

# Astrocyte Regulated Neuromorphic Central Pattern Generator Control of Legged Robotic Locomotion

Zhuangyu Han, Abhranil Sengupta, *Senior Member, IEEE*

**Abstract**—Neuromorphic computing systems, where information is transmitted through action potentials in a bio-plausible fashion, is gaining increasing interest due to its promise of low-power event-driven computing. Application of neuromorphic computing in robotic locomotion research have largely focused on Central Pattern Generators (CPGs) for bionics robotic control algorithms - inspired from neural circuits governing the collaboration of the limb muscles in animal movement. Implementation of artificial CPGs on neuromorphic hardware platforms can potentially enable adaptive and energy-efficient edge robotics applications in resource constrained environments. However, underlying rewiring mechanisms in CPG for gait emergence process is not well understood. This work addresses the missing gap in literature pertaining to CPG plasticity and underscores the critical homeostatic functionality of astrocytes - a cellular component in the brain that is believed to play a major role in multiple brain functions. This paper introduces an astrocyte regulated Spiking Neural Network (SNN)-based CPG for learning locomotion gait through Reward-Modulated STDP for quadruped robots, where the astrocytes help build inhibitory connections among the artificial motor neurons in different limbs. The SNN-based CPG is simulated on a multi-object physics simulation platform resulting in the emergence of a trotting gait while running the robot on flat ground. 23.3 $\times$  computational power savings is observed in comparison to a state-of-the-art reinforcement learning based robot control algorithm. Such a neuroscience-algorithm co-design approach can potentially enable a quantum leap in the functionality of neuromorphic systems incorporating glial cell functionality.

**Index Terms**—Spiking neural networks, Central pattern generator, Astrocyte regulation, Locomotion gait.

## I. INTRODUCTION

CURRENT brain-inspired engineered systems have primarily focused on the emulation of bio-plausible computational models of neurons and synapses. Incorporation of other cellular units from the brain is lacking. Over the past few years, a growing body of evidence has demonstrated that glial cells, and in particular astrocytes, play an important role in the maintenance and modulation of neuronal dynamics facilitating brain function [1]. This work is inspired by recent theories suggesting that rich temporal dynamics such as synchrony form the dynamical basis of learning and memory and are expected to play a key role in enabling a *dynamical view of intelligence* for the design of next-generation brain-inspired engineered learning systems (see report based on 6th US/NIH BRAIN Initiative Investigators Meeting [2]). In particular, our work focuses on spinal central pattern generators (CPGs), which are neural circuits that generate spontaneous rhythmic

patterns and serve as a model “brain-like” system to design engineered learning platforms, specifically robotic locomotion controllers.

Multiple parallel works have pointed toward the critical role of astrocytes in spinal CPGs. Recent literature has reported that astrocytes in the spinal cord can influence the activity of the spinal motor network. Specifically, the excitation of spinal cord astrocytes causes a decline in fictive locomotion frequency [3]. Neuroscientists have further illustrated that excited astrocytes suppress the rhythmic locomotion activity by releasing ATP, which is metabolized into adenosine (ADO) immediately. Adenosine works as an antagonist of the excitatory neurotransmitter dopamine (DA) in the motor neurons [4]. Also, researchers have observed that, in larval zebrafish, the futile swimming effort can activate radial astrocytes, which suppresses continuous unsuccessful attempts [5]. Further, it is reported that such inhibitory effect of astrocytes on swimming in zebrafish is mediated by a signaling pathway involving ATP - adenosine - A<sub>2</sub> receptor [6]. Such results are consistent with the claim that astrocytes play a homeostatic role in the neural system [7]. Driven by such insights, we investigate the homeostatic regulation role of astrocytes in neuromorphic Spiking Neural Network (SNN) based CPG circuits for gait learning in quadruped robots. Neuromorphic robotic control has been recently gaining attention due to the promise of enabling low-power edge robotics in resource-constrained environments [8]–[10]. However, the works remain preliminary with application to simple robotic platforms primarily due to overtly simplistic computational modeling of the biological system, such as single-neuron rhythm generators. This work forges stronger connections with neuroscience computational models of CPG and astrocyte regulated synaptic plasticity to develop an algorithmic learning framework for gait emergence in legged robotic locomotion. The key distinguishing factors of our work against prior proposals are:

**(i) Neuroscience Inspired CPG Formulation:** Based on the bio-inspiration premise that CPG formulations need to have stronger neuroscience correlation in order to harness the efficiency of biological systems, we design a CPG circuit that captures the temporal dynamics of the locomotion unit and inter-limb communication to a much higher degree of detail. As shown in the paper, the bio-inspiration route enables us to scale neuromorphic robotic control, for the first time, to quadruped systems.

