the School of Electrical and Computer Engineering at the Georgia Institute of Technology where he joined in January 2013. From 2013 to July 2019 he was an Associate Professor and held the ON Semiconductor Junior Professorship in the department. He received his Ph.D. degree in Electrical and Computer Engineering from Purdue University (2007) and his B.E. in Electrical and Telecommunication Engineering from Jadavpur University, India (2001). His industry experience includes five years as a Staff Scientist in the Circuits Research Lab, Intel Corporation, and a year as an Analog Circuit Researcher with Texas Instruments Inc. His research interests include low power digital and mixed-signal circuit design, design of power converters, sensors and exploring interactions of circuits with device technologies. Dr. Raychowdhury holds more than 25 U.S. and international patents and has published over 200 articles in journals and refereed conferences. He currently serves on the Technical Program Committees of ISSCC, VLSI Circuit Symposium, CICC, and DAC. He was the Associate Editor of the IEEE Transactions on Computer Aided Design from 2013-2018 and the Editor of the Microelectronics Journal, Elsevier Press from 2013 to 2017. He is the winner of Qualcomm Faculty Award, 2020; IEEE/ACM Innovator under 40 award; the NSF CISE Research Initiation Initiative Award (CRII), 2015; Intel Labs Technical Contribution Award, 2011; Dimitris N. Chorafas Award for outstanding doctoral research, 2007; the Best Thesis Award, College of Engineering, Purdue University, 2007; SRC Technical Excellence Award, 2005; Intel Foundation Fellowship, 2006; NASA INAC Fellowship, 2004; the Meissner Fellowship 2002. He and his students have won several fellowships and eleven best paper awards over the years. Dr. Raychowdhury is a Senior Member of the IEEE.