Learning to Walk: Bio-Mimetic Hexapod Locomotion via Reinforcement-Based Spiking Central Pattern Generation

Ashwin Sanjay Lele, *Student Member, IEEE*, Yan Fang, *Member, IEEE*, Justin Ting, *Student Member, IEEE*, and Arijit Raychowdhury, *Senior Member, IEEE*

***Abstract*— Online learning for the legged robot locomotion under performance and energy constraints remains to be a challenge. Methods such as stochastic gradient, deep reinforcement learning (RL) have been explored for bipeds, quadrupeds and hexapods. These techniques are computationally intensive and thus difficult to implement on edge computing platforms. These methods are also inefficient in energy consumption and throughput because of their reliance on complex sensors and pre-processing of data. On the other hand, neuromorphic computing paradigms, such as spiking neural networks (SNN), become increasingly favorable in low power computing on edge intelligence. SNN has exhibited the capability of performing reinforcement learning mechanisms with biomimetic spike time-dependent plasticity (STDP) of synapses. However, training a legged robot to walk in the synchronized gait patterns generated by a central pattern generator (CPG) in an SNN framework has not yet been explored. Such a method can combine the efficiency of SNNs with the synchronized locomotion of CPG based systems – providing breakthrough performance improvement of end-to-end learning in mobile robotics. In this paper, we propose a reinforcement based stochastic learning technique for training a spiking CPG for a hexapod robot which learns to walk using bio-inspired tripod gait without prior knowledge. The whole system is implemented on a lightweight raspberry pi platform with integrated sensors. Our method opens new opportunities for online learning with limited edge computing resources.**

***Index Terms*— Central pattern generator, spiking neural networks, spike time-dependent plasticity, stochastic reinforcement-based STDP, robotic locomotion.**

I. Introduction

RHYTHMIC activities like walking or breathing require temporally correlated muscle movements. Neuronal circuits in the spinal cord called Central Pattern Generators (CPG) cause coupled firing of motor neurons to actuate the limbs in a temporally correlated fashion [1]. The CPGs of animals and insects enable various gaits with seamless transitions between gaits and such biological systems can adapt the biomechanical interactions between the body and the environment [2]. This is controlled by both the brain as well as the central nervous system with co-ordination across various muscle groups [2]. The brain gathers the information from the sensory neurons, processes it with different cortices and modulate the CPG, constituting an end-to-end feedback system between sensing, spike-based processing and actuation [3]. The vestibular system in cockroaches, for example, is responsible for bio-circuitry that controls such pattern generation during walking [3]. Further, [4] studies the variation in the walking gait generated with gravitational forces acting upon the insect establishing a strong connection between CPG and gravity sensors in insects. The CPG also gets modulated by the information from the visual cortex to alter the gait to achieve a particular goal, such as tracking prey or approaching a source of food. These biological feedback systems can be inspirations of locomotion control in autonomous robots. Specifically, spiking neural networks (SNN) provide a computational tool for modelling the mechanism of CPG, which we define as Spiking-CPG (SCPG) in this work.

One of the key advantages in such systems comes from the low-power spiking neural network architecture used in the end-to-end decision making. Neuromorphic implementation of SNNs for cognitive tasks makes them strong candidates for edge-robotic platforms [5], [6]. Apart from the energy-efficiency, CPG controlling each leg independently enables decentralized control in the system. This is in sharp contrast to more traditional centralized control that can control global dynamics. Therefore, it results in the reduction of dimensionality of the sensing and control models and decreases the latency of processing [7]. This coupling of distributed computing and energy-efficiency with closed-loop architecture makes them candidates for control systems in edge-robots.

Electronic implementations of CPG have been tried out for legged robots for locomotion and prosthetic systems [8]–[10]. Fig. 1(a) shows a hexapod robot with a generic model of the spiking CPG (Fig. 1(b)) generating locomotion in this robot. Each leg is connected to one neuron and the spikes fired by the neurons trigger the motion of corresponding legs. Thus, the task of generating motion boils down to programming the desired spiking activity in the CPG. In previous work, a CPG based on digitized spiking neural network (SNN) has

Manuscript received July 1, 2020; revised August 28, 2020; accepted October 8, 2020. Date of publication October 22, 2020; date of current version December 11, 2020. This work was supported by the Center for Brain-inspired Computing (C-BRIC), one of six centers in the Joint University Microelectronics Program (JUMP), a Semiconductor Research Corporation (SRC) Program sponsored by the Defense Advanced Research Projects Agency (DARPA). This article was recommended by Guest Editor M. Valle. *(Corresponding author: Ashwin Sanjay Lele.)*

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30313 USA (e-mail: alele9@gatech.edu; yan.fang@gatech.edu; jting31@gatech.edu; arijit.raychowdhury@ece.gatech.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JETCAS.2020.3033135

2156-3357 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

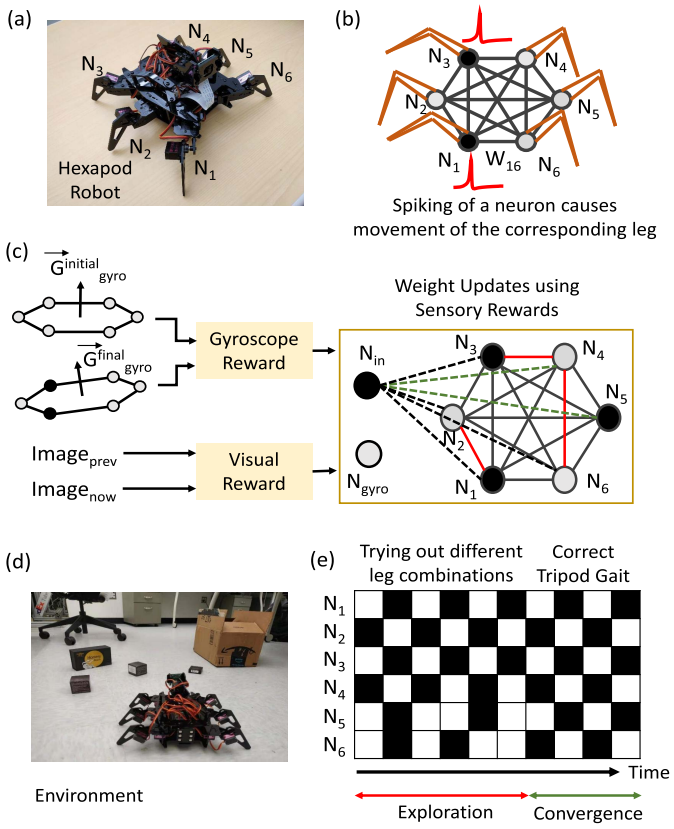


Fig. 1. (a) Hexapod robot with labeled neurons for mimicking the CPG (b) Generic SCPG model. Each neuron corresponds to a leg. Spiking of a neuron causes motion of the corresponding leg (c) Proposed algorithm converting visual and gyroscope inputs to rewards for reinforcement-based training of the SCPG. G_{gyro} is the gyro sensor's orientation vector (d) Office environment for demonstration. The robot without any prior knowledge of the spiking pattern required for walking learns to walk forward using tripod gait (e) SCPG spiking as the algorithm progresses. The spiking is random in the exploration phase where the SCPG tries out different combinations until it gets latched to the correct tripod gait showing convergence.

been proposed and implemented on an FPGA [9]. However, the generated patterns are pre-programmed, and the weights of the neural networks are reversed engineered form the patterns in known walking gaits. An evolutionary algorithm-based approach is also proposed for gait learning of hexapod robot [11] with offline training. A setup exploring imitation learning on with learning the gait pattern from pre-trained teacher robot is demonstrated [12]. There are a few previous publications on SNNs that focus on reinforcement learning [13], [14]. However, most of these attempts focus solely on the decision making and task planning of wheel robots. Thus, SNN based CPG with autonomous learning ability for legged robots has not been fully explored. The key factor unexplored in the field is the coupling between the sensory inputs like a gyroscope and visual data to the locomotion which forms the basis of end-to-end processing. An interesting task which lies unexplored comprises of enabling the agent to learn to walk or run autonomously without any prior knowledge of correct spiking pattern required for generating a stable forward motion.

In this work, we demonstrate an end-to-end SNN based CPG carrying out online processing of sensory information and learning of gait generation in hexapod robots. We use gyro sensor and camera as the sensory inputs to provide reward signals, which either reinforce (i.e, potentiate) or penalize (i.e., de-potentiate) the SNN weights to stochastically learn the correct gait for locomotion. This online learning is seen to autonomously result in bio-observed tripod gait in most of the cases. In some cases, convergence to non-bio-observed intermediate gaits is observed. These sub-optimal gaits still cause forward motion but at a slower speed. To the best of our knowledge, this is the first work that describes online learning for locomotion in robots using spiking network dynamics which may find potential application in edge robotics or other low power embedded systems.

II. Proposed Spiking CPG System: Algorithm and Hardware

The proposed system consists of a spiking CPG driven by input neurons, camera and gyroscope. The camera and gyroscope serve as the sensory inputs to the system for generating the rewards. The weights are modulated using the rewards, similar to previously used for spiking reinforcement learning scheme [15]. The CPG finally controls the locomotion of the hexapod robot. The SNN of CPG has to be trained to generate a sequence of leg motions so that the balance is maintained while walking, and thus prevent the robot from either falling or collapsing on its side. This embodies the notion of "learning to walk" which is a fundamental task accomplished by all legged organisms.

A. Network Structure

In our prototypical design, fully connected neurons form the SCPG network. The neurons obey leaky-integrate and fire (LIF) dynamics. Each neuron is connected to one of the legs and the firing of the neuron causes the corresponding leg to move. The movement consists of lifting, turning and landing of the leg, which are controlled by two servos on the hip and knee joint of each leg. However, we do not need to individually control this sequence of actions. Once the SCPG neuron corresponding to a leg fires, the series of actuations (e.g., lift, turn and land) follow one after another. An input neuron (N_{in}) and a gyroscope activated neuron (N_{gyro}) are connected to all CPG neurons. All neurons are excitatory. The CPG neurons have a refractory period of 2 time units. The LIF dynamics of the neurons are expressed below,

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

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

The leakage current is modelled with a decay factor α (Equation (1)). 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 V_j. 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). With spiking of

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

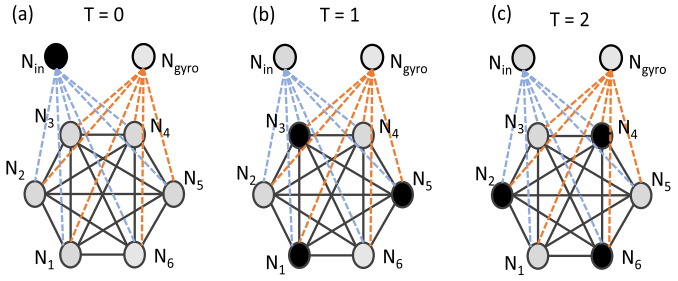


Fig. 2. Desired spiking pattern for forward motion (tripod gait) (a) The input neuron fires (b) This triggers neuron $N_{1,3,5}$ to fire in the next time instance making the corresponding legs move. (c) The internal connections in the SCPG make the remaining three neurons fire in the next time instance and the cycle repeats.

a pre-neuron, the input current is fed into the post-neuron in the immediate next cycle. Thus, we do not add any synaptic delay. Therefore, the membrane voltage is increased by the pre-synaptic spikes in the immediate next cycle. These simple discretized dynamics is voltage-based and easy to emulate on a digital hardware platform. Such a method bypasses the computation of time-delayed membrane current and enables a simpler model that can be implemented on a low-power computing platform. The CPG is implemented on a Raspberry Pi 3 B+ single-board computer, mounted on an Adeept Raspclaws hexapod.

The desired activity of neurons for tripod gait is shown in Fig. 2. The input neuron excites the network at t = 0. This now triggers the neurons in the SCPG to spike at t = 1. These neurons, in turn, trigger the other neurons in the CPG to spike at t = 2. The correct spiking pattern generates the tripod gait in which alternate legs move in two consecutive cycles. Thus, the input neuron keeps firing with a period of three time units triggering the CPG at the end of every gait instance. The gyro neurons N_{gyro} fires when at any step the robot loses balance. The reasoning behind this is explained in the next section.

B. Overview of the Algorithm

The algorithm described below computes the neural dynamics to uncover the spiking patterns of the CPG neurons. The spiking of the input neurons makes different combinations of legs to move. The algorithmic framework receives sensory inputs from the camera and the gyroscope (Fig. 3(a)) at the beginning and the end of every step; and checks if balance maintained and the forward motion is achieved. A positive reward is generated when the system determines that both balance and forward motion have been maintained. The reward is stochastically modulated to achieve weight updates.