**(ii) Astrocyte Regulated CPG Plasticity:** While the dynamics of structures mediating CPGs have been extensively studied [11], [12], rewiring or plasticity of CPGs remains relatively underexplored [13], [14], especially in terms of

Z. Han, and A. Sengupta are with the School of Electrical Engineering and Computer Science, The Pennsylvania State University, University Park, PA 16802, USA. E-mail: sengupta@psu.edu.

understanding the homeostatic role of astrocytes. This work illustrates that astrocytes, working in parallel with local learning rules like reward-modulated Spike Timing Dependent Plasticity (STDP), suppress over-activated motor neurons directly or indirectly, by creating inhibitory synapses. This enables the emergence of a gait that provides continuous and smooth locomotion, while allowing us to harness the benefits of local, event-driven learning capability of bio-inspired systems when implemented on neuromorphic chips - paving the pathway for edge processing in robotics [15].

Additionally, the proposed algorithm is neuromorphic hardware compatible. For instance, stochastic leaky-integrate-fire spiking neuron characteristics used herein can be emulated by magnetic tunneling junction spintronic devices [16]. Additionally, the astrocyte models considered in this work are all local models, which contain only local variables and communicate with only the neighboring components in the architecture. As a result, the proposed algorithm is both bio-inspired and compatible with existing neuromorphic hardware or can be implemented by simple extensions of current neuromorphic computing architectures [17].

## II. RELATED WORK AND MAIN CONTRIBUTIONS

In recent years, global policy optimization for robotic locomotion through reinforcement learning has been widely investigated [18]–[20], where a policy network receives a system state vector including the joint positions and velocities, and outputs the driving signal to the motors. The reinforcement learning alleviates the difficulty of tuning complex parameter sets. However, the policy network-driven locomotion usually requires continuous neural network inference for every control time step, which is computationally expensive. Further, training such systems is even more computationally challenging, thereby limiting their applicability for on-chip learning in edge robotic systems with resource constraints.

Another category of robotic locomotion algorithms involves CPG circuits organizing the collaboration among joints or limbs, where one CPG unit controls a single joint or multiple joints. Some literature has applied classical control methods, such as phase oscillators, for the CPG [21]–[23]. From the neuromorphic implementation standpoint, there has been some preliminary work aimed at constructing the CPG by using SNNs [8]–[10]. However, existing works on SNN CPG use over-simplified neuron models and network architectures. For example, the flexor and the extensor motor neuron pools are usually simplified into a network of two neurons, where the internal fine structure is neglected. The lack of inclusion of biological details in the CPG architecture has resulted in limited flexibility of the control system, thereby constraining their applicability to mostly simple robotic platforms like hexapod robot, in contrast to the more complex design space of real-world robotic locomotion control of more intensively researched models, such as quadruped robots. In this work, we forge stronger connections with theoretical neuroscience to develop a detailed bio-inspired CPG model and show that astrocyte control is instrumental to ensure optimal and stable gait emergence in robotic quadruped locomotion systems.

Local learning mediated gait emergence ensures compatibility of our proposed control system with neuromorphic hardware, thereby leading to the potential of enabling real-time, low-power on-chip learning.

Finally, we also note that while recent work has explored the role of astrocytes in neuromorphic computing, it primarily remains limited to self-repair functionalities [24]–[28] and more recently self-attention mechanisms [29], working memory [30], energy minimization in neural networks [31], structure learning [32], and so on.

## III. QUADRUPED LOCOMOTION CONTROL CIRCUIT

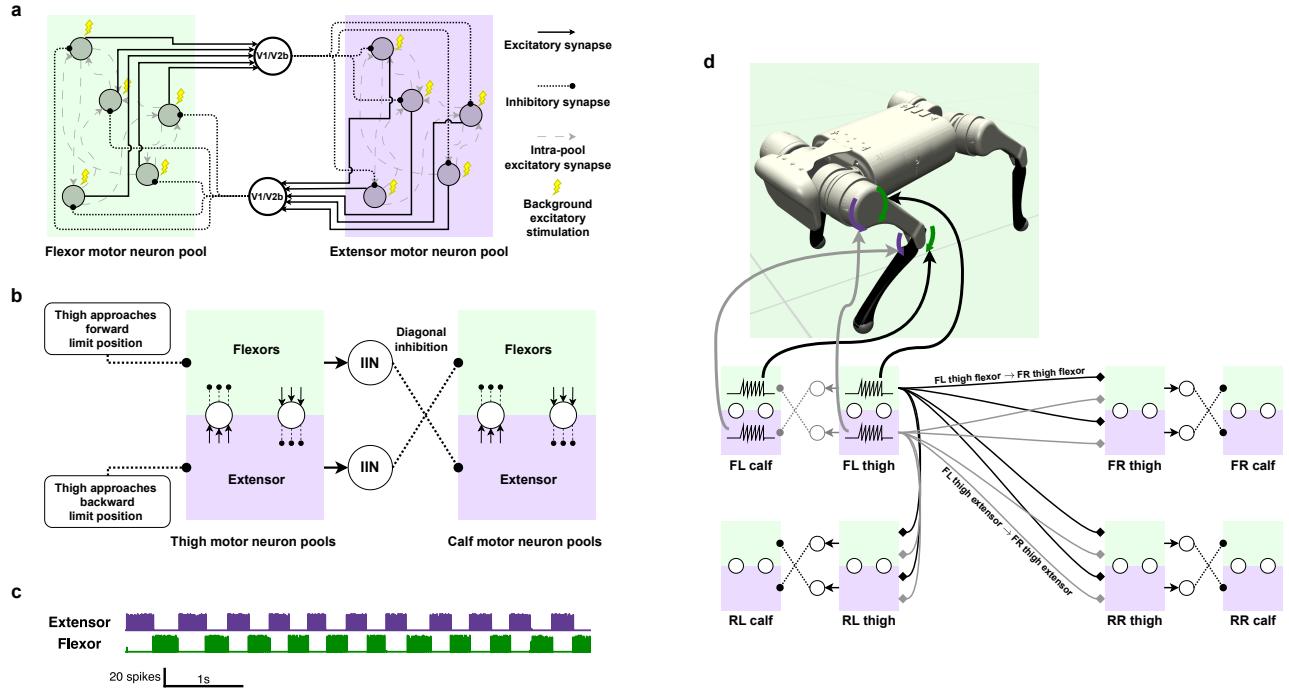
### A. Quadruped System Overview

The defining feature of a quadruped system (applicable to both biological systems and robots) is the set of four limbs. In the locomotion of a quadruped system, the contact between the ground and the feet of the limbs provides support to the torso and acceleration to the entire animal/robot. In case of a biological system, the lower motor neurons (located in the ventral horn of the spinal cord) directly control the tension of the joint flexors and extensors in the limbs [33]. The dynamics of the joint movement in the gait during locomotion, specifically the torque, angular position, angular velocity and angular acceleration of the joints are determined by the activity of the lower motor neuron themselves and the communication among the lower motor neurons pools; between the upper motor neuron (located in the brainstem) and the lower motor neuron; and from the Golgi tendon organ to the lower motor neurons [34], [35]. The gait locomotion functionality of a cat is still complete even when the cerebrums are removed, which indicates that the locomotion control system of four-limbed vertebrates is largely a local neural circuit in the brainstem and spinal cord [36]. Deriving insights from such neuroscience studies, we develop a bio-inspired CPG controller that models the collaboration among the limbs of a quadruped system. The framework is evaluated for a quadruped robot in a multi-body interaction physics simulation environment. In this paper, the Unitree A1 quadruped robot [37] system is adopted. The dynamics of the robot is simulated in the MuJoCo platform [38].

Figure 1 shows the proposed CPG architecture for robotic locomotion, which can be summarized below. A locomotion unit for a single joint contains 2 motor neuron pools: the extensor and the flexor. Through internal connections, the extensor and flexor motor neurons can generate alternating bursts. The extensor and flexor activities are converted into the direction and magnitude of the joint actuator torque of the corresponding joint. Through STDP learning in the inter-limb connection modulated by the astrocyte and reward signal, thigh locomotion limbs inhibit or excite each other, through which the robot's gait emerges.

### B. Central Pattern Generator

Critical behaviors of the gait locomotion of four-limbed animals are: (i) the alternating activity of the flexor and extensor muscles in the limbs [35], [39] and (ii) the phase difference among the limb movements. The alternating excitation of the



**Fig. 1. SNN-based CPG architecture.** **a**, The reciprocal inhibition of flexor and extensor motor neuron pools, where the flexor and extensor motor neuron pools are connected through V1/V2b inhibitory interneurons. For clarity, only 5 motor neurons for each pool are displayed. The symbols for synapses in the following sub-figures are consistent with those in sub-figure **a**. **b**, The thigh muscle inhibits its antagonistic muscle in the calf through an inhibitory interneuron (IIN). **c**, A sample snapshot of spontaneous rhythmic burst pattern output from the reciprocal inhibition structure, under non-feedback condition. The vertical axis represents the number of motor neurons spiking at each time step. **d**, The motor neuron pool of the thigh sends synapses (solid lines with diamond ends) to the motor neuron pool of the thigh in all other limbs. It is worth noting here that these synapses could be either excitatory or inhibitory. For clarity, only the synapses from the front-left limb are displayed. FR: front-right, FL: front-left, RR: rear-right, RL: rear-left.

two muscles of a single joint allows the joint to swing forward and backward periodically, which further converts the limb movement between the swing and stance phases [40]. In the past decade, research has elucidated the alternating excitation mechanism of the flexor and extensor motor neuron pools - reciprocal inhibition [41]. Past research has divided the entire CPG for gait locomotion into locomotion units that individually control the movement of a single limb [35]. Within the locomotion units, the motor neurons to an individual muscle inhibit or excite the motor neurons to other muscles [35]. The phase difference among the limbs usually corresponds to the specific locomotion speed of the animal [41]. Based on the phase difference, the gaits are usually categorized into walking, trotting and bounding [41]. The following sections will explain (a) how a locomotion unit, controlling a single limb, is composed of reciprocally inhibiting spiking motor neuron groups and controls the joint movement, and (b) how the connections among all the locomotion units (i.e., inter-limb connection) determine the phase difference among the limbs.

**1) Locomotion Unit:** Related experiments illustrate that the flexor and extensor muscles of a single joint always excite in opposite phases, i.e., when the flexor muscle bursts, the extensor muscle does not fire, and vice versa [35]. Also, the duration of the burst of both the flexor and extensor is approximately constant, which indicates that an active muscle will spontaneously stop firing after a specific duration of time

[35]. Such a burst pattern raises two critical questions: (1) *How does the motor neuron burst terminate spontaneously after a specific duration of time?* (2) *How does the halt of one muscle's activity trigger the activity of the antagonistic muscle?* We derive qualitative inspiration from neuroscience to answer these questions. The continuous firing of a motor neuron will accumulate  $\text{Ca}^{2+}$  in the cytoplasm, which later opens the  $\text{K}^+$  channels [35]. The decrement in the spike frequency of one muscle's motor neuron pool leads to the decline of inhibition to the antagonistic motor neuron pool. Based on this insight, in this paper, we assume that each motor neuron is receiving continuous background excitatory stimulation, and consequently, the flexor and the extensor reciprocally compete with each other, which finally forms periodic complementary burst patterns. The structure of such a reciprocal inhibiting SNN for a single joint is shown in Figure 1a. Two types of neurons are contained in this architecture: the motor neuron and the V1/V2b inhibitory interneuron. The motor neuron model is Pacemaker Stochastic Leaky-Integrate and Fire (PSLIF), which includes the cytoplasmic  $\text{Ca}^{2+}$  dynamics and the opening and closing of the  $\text{K}^+$  channel. The dynamics of the membrane potential of the motor

neuron  $i$  in the motor neuron pool can be formulated as:

$$\begin{aligned} \frac{dv_{\text{motor},i}(t)}{dt} &= \frac{(v_{\text{motor\_rest}} - v_{\text{motor},i}(t))}{\tau_{\text{motor}}} \\ &+ I_{\text{background}}(t, v) \cdot (1 + \text{Noise}(t)) \\ &- C_{K\_chan} \cdot \text{Sigmoid}(S_{K\_chan} \cdot ([\text{Ca}^{2+}] - \text{Thres}_{\text{Ca}^{2+}})) \\ &+ w_{V1/V2b \rightarrow \text{motor}} \cdot \delta(t - t_{V1/V2b\_spike}) \\ &+ \sum_{j \neq i} [w_{\text{motor},j \rightarrow i} \cdot \delta(t - t_{\text{motor\_spike},j})] \end{aligned} \quad (1)$$

Here,  $v_{\text{motor},i}$  is the membrane potential of neuron  $i$  in the pool.  $\tau_{\text{motor}}$  is the motor neuron membrane potential decay time constant.  $I_{\text{background}}(t, v) = I_{\text{background},0} + k_{\text{background},v} \cdot v_{\text{torso}}$  is the continuous background stimulation.  $I_{\text{background},0}$  is the strength of the background stimulation when the magnitude of the velocity of the torso is zero.  $k_{\text{background},v}$  is a positive background stimulation increment coefficient, and  $v_{\text{torso}}$  is the magnitude of the velocity of the torso. The velocity-related background stimulation will lead to an increasing frequency of the swing of the limbs when the robot is speeding up. The strength of the background stimulation also includes randomness. In the discrete implementation  $\text{Noise}(t) = A_{\text{random}} \cdot U_{[-1,1]}(t)$  where  $U_{[-1,1]}(t)$  is a sampling of a random variable whose distribution is Uniform $[-1, 1]$ . The random variable is sampled at every time step of the simulation.  $A_{\text{random}}$  is the amplitude of randomness.  $C_{K\_chan}$  is the potential drop rate when the  $K^+$  channel is fully on.  $\text{Sigmoid}(x) = \frac{1}{1+e^{-x}}$ .  $S_{K\_chan}$  is the sensitivity of the  $K^+$  channel to the  $\text{Ca}^{2+}$  concentration.  $[\text{Ca}^{2+}]$  is the cytoplasmic  $\text{Ca}^{2+}$  concentration.  $\text{Thres}_{\text{Ca}^{2+}}$  is the  $[\text{Ca}^{2+}]$  threshold for opening the  $K^+$  channel.  $w_{V1/V2b \rightarrow \text{motor}}$  is a negative synapse strength from the V1/V2b interneuron to the motor neurons, and this value is uniformly the same for all the motor neurons.  $w_{\text{motor},j \rightarrow i}$  is the intra-pool excitatory synapse strength from motor neuron  $j$  to motor neuron  $i$ .  $t_{V1/V2b\_spike}$  and  $t_{\text{motor\_spike},j}$  are the spike timings of V1/V2b interneuron and motor neuron  $j$  respectively. The cytoplasmic  $\text{Ca}^{2+}$  dynamics is:

$$\frac{d[\text{Ca}^{2+}]}{dt} = \frac{-[\text{Ca}^{2+}]}{\tau_{\text{Ca}^{2+}}} + r_{\text{Ca}^{2+}} \delta(t - t_{\text{spike}}) \quad (2)$$

where,  $[\text{Ca}^{2+}]$  is the concentration of cytoplasmic  $\text{Ca}^{2+}$ ,  $\tau_{\text{Ca}^{2+}}$  is the  $\text{Ca}^{2+}$  decay time constant,  $r_{\text{Ca}^{2+}}$  is the  $[\text{Ca}^{2+}]$  unit increment per action potential,  $t_{\text{spike}}$  is the time of firing of the motor neuron. Also, the stochasticity of PSLIF neuron is attributed to the randomness of the condition to trigger an action potential. The probability of triggering an action potential by the PSLIF neuron at any time step is:

$$p_{\text{spike}} = \text{Sigmoid}\left(\frac{v - v_{\text{th}}}{S_{\text{spike}}/2}\right) \quad (3)$$

where,  $v$  is the membrane potential,  $v_{\text{th}}$  is the threshold potential,  $S_{\text{spike}}$  is transition width of spike probability. The motivation for introducing stochasticity is to prevent the simultaneous spike of all the motor neurons in a motor neuron pool.

For each muscle, the corresponding motor neuron pool contains 20 inter-connected motor neurons. The synapses among

the motor neurons within a single pool are generated in a spatial manner: the location of each motor neuron is randomly generated in the 3-dimensional region  $[0, 1] \times [0, 1] \times [0, 1]$  from a uniform distribution and the synapses between a pair of motor neurons are generated according to the distance between two motor neurons, as shown below:

$$w_{\text{motor},j \rightarrow i} = w_{0\text{dist}} \cdot e^{-c_{\text{dist}} \cdot \text{dist}(j, i)} \quad (4)$$

where,  $w_{\text{motor},j \rightarrow i}$  is the synapse strength from motor neuron  $a$  to motor neuron  $b$ ,  $w_{0\text{dist}}$  is the synapse strength at zero distance,  $\text{dist}(j, i)$  is the spatial distance between motor neuron  $j$  and motor neuron  $i$ ,  $c_{\text{dist}}$  is the synapse strength spatial decay coefficient.

The inhibition is reported as a result of the activity of V1 and V2b inhibitory interneurons in the spinal cord [42]. Thus, in our SNN design, every motor neuron pool excites a V1/V2b interneuron, which in turn inhibits the motor neuron pool of the antagonistic muscle. The neuron model of V1/V2b interneuron is the Stochastic Leaky-Integrate and Fire (SLIF), where the stochasticity is generated in a similar fashion as in PSLIF. The rhythmic pattern generated by the reciprocal inhibition is shown in Figure 1c.

While previous work has proposed the possible connection of motor neurons in one locomotion unit [35], the mechanical structure of the limbs of the robot used in this paper is slightly different from a common quadruped animal, i.e., the number of joints responsible for locomotion in a single limb and the length ratio between the thigh and calf are different. Hence, the common quadruped locomotion unit architecture cannot be directly applied to the A1 robot. Therefore, we propose a locomotion unit SNN enabling the feet tips to move in a circular path (specific to the A1 robot). The diagram of the locomotion unit SNN is shown in Figure 1b. There is a second pair of motor neuron pools driving the flexor and extensor in the calf. Additionally, the thigh flexor inhibits the calf extensor, and vice versa. The neuron model of the inhibitory interneuron is SLIF, and they share the same model hyperparameters with the V1/V2b interneurons due to similar functionality. At the same time, the thigh flexor and extensors receive inhibitory sensory input when the thigh is approaching the limit position, which allows the transition between the swing and stance phases. For example, the angular range of the front thigh is  $[0.6, 1.4]$ . The front thigh flexor and the front extensor motor neurons will be inhibited when the thigh enters the regions  $[0.6, 0.65]$  and  $[1.35, 1.4]$  correspondingly. Given the diagonal inhibition from the motor neuron pools of the thigh to those of the calf, the phase transition of the calf is completely determined by the thigh motor neuron pools. Therefore the limit position inhibition for calf muscles is not necessary.

Regarding driving the motors at robot joints, an angular impulse with the corresponding sign is generated every time the flexor and the extensor for this joint spike. The motor torque trace accumulates the angular impulses and decays exponentially when there is no stimulus. In the A1 robot, we

use:

$$\frac{dh_{\text{thigh}}(t)}{dt} = -\frac{h_{\text{thigh}}(t)}{\tau_h} + I_{\text{thigh}} \cdot \delta(t - t_{\text{thigh\_extensor\_spike}}) - I_{\text{thigh}} \cdot \delta(t - t_{\text{thigh\_flexor\_spike}}) \quad (5)$$

where,  $h_{\text{thigh}}$  is the motor torque trace for the thigh joint,  $\tau_h$  is the decay time constant of the motor torque trace,  $I_{\text{thigh}}$  is the increment of torque for a single action potential for the thigh joint and  $t_{\text{thigh\_extensor\_spike}}$  is the time of firing of any individual neuron in the thigh extensor motor neuron pool (similar discussions are valid for the flexor). It should be noted that, in a discrete implementation of the SNN simulation, multiple motor neurons can possibly (actually they frequently do) fire at the same time step. In such a case, all the concurrent spikes in this time step contribute to the torque. This discussion is valid for all the following scenarios where the time of firing of a motor neuron pool is considered. In the MuJoCo configuration of the robot provided by the manufacturer, the joint motors of the A1 robot are working in torque mode. Consequently, the value of  $h_{\text{thigh}}$  is directly provided as output to the thigh motor actuator. The calf joint motor is driven by the same dynamics, but the angular impulse,  $I_{\text{calf}}$ , can be a different value.

2) **Inter-limb Connection:** Recent works have elaborated that there are excitatory and inhibitory synapses from the motor neurons for one limb to the motor neurons for another limb [35]. In this paper, the phase difference among the limbs is modulated by the inter-limb excitatory/inhibitory connections. As shown in Figure 1d, the thigh muscles from one limb send axons to all the muscles in all the other limbs. For example, the motor neurons of the thigh extensor in the front-left limb send connections to the thigh flexors and the extensors in front-right, rear-left and rear-right limbs. The calf motor neurons are not involved in the inter-limb connection. The source motor neuron pool and the target motor neuron pool are fully connected, but the entire connection shares a single synapse strength value. Such a connection could be expressed as a table of size 8 by 8, shown in Figure 2a, where the value in row  $i$  and column  $j$  represents the synapse weight of the connection projected from muscle  $i$  to muscle  $j$ . The inter-limb connection is trained by reward-modulated Spike Timing Dependent Plasticity (STDP), including the necessary modulation of astrocytes, which will be introduced in a later section. In the next subsection, the cytoplasmic signaling pathway will be introduced. The computational detail regarding the inhibitory effect of astrocyte output on inter-limb connection will be illustrated in Section IV.

### C. Astrocyte Regulation

The astrocyte is reported to play a homeostatic role in spinal cord locomotion control circuitry [3], [5], [6]. Specifically, the astrocytes, when activated, release signaling molecules that suppress over-excited neural groups to prevent energy waste. The spiking activity of neighboring neurons is a trigger of the cytoplasmic calcium activity of astrocytes. In general, first, specific receptors on the plasma membrane of the astrocyte

sense the product of the activated neighboring neurons, such as 2-arachidonyl glycerol (2-AG) or glutamate. Next, a sequence of interactions between  $\text{Ca}^{2+}$ , inositol trisphosphate (IP<sub>3</sub>) and IP<sub>3</sub> receptor responds to the activated membrane receptors. Finally, multiple types of signaling molecules are released from the astrocytes, such as glutamate or ATP, following the  $\text{Ca}^{2+}$  or IP<sub>3</sub> activity.

Astrocytes are diverse across the entire nervous system. For the locomotion control system, a previous work suggested that Endocannabinoid receptor CB<sub>1</sub> is expressed by the astrocytes in multiple brain regions and the dorsal horn of the spinal cord [43]. In this paper, we regard 2-AG, a type of endocannabinoid, as the effective neurotransmitter signaling from motor neurons to the astrocytes. The motor neurons release a constant amount of 2-AG every time there is an action potential. The concentration of 2-AG decays exponentially when no motor neuron fires:

$$\frac{d([\text{AG}])}{dt} = -\frac{[\text{AG}]}{\tau_{\text{AG}}} + r_{\text{AG}} \delta(t - t_{\text{motor\_spike}}) \quad (6)$$

where,  $[\text{AG}]$  is the concentration of 2-AG,  $\tau_{\text{AG}}$  is the decay time constant of 2-AG,  $r_{\text{AG}}$  is the production rate of 2-AG and  $t_{\text{motor\_spike}}$  is the time instant when the motor neuron spikes. 2-AG binds to CB<sub>1</sub> receptor on the astrocyte and then triggers the signaling pathway inside the astrocyte.

For the astrocyte-to-neuron signaling pathway, it is well established that ATP is the direct product of astrocytes that suppresses the locomotion activity [3], [4], [6]. Nevertheless, the mathematical model for the relationship between the output signals of the astrocyte and the  $\text{Ca}^{2+}$  and IP<sub>3</sub> activity in locomotion circuitry has not been detailed in prior works. Also, the detailed intra-astrocyte dynamics is absent in previous works on astrocyte-modulated locomotion. Instead, these works concentrate on the communication between astrocytes and the surrounding neurons [3], [6]. Therefore, a computational model for intra-astrocyte activity and ATP production needs to be adopted. A prior study [44] proposed a single-astrocyte model whose output is extracellular ATP. However, according to the paper, the evidence supporting the relationship between IP<sub>3</sub> and ATP release was not concrete [44]. Chen *et al.* also state that the calcium activity is believed to be the direct cause of the rise in extracellular ATP concentration [6]. Further, another work proposed a data-driven mathematical model of astrocyte, which, nevertheless, concentrates on the propagation of calcium activity in an astrocyte network [45]. As a result, adopting a customized model for describing the intra-astrocyte activity and the astrocyte-ATP relationship is inevitable. Here, in the absence of backing of in vivo experiments on astrocyte modulation in the spinal cord, we use the Li-Rinzel model for intra-astrocyte dynamics and use the unit ATP release model for the ATP production from astrocytes, where the ATP release directly depends on the astrocyte cytoplasmic  $\text{Ca}^{2+}$  activity. This is consistent with results reported in Refs. [3], [6] and at the same time minimizes the number of redundant assumptions.

The Li-Rinzel model [46], which describes the general cytoplasmic  $\text{Ca}^{2+}$ -IP<sub>3</sub> dynamics, is the basis of many following astrocyte dynamics models [47]. In the Li-Rinzel model,

the activation of CB<sub>1</sub> receptor results in IP<sub>3</sub> release in the astrocyte cytoplasm. IP<sub>3</sub> further regulates the release of Ca<sup>2+</sup> from the Endoplasmic Reticulum (ER) [24]. The transportation of Ca<sup>2+</sup> between the cytoplasm and ER can be divided into three components: the channel current from ER to the cytoplasm,  $J_{\text{chan}}$ ; the passive leakage current from the ER to the cytoplasm,  $J_{\text{leak}}$ ; the active pumping current from the cytoplasm to ER,  $J_{\text{pump}}$ . In summary:

$$\frac{d[\text{Ca}^{2+}]}{dt} = J_{\text{chan}} + J_{\text{leak}} - J_{\text{pump}} \quad (7)$$

The [Ca<sup>2+</sup>] here is the concentration of cytoplasmic Ca<sup>2+</sup>. Readers seeking a detailed computational model are directed to [24].

Chen *et al.* indicate that the extracellular ATP rises following the calcium activity in astrocytes [6]. The extracellular ATP, once released, is metabolized into adenosine immediately [3], [4], [6]. The adenosine plays an inhibitory role in the modulation of locomotion in the spinal cord [3], [6]. Multiple works have reported that the adenosine receptor on the membrane of the locomotion-related interneurons is a part of the inhibitory signaling pathway, but the specific type of receptor varies among different reports. Ref. [4] suggests that adenosine activates the A<sub>1</sub> receptor of the locomotion-related neurons in mice, while another report demonstrates that A<sub>2</sub> is the target of adenosine in swimming behavior modulation in zebrafish [6]. As ATP is always metabolized into adenosine rapidly [3], [4], [6], and the time for degeneration from ATP to adenosine can be neglected, we simplify the astrocyte-ATP-adenosine dynamics into astrocyte-adenosine dynamics for reducing the complexity of the computational model. The product of ATP metabolism, adenosine, is regarded as the direct output of the astrocyte. It is also assumed that one unit of adenosine  $r_{\text{ADO}}$  is released every time when the calcium concentration exceeds a threshold. Then, adenosine diffuses in the extracellular space and experiences exponential decay.

$$\frac{d[\text{ADO}]}{dt} = \frac{-[ \text{ADO} ]}{\tau_{\text{ADO}}} + r_{\text{ADO}} \delta(t - t_{\text{Ca}^{2+}, \text{ADO\_prev}}) \quad (8)$$

Here, [ADO] is the concentration of adenosine in the nearby region of the astrocyte,  $\tau_{\text{ADO}}$  is the adenosine decay time constant,  $r_{\text{ADO}}$  is the production rate of adenosine,  $t_{\text{Ca}^{2+}, \text{ADO\_prev}}$  is the time instant when [Ca<sup>2+</sup>] is above the threshold satisfying the condition that the time from the last adenosine release is not shorter than the adenosine refractory period. The adenosine received by the motor neurons negatively contributes to the synapse strength sent to this motor neuron pool. Figure 2b illustrates the negative feedback relationship between the motor neuron pool and the adenosine.

#### IV. GAIT LEARNING ALGORITHM

Previous works have discussed the reward-modulated STDP for encouraging actions through a delayed reward signal [48]. The basic idea of gait search is to reward the gait, i.e., the phase difference of the bursts among all the motor neuron pools, providing a larger forward speed and more stable torso attitude. The effect of different collaborations of the muscles is usually delayed, i.e., a good gait leads to a large forward

speed, but the speeding up procedure still costs a duration of time. Hence, reward-modulated STDP is necessary for gait search. The reward function is given by:

$$\begin{aligned} r(t) = & \alpha_{\text{vel}_x} \cdot \text{vel}_x(t) \\ & - \alpha_{\omega_{\text{roll}}} \cdot |\omega_{\text{roll}}(t)| \\ & - \alpha_{\omega_{\text{pitch}}} \cdot |\omega_{\text{pitch}}(t)| \\ & - \alpha_{\omega_{\text{yaw}}} \cdot |\omega_{\text{yaw}}(t)| \end{aligned} \quad (9)$$

where,  $\text{vel}_x$  is the velocity of the torso in  $x$  direction, which is the default heading direction at the beginning of the training process.  $\omega_{\text{roll}}$ ,  $\omega_{\text{pitch}}$  and  $\omega_{\text{yaw}}$  are the angular velocities in the roll, pitch and yaw directions. The parameters  $\alpha_{\text{vel}_x}$ ,  $\alpha_{\omega_{\text{roll}}}$ ,  $\alpha_{\omega_{\text{pitch}}}$  and  $\alpha_{\omega_{\text{yaw}}}$  are the reward coefficients. The reward-induced weight update from motor neuron pool  $x$  to motor neuron pool  $y$  in reward-modulated STDP used in this work can be written as:

$$\Delta w_{x,y}^{\text{reward}}(t) = \eta_{\text{eff}}(t) r_{\text{eff}}(t) \cdot \text{STDP}_{x,y}(t) \cdot \