Fig. 3 shows the block diagram that explains the algorithm. Every step is initialized by reading the gyroscope and capturing an image using the system camera. Next, we compute the neuronal spiking of CPG neurons by evaluating the LIF dynamics. The legs corresponding to the spiking CPG neuron(s) is(are) activated. The motion of a leg comprises of lifting-moving-landing actions carried out by two servos that drive the leg. Because each leg goes through the same sequence of actions when triggered, a single neuron activates both servos in a leg instead of the assignment of one neuron for

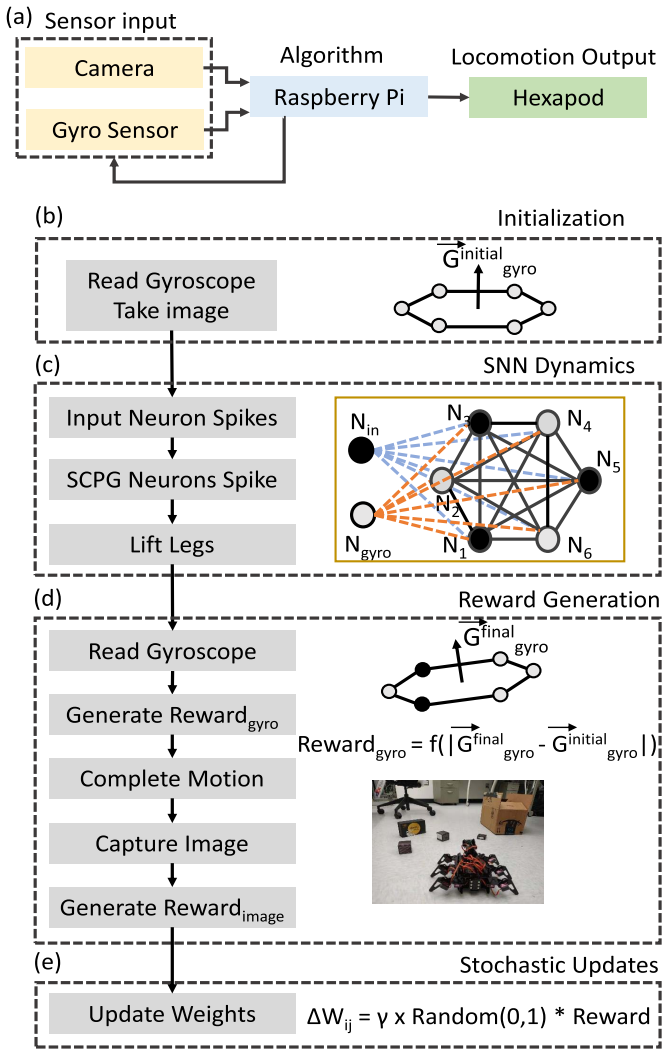


Fig. 3. Block diagram of the algorithm (a) The algorithm receives data from sensory inputs to drive locomotion of the hexapod which in turn generates new sensory inputs for close-loop processing (b) Every step is initialized with a gyroscope reading and an image of the surroundings (c) LIF neuronal dynamics compute the spiking of the neurons for that time instance (d) New sensory inputs are read when the legs are moved. If the desired result is achieved a positive reward is given and vice-versa (e) The weights are updated according to the calculated rewards with γ being the learning rate and the next cycle continues.

each servo. This simplifies the design significantly by reducing the number of neurons by half and the number of synapses by a factor of four. During activation, the legs are lifted and when they are half-way in the air, another gyroscope reading is captured to check if the balance is preserved. If the balance is lost, the legs are restored to the initial position without moving forward as this is guaranteed to give an erroneous gait. On the other hand, if the balance is preserved, the legs move forward causing a forward motion.

After completing the step, an image is captured using the camera. The photo-metric difference between the initial and final images indicates if a forward motion has occurred or not. The gyroscope and image differences are used to compute the reward for weight updates. Algorithm 1 shows the pseudo-code for the proposed algorithm.

The gyroscope driven neuron (N_{gyro}) mimics the biological modulation of CPG from the vestibular system. If the balance

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Learning to Walk

Initialize weights randomly $W_{in}, W_{Gyro}, W_{CPG}$
Initialize CPG neuron voltages, $V_{CPG}[0] = 0$
Initialize Spikes $S_{in} = 1, S_{Gyro} = 0, S_{CPG} = 0$
for $time = 1$ to T **do**
  Read camera ($Image_{init}$) and gyroscope ($G_{initial}$) for initial reading
  for $neuron = 1$ to 6 **do**
    $I = W_{in}S_{in} + W_{Gyro}S_{Gyro} + W_{CPG}S_{CPG}$
    $V_{neuron}[t] = V_{neuron}[t-1]/\alpha + I$
    if ($V_{neuron}[t] > V_{Thresh}$) **then**
      Update Spike, $S_{CPG} = 1$, $V_{neuron}[t] = 0$
      Move Corresponding Leg
    end if
  end for
  Read camera and gyro sensor for final reading
  Calculate reward
  Update weights
end for

is lost, N_{gyro} fires a spike in the next time instance. Losing balance requires an exploration of more alternatives for identifying a favourable combination of CPG firing. Spiking of N_{gyro} provides additional stimulation to the CPG to compensate for the loss of balance. This stimulation is in addition to the reward generated for altering the synaptic weights.

C. Reward Calculation

Both sensory inputs generate their rewards depending upon the performed action. The gyroscope reward ($Reward_{gyro}$) indicates the stability achieved by the robot while performing the step. Lesser the difference between the initial and final orientations of the gyroscope as shown in Fig. 3, higher is the goodness of the action and the reward. Algorithm 2 shows the reward generation scheme for the gyroscope. The gyroscope readings are typically noisy and therefore cannot be used directly as a reward value. Hence, the reward generated by the gyroscope is discretized into two categories. If the difference between the initial and final reading is below the threshold, it indicates that the action is stable. Therefore, this action earns a high positive reward from the gyroscope.

On the other hand, if the set of legs moved to cause the robot to tilt, this results in a high difference between the initial and final values of the gyroscope reading. Therefore this is regarded as an incorrect action. This is further categorized into two categories. If the total number of legs moved is less than three, this indicates insufficient activity in the CPG. Thus, the CPG requires more activity requiring higher weight values. Therefore a positive reward is given for this action. This particular case is commonly observed when two the legs on the same side of the hexapod are simultaneously lifted, which makes the robot to tilt. On the other hand, if more than three legs are being simultaneously activated, then the robot tilts and collapses. Hence, this CPG pattern needs to be automatically suppressed. This is achieved by designing a negative reward for this combination. The relative values of the rewards are optimized, similar to hyperparameter tuning in deep neural networks.

Algorithm 2 Gyroscope Reward Calculation

for $time = 1$ to T **do**
  $G_{initial}$ = Read gyro sensor
  Complete the movement
  G_{final} = Read gyro sensor
  if ($G_{final} - G_{initial} > G_{Thresh}$) **then**
    balance lost
    if (Number of legs moved > 3) **then**
      $Reward_{Gyro}$ = -2
    end if
    if (Number of legs moved < 3) **then**
      $Reward_{Gyro}$ = +2
    end if
  end if
  if ($G_{final} - G_{initial} < G_{Thresh}$) **then**
    balance maintained
    $Reward_{Gyro}$ = +5
  end if
end for

Apart from the $Reward_{gyro}$, the camera also generates a reward. This is needed to avoid the case where no movement occurs in the system and the gyroscope concludes the system is stable even without performing any action. Visual reward captures images before and after the movement and decides whether a forward motion has occurred. If the forward motion has occurred, this results in a positive reward, while motion in any incorrect direction provides a negative reward. The gyro-sensor also provides acceleration reading. However, the time integral of noisy reading for distance calculation gives an incorrect estimation of forward locomotion, which makes the camera an effective solution.

We have used a light-weight odometry [16] to determine if forward motion has occurred. The method is based on determining the photo-metric error in scan-line intensity profiles to estimate the amount of rotational and translational motion. Scan-lines are calculated by summing up all pixel values in a column. Equation 3 shows photometric error calculation where the shift and subtract operation calculates the error in the scan-line profiles. Shift corresponding to the minimum error corresponds to the rotation given by equation 4. σ is the constant converting the pixel shift to angular shift which is dependent upon the camera resolution, field of view, intensity etc. (equation 5). The velocity during the step is proportional to the minimum photometric error value as per equation 6.

$$f(s, I^j, I^k) = \frac{1}{w - |s|}\left(\sum_{n=1}^{w-|s|} |I^j_{n+max(s,0)} - I^k_{n-min(s,0)}|\right) \quad (3)$$

$$S_m = \underset{x \in [\rho - w, w - \rho]}{\operatorname{argmin}} f(s, I^j, I^k) \quad (4)$$

$$\Delta\theta = \sigma S_m \quad (5)$$

$$v = min[v_{cal} f(s, I^j, I^k), v_{max}] \quad (6)$$

The inference of odometry algorithm is used to calculate the visual reward. If no SCPG neuron fires and no leg moves,

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

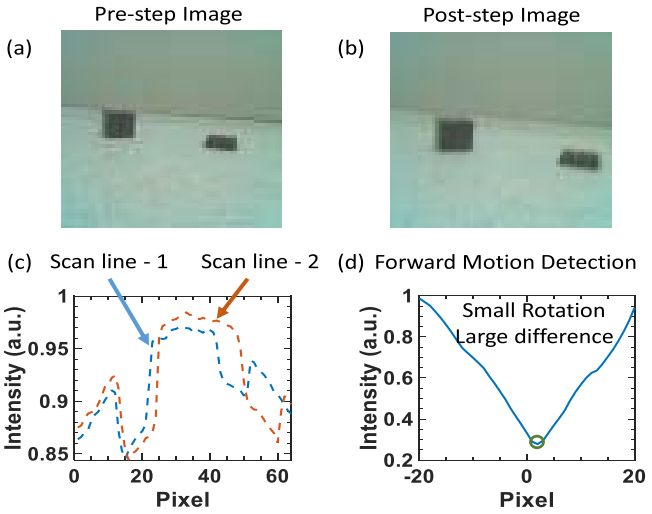


Fig. 4. (a) Image taken before taking the step (b) Image taken after completing the step (c) The column-wise sum of intensities of the pixel values for both images (scan-lines) (d) Minimum of scan-line difference is calculated for rotational matching for identifying the forward motion.

then the images are the same with a very small difference (caused by noise) and therefore the scanlines overlap with a very low photometric difference. Thus, if the magnitude of the total difference between scan lines is very small, we conclude that no motion has occurred incurring a negative reward.

In the case of motion, two cases occur. In the first case, the correct set of legs are moved resulting in forward motion. In this case, the odometry shows small rotation and significant translational motion. Thus, a positive reward is given. On the other hand, if an incorrect set of legs are moved which still results in the preservation of balance, this results in the robot moving along with rotation. Thus, a significant rotation along with translation is observed. This case also results in a negative reward.

Fig. 4 shows an example where the forward motion has occurred with slight rotation. The pre and post motion images are shown in Fig. 4(a,b). The resolution of the images is kept low at 64×64 to save power and latency in computation. The boxes are kept in the environment of the robot to provide basis points. Scan lines for an image are calculated by adding up all pixels in a column. Scan-lines corresponding to the images are plotted in Fig. 4(c). Now, the rotation and forward motion are calculated as described previously. Fig. 4(d) shows the total difference in scan-lines upon shift and subtract operation. The difference minimizes at the rotation of 3 pixels indicating the amount of rotation as small. However, the absolute difference in the intensity is large. This shows that the images have not rotated much but have caused a significant intensity difference. This is attributed to the forward motion and obtains a positive reward.

The reward is provided as a binary value with forward motion resulting in a reward of $\{+1\}$. Otherwise, a penalty of $\{-1\}$ is generated. This binarization makes the algorithm robust to low resolution as shown. Additionally, this allows high noise tolerance in the lightweight vision system. The variable intensity of the surrounding can be calibrated in the empirical parameter σ making the vision system suited for low-power applications. We note that the negative reward generated by the movement should not be very high in magnitude; particularly, for the first few iterations. This allows the system and the corresponding SNN network to explore and determine the correct CPG activation patterns. This illustrates the very traditional exploration-exploitation conundrum typical of all real-time learning systems [17]. We achieve a favourable trade-off between exploration and exploitation by modeling the reward at time instance t as:

$$Reward_{total}[t] = Reward_{gyro} + Reward_{visual}(t/T_1) \quad (7)$$

A positive total reward occurs when the correct action has taken place or when the network is inactive compared to the desired activity. A negative reward occurs to suppress unnecessary activity.

TABLE I
HARDWARE PLATFORM

Item	Component
Locomotion Platform	Adeept RaspClaws Hexapod Spider Robot [18]
Computation Platform	Raspberry pi 3 Model B+ [19]
Vestibular Input	Gy-521 MPU-6050 MPU6050 gyro sensor [20]
Visual Input	PiCamera [21]

D. Mechanisms for Synaptic Weight Updates

Combining synaptic reinforcement with reward function has been demonstrated previously in [15]. In neuro-biology, this mimics the release of dopamine in the human brain that acts as a reward for performing a certain desired action. The synaptic updates are generated for the synapses whose pre-neurons spike in the previous time instance. The updates are the reward generated by the action, modulated by a random number between zero and one. A positive (negative) reward causes an increase (decrease) in the synaptic weight. This enables the post-synaptic neuron to spike faster (slower) in subsequent time instances. We encourage stochasticity in the network by modulating the reward with a random number to avoid the system from getting stuck in a sub-optimal firing pattern. This also allows the system to quickly reach and converge on the final, desired firing pattern. The change in weights is calculated as given below (γ is the learning rate). The learning rate must be chosen carefully to enable the system to reach the desired final state with a minimum number of learning iterations. The weight evolution with time is given by

$$W_{ji}[t+1] = W_{ji}[t] + \gamma \times Reward_{total}[t] \times random(0,1) \quad (8)$$

The weight values are clipped to a maximum of W_{high} and a minimum of W_{low}. In the current design $W_{high} = 12$ and $W_{low} = 0$ are chosen.

E. Hardware Platform and Verification

The hardware details are shown in Table I. The processor has 4 cores and operates at 1.5GHz frequency. The demonstration is carried out in an indoor office environment shown in Fig. 1.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

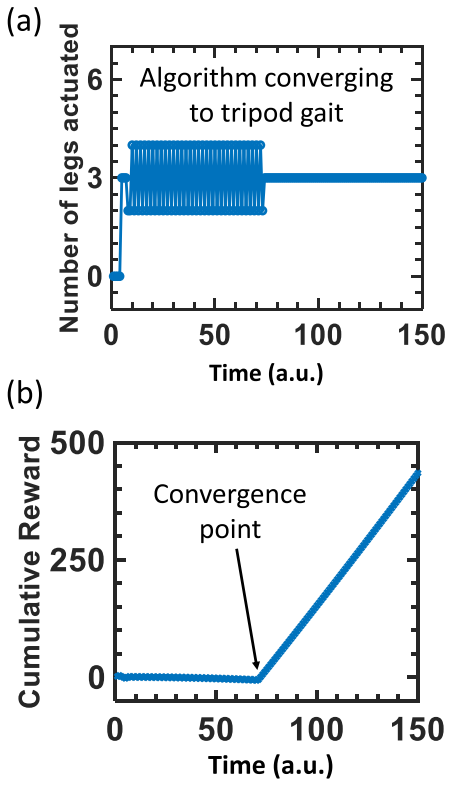


Fig. 5. (a) The number of legs moved at a time instance oscillates around 3 and converges to 3 in a tripod gait resulting in accumulation of high constant reward (b) Accumulated reward with time. The reward is negative till the network converges to correct tripod gait after which it keeps increasing. Figure modified from [22].

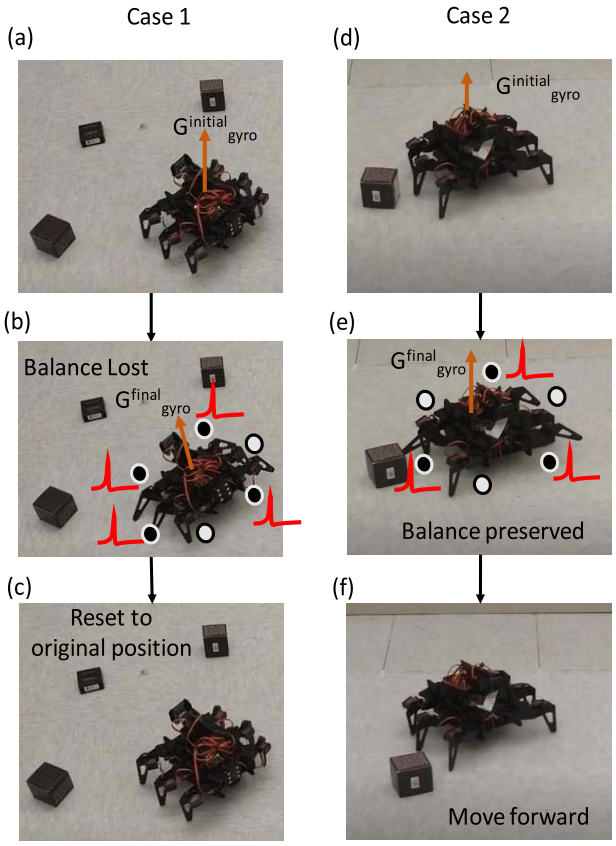


Fig. 6. (a) Initial position of the robot at the beginning of training (b) With the movement of four legs, the robot loses balance making the gyroscope read the tilted value (c) Robot is reset to the original position by taking the legs back (d) Robot standing after learning the correct gait pattern (e) Movement of correct set of three legs, robots preserves the balance (f) The legs are placed forward completing the forward motion.

III. Results

A. Simulation Results

A complete modeling and simulation framework has been set up. Fig. 5 shows the time evolution of the end-to-end simulation of the system. As the simulation progresses with SNN dynamics and with the different leg combinations, synthesized visual/gyroscope data are captured and the reward is calculated. The corresponding CPG patterns and the movement of the legs are recorded. We note that the hexapod starts with no movement. Gradually it explores two firing patterns that correspond to moving 2 and 4 legs simultaneously. When it finally reaches the correct tripod gait, through a series of rewards and penalties, the synaptic weights reach their steady-state values and the hexapod continues to walk in the tripod gait. In steady-state three legs are actuated simultaneously and the hexapod maintains balance and moves forward. Fig. 5(b) shows the total accumulated reward over time. With both positive and negative rewards coming in, the cumulative reward remains low until the correct gait found. It rapidly increases after that with a constant high positive reward.

B. Hardware Demonstration

We apply the proposed method to an Adeept hexapod robot with the hardware configuration described in section III.C. The videos demonstrating "Learning to Walk" are available on Youtube. In the first demonstration video (demo-1)[1], the learning process converges to the target gait pattern at the 66^{th} cycle. Fig. 6 shows screenshots of different instances in the evolution of the algorithm. Fig. 6(a) shows the initial orientation of the gyroscope in the exploration phase. When the hexapod explores a configuration where four legs are simultaneously lifted, as shown in Fig. 6(b), the robot loses the balance. This makes the orientation of the gyroscope to deviate significantly from the original position. Hence, a negative reward is generated. In this case, the legs that were triggered are restored to the original position (Fig. 6(c)) as this is guaranteed to give erroneous motion. Fig. 6(d) shows the position of the robot before it takes a step after the algorithm has converged correctly. By simultaneously lifting the correct three legs, the robot preserves its balance and the deviation of the gyroscope's reading is small. The legs are now lowered in a forward position, and the robot moves forward. Captured images from the initial and final position validate the forward motion, which in turn results in a positive reward.

However, the weight updates being stochastic do not always result in the convergence to bio-observed tripod gait. The reward is determined by only the balance preservation and forward motion and not by the exact combination of legs that

[1]Demo 1: https://www.youtube.com/watch?v=1HqeISAkAs4&feature=youtu.be

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

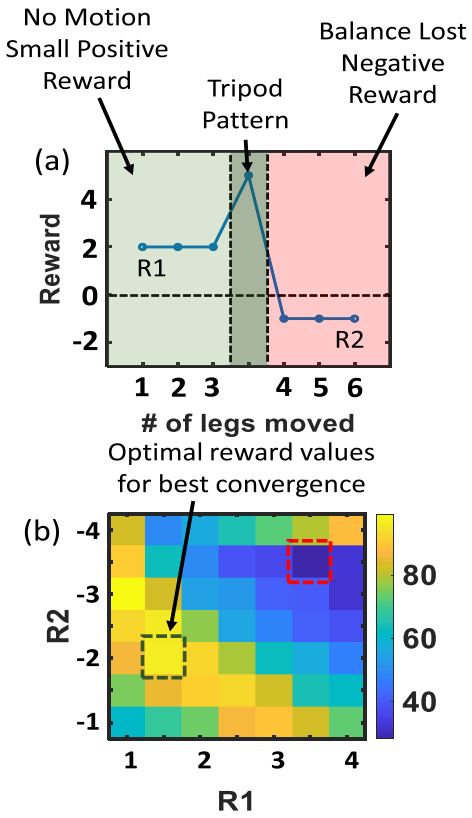


Fig. 7. (a) Reward value with the number of legs moved. For less than 2 legs moving, positive reward (R1) is given to encourage more spiking required for tripod gait. A negative reward (R2) is when more than 3 legs move. (b) Optimal combination of R1 and R2 for the highest convergence percentage.

are to be moved to cause the tripod gait. Therefore, the convergence also happens to intermediate non-bio-observed gaits that result in forward motion with balance preservation. These are caused by a sub-optimal combination of legs causing the motion at a slower speed. These cases correspond to weight parameters getting stuck into local minima.

Another demonstration video (demo-2)[2], shows one such case of learning process converged to an unwanted alternative gait pattern at the 7th cycle (0:18). This gait pattern is a local minimum in an expected learning process. The gait also shows maintenance of balance along with forward motion without using the biologically observed combination of tripod gait. The hexapod can move forward with this gait, but less efficient when compared to the bio-inspired target gait. The occasional tremor of servos is caused by an instantaneously insufficient current supply.

C. Reward Generation

The reward described in the previous section takes a general form as shown in Fig.7(a). If less than 3 legs spike simultaneously, the balance is preserved but the more excitation is required for exploration of other patterns. This requires positive reward (R1) for reinforcing the weights to explore more combinations with three legs. On the other hand, spiking of more than 3 SCPG neurons causes the robot to lose balance

[2]Demo 2: https://www.youtube.com/watch?v=ypW0V23gEj0&feature=youtu.be

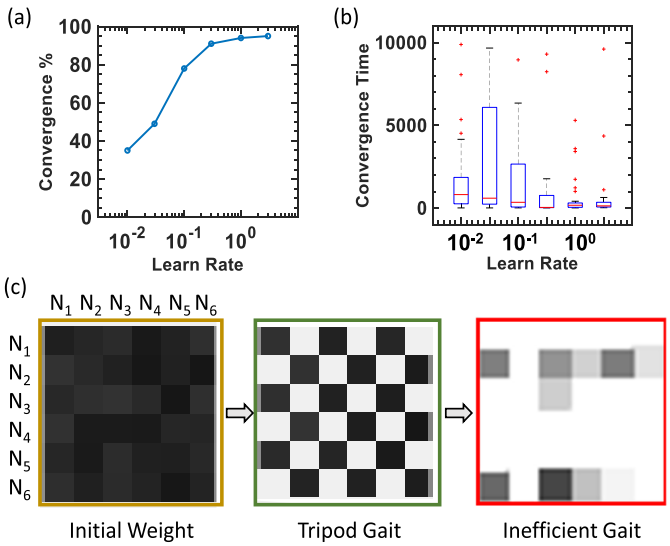


Fig. 8. (a) Percentage of simulations converging to the correct tripod gait with varying learning rate. The gait pattern after correct convergence can be seen in demo-1. Simulations not converging to the tripod gait still move with inefficient gaits as observable in demo-2. (b) Distribution of convergence time with learning rate. Convergence time affects the power consumed in learning. Median convergence time is 170-time instances. (c) Weight matrices of the SCPG before and after the learning. Convergence to tripod gait makes alternate neurons stimulating each other. Figure is modified from [22].

acquiring a negative reward (R2). A high positive reward is accumulated when the correct set of legs fire, thus maintaining balance as well as ensuring forward motion. We are concerned with finding the optimal values for R1 and R2.

100 simulations for each combination of the rewards shown in Fig 7(b) are run to identify the fraction of simulations converging to the tripod gait. The plot shows almost 90% convergence for the combination shown by a green bounding box. The red bounding box shows the case (R1 and R2 combination) that results in the lowest fraction of convergence. Since we enable stochastic updates, a monotonic trend is not established. However, it can be noted that for intermediate values of R1 and R2, the convergence is high whereas for extreme values the fraction of cases where the system convergences to the correct tripod gait reduce. This is because, for very small rewards, the weight changes are too small to cause convergence while for large rewards, the updates make the CPG system to oscillate between multiple stable gaits, without latching on to the correct gait. These optimal values of R1 and R2 were determined through simulations and used in the hardware platform.

D. Determining the Learning Rate

The stochastic updates in the algorithm are intrinsically connected to the optimal choice of learning rate (γ). To identify the best learning rate, we simulate 100 iterations of the algorithm with different random initial conditions for different learning rates and with the rewards, R1 and R2 that were described previously. This is shown in Fig. 8. Fig. 8(a) shows the percentage of simulations converging to the correct tripod gait. The results show that the system convergences faster as the learning rate increases. The cases where convergence is achieved, the robot moves forward efficiently. In cases, where the system converges to non-tripod, non-bio-mimetic gaits,

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

we still observe location albeit inefficiently. These inefficient gaits cause the hexapod to frequently turn its direction of motion and the hexapod fails to reliably move forward at a constant speed.

Another important parameter controlling the energy consumption of the algorithm is the time to converge for the algorithm. Fig. 8(b) shows the convergence time for the iterations converging correctly. Fig. 8(c) shows the weight maps of synapses forming the connections between the CPG the neurons before and after the completion of the learning. For convergence to the global minima of a bio-mimetic tripod gait, the random weights (initialization phase) among the neurons enter into a periodic pattern such that neurons 1,3,5 drive neurons 2,4,6 and vice versa.

E. Estimated Power Consumption for Locomotion in a Neuromorphic Processor

The simple two-layered spiking network used in this network is expected to show a high reduction in the energy required in learning as opposed to conventional approaches involving artificial neural networks. We envision future systems where ultra-low power MEMS [23] and PZT [24] actuators, combined with ultra-low power neuromorphic algorithms and hardware [25] can realize power and volume constrained robots at the edge of the cloud.

To estimate the energy consumed by the algorithm, we calculate the total number of spikes issued by the SCPG in the course of learning to estimate the energy spent in learning. We run 100 iterations of the algorithm in software to identify the statistical convergence to the correct gait pattern. The median of the total number of spikes issued for correct convergence is 170. To estimate how a potential neuromorphic ASIC will perform, we note that Intel's Loihi [15] requires 1.7 nJ for generation a single spike. Therefore, we estimate that the total energy consumption in the learning process on an equivalent design is $\approx$ 289 nJ. After the system has learned the correct gait, the energy consumed in every step is $\approx$ 9.1 nJ. These estimates are simply meant to motivate further work in neuromorphic ASIC design which can enable ultra-low-power SCPG based learning for legged motion. Along with the low energy consumption, this work also shows an end-to-end spike driven learning system for sensor data processing and actuation.

IV. Discussion

A. Comparison With the Prior Work

The comparison Table II reveals that this is the first work showing online learning of a gait using the sensory inputs from the environment. It is also worth noting that the learning occurs in the absence of any model and uses only the rewards generated during the motion of the legs of the hexapod. The algorithm explores the possible combinations before stabilizing to the correct gait. Convergence to non-biologically observed gaits poses interesting future research directions. It is unclear if all non-bio-mimetic gaits are inefficient, although we have empirically observed that to be the case. For more complex systems with more legs, it remains to be seen in bio-mimetic CPG results in the most efficient patterns of actuation.

TABLE II
Comparison With Prior Work

Reference	Training Approach	Sensory Feedback	Online / Offline
[9]	Linear Equation Solving	None	Offline
[27]	Grammar Evolution	None	Offline
[28]	Reward STDP	Olfactory + Visual	Offline
[30]	Equally weighted synapses	None +	Offline
[31]	Remote supervision method [32]	None	Offline
[33]	Emulating Connectome Structure	None	Offline
[34]	Manual Design	None	Offline
This Work	Stochastic Reward	Balance + Visual	Online

B. Biological Basis of the Model Presented

Our SCPG is triggered by two neurons namely N_{gyro} and N_{in} where N_{gyro} corresponds to the input signal from the vestibular system. This mimics the principles discussed in [3] where the authors have demonstrated the close interactions between the vestibular system and central pattern generation in invertebrates. Further, [4] studied the variation in the walking gait under different gravitational forces acting upon an insect and established a strong connection between CPG and gyro sensors in natural organic systems. Identification of the exact set of neurons in vestibular sensing in the housefly has also been observed and is seen to be present outside the CPG. [33] confirms the bio-plausibility of the proposed model, that CPG and gyro sensors and coupled but independent. This proposed model uses single neuron corresponding to each leg along with a simplified view of the gyro-system. Despite its apparent simplicity, the model is bio-plausible and we observe how a CPG system "learns to walk" using simple rules of reward-based learning.

Hoyt *et. al.* showed the natural gait at a speed in horses consumes the least amount of oxygen corresponding to the smallest energy expenditure [34]. This is comparable to the observation where inefficient gaits are observed along with the tripod gait but the motion caused by them is slower. This unfolds another interesting idea of reward design to incorporate the efficiency into the picture to mimic the natural optimization that has occurred through evolution.

C. Extension to Customized Hardware

The power-efficient operation, as demonstrated with SCPG, naturally extends its application to bio-inspired edge-robotics. Amaravati et. al. demonstrated conventional Q-learning for autonomous motion on a wheeled robot with tight constraints on power consumption [35], [36]. The current algorithm and hardware platform demonstrated here can push the boundaries even further with legged robots. Insects also demonstrate coordinated activities in swarms that have been recently mimicked

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

in electronic platform [37], [38]. Extension of learning based CPG in swarms of legged hexapods need to be explored.

V. Conclusion

We propose a closed-loop end-to-end spiking central pattern generator with real-time learning based on a bio-plausible spiking neural network. We demonstrate the proposed SCPG on a hexapod robot and achieve autonomous online reinforcement learning of bio-mimetic walking gaits in an energy-efficient manner. The computational requirement of the proposed online learning system is light-weight and it is implemented on a simple embedded system. Interestingly, we note that learning process converges to the bio-observed tripod gaits in most of the cases; while in other cases it converges to sub-optimal gaits that still enable locomotion, albeit inefficiently.

References

[1] A. I. Selverston, "Invertebrate central pattern generator circuits," *Phil. Trans. Roy. Soc. B, Biol. Sci.*, vol. 365, no. 1551, pp. 2329–2345, Aug. 2010.

[2] N. Bernstein, "The co-ordination and regulation of movements," in *The Co-Ordination and Regulation of Movements*. New York, NY, USA: Pergamon, 1967. [Online]. Available: https://www.worldcat.org/title/co-ordination-and-regulation-of-movements/oclc/598328285

[3] S. Zill, "Invertebrate neurobiology: Brain control of insect walking," *Current Biol.*, vol. 20, no. 10, pp. R438–R440, May 2010.

[4] C. S. Mendes, S. V. Rajendren, I. Bartos, S. Márka, and R. S. Mann, "Kinematic responses to changes in walking orientation and gravitational load in drosophila melanogaster," *PLoS ONE*, vol. 9, no. 10, Oct. 2014, Art. no. e109204.

[5] P. A. Merolla *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, Aug. 2014.

[6] M. Davies *et al.*, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Jan. 2018.

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

[8] D. Owaki and A. Ishiguro, "A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping," *Sci. Rep.*, vol. 7, no. 1, pp. 1–10, Dec. 2017.

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

[10] X. Guo, L. Chen, Y. Zhang, P. Yang, and L. Zhang, "A study on control mechanism of above knee robotic prosthesis based on CPG model," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2010, pp. 283–287.

[11] 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 Neurorobotics*, vol. 10, p. 6, Jul. 2016.

[12] J. Ting, Y. Fang, A. Sanjay Lele, and A. Raychowdhury, "Bio-inspired gait imitation of hexapod robot using event-based vision sensor and spiking neural network," 2020, *arXiv:2004.05450*. [Online]. Available: http://arxiv.org/abs/2004.05450

[13] R. V. Florian, "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity," *Neural Comput.*, vol. 19, no. 6, pp. 1468–1502, Jun. 2007.

[14] E. Rueckert, D. Kappel, D. Tanneberg, D. Pecevski, and J. Peters, "Recurrent spiking networks solve planning tasks," *Sci. Rep.*, vol. 6, no. 1, pp. 1–10, Aug. 2016.

[15] E. M. Izhikevich, "Solving the distal reward problem through linkage of STDP and dopamine signaling," *Cerebral Cortex*, vol. 17, no. 10, pp. 2443–2452, Oct. 2007.

[16] M. J. Milford and G. F. Wyeth, "Mapping a suburb with a single camera using a biologically inspired SLAM system," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1038–1053, Oct. 2008.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[18] *Adeept_RaspClaws*. [Online]. Available: https://www.adeept.com/video/static1/itemsfile/49523Tutorial_V2.pdf

[19] [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

[20] [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/mpu-6000-datasheet1.pdf

[21] [Online]. Available: https://www.raspberrypi.org/products/camera-module-v2/

[22] A. S. Lele, Y. Fang, J. Ting, and A. Raychowdhury, "Learning to walk: Spike based reinforcement learning for hexapod robot central pattern generation," in *Proc. 2nd IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Aug. 2020, pp. 208–212.

[23] D. Tanaka, Y. Uchiumi, S. Kawamura, M. Takato, K. Saito, and F. Uchikoba, "Four-leg independent mechanism for MEMS microrobot," *Artif. Life Robot.*, vol. 22, no. 3, pp. 380–384, Sep. 2017.

[24] D. Kim, Z. Hao, J. Ueda, and A. Ansari, "A 5 mg micro-bristle-bot fabricated by two-photon lithography," *J. Micromech. Microeng.*, vol. 29, no. 10, Oct. 2019, Art. no. 105006.

[25] *HM01B0 Ultra Low Power Camera Sensor*, Himax, Taipei, Taiwan, 2018.

[26] L. Patanè, R. Strauss, and P. Arena, *Nonlinear Circuits and Systems for Neuro-Inspired Robot Control*. Berlin, Germany: Springer, 2018.

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

[28] 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*, vol. 381, pp. 10–19, Mar. 2020.

[29] E. Aljalbout *et al.*, "Task-independent spiking central pattern generator: A learning-based approach," *Neural Process. Lett.*, vol. 51, pp. 2751–2764, 2020, doi: 10.1007/s11063-020-10224-9.

[30] F. Ponulak, "Resume-new supervised learning method for spiking neural networks," Inst. Control Inf. Eng., Poznoń Univ. Technol., Poznan, Poland, Tech. Rep., 2005.

[31] I. Polykretis, G. Tang, and K. P. Michmizos, "An astrocyte-modulated neuromorphic central pattern generator for hexapod robot locomotion on Intel's Loihi," in *Proc. Int. Conf. Neuromorphic Syst.*, 2020, pp. 1–9.

[32] A. Spaeth, M. Tebyani, D. Haussler, and M. Teodorescu, "Neuromorphic closed-loop control of a flexible modular robot by a simulated spiking central pattern generator," in *Proc. 3rd IEEE Int. Conf. Soft Robot. (RoboSoft)*, May 2020, pp. 46–51.

[33] C. S. Mendes, I. Bartos, T. Akay, S. Márka, and R. S. Mann, "Quantification of gait parameters in freely walking wild type and sensory deprived drosophila melanogaster," *eLife*, vol. 2, Jan. 2013, Art. no. e00231.

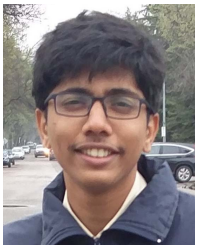
[34] D. F. Hoyt and C. R. Taylor, "Gait and the energetics of locomotion in horses," *Nature*, vol. 292, no. 5820, pp. 239–240, Jul. 1981.

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

[36] A. Amaravati, S. B. Nasir, J. Ting, I. Yoon, and A. Raychowdhury, "A 55-nm, 1.0–0.4V, 1.25-pJ/MAC time-domain mixed-signal neuromorphic accelerator with stochastic synapses for reinforcement learning in autonomous mobile robots," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 75–87, Jan. 2019.

[37] N. Cao, M. Chang, and A. Raychowdhury, "A 65-nm 8-to-3-b 1.0–0.36-V 9.1–1.1-TOPS/W hybrid-digital-mixed-signal computing platform for accelerating swarm robotics," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 49–59, Jan. 2020.

[38] N. Cao, M. Chang, and A. Raychowdhury, "14.1 a 65 nm 1.1-to-9.1TOPS/W hybrid-digital-mixed-signal computing platform for accelerating model-based and model-free swarm robotics," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 222–224.



Ashwin Sanjay Lele (Student Member, IEEE) received the B.Tech. and M.Tech. degrees from the Indian Institute of Technology Bombay, Mumbai, India, in 2019. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology. Previously, he interned at the University of Alberta in 2017, and Intel India in 2018. His research interests include brain-inspired spiking neural networks and low power computing for edge robotics.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.

Yan Fang (Member, IEEE) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Pittsburgh. He is currently a Post-Doctoral Researcher with the School of Electrical and Computer Engineering, Georgia Institute of Technology. His research interests include 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 (Student Member, IEEE) is currently pursuing the bachelor's degree in computer science with the School of Electrical and Computer Engineering, Georgia Tech. He joined the Georgia Tech's Integrated Circuits and Systems Research Lab (ICSRL) in Summer 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. He continues to work with ICSRL on algorithms for bio-inspired hardware, and aspires for a Ph.D. after his graduation.

Arijit Raychowdhury (Senior Member, IEEE) received the B.E. degree in electrical and telecommunication engineering from Jadavpur University, India, in 2001, and the Ph.D. degree in electrical and computer engineering from Purdue University in 2007.

He is currently a Professor with the School of Electrical and Computer Engineering, 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. His industry experience includes five years as a Staff Scientist with the Circuits Research Lab, Intel Corporation, and a year as an Analog Circuit Researcher with Texas Instruments Incorporated. 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. He 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 is the winner of Qualcomm Faculty Award in 2020, the IEEE/ACM Innovator under 40 award, the NSF CISE Research Initiation Initiative Award (CRII) in 2015, the Intel Labs Technical Contribution Award in 2011, the Dimitris N. Chorafas Award for outstanding doctoral research in 2007, the Best Thesis Award, College of Engineering, Purdue University, in 2007, the SRC Technical Excellence Award in 2005, the Intel Foundation Fellowship in 2006, the NASA INAC Fellowship in 2004, and the Meissner Fellowship in 2002. He and his students have won several fellowships and eleven best paper awards over the years. He was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS from 2013 to 2018, and an Editor of the *Microelectronics Journal* (Elsevier) from 2013 to 2017.

Authorized licensed use limited to: Georgia Institute of Technology. Downloaded on June 28,2021 at 12:35:05 UTC from IEEE Xplore. Restrictions apply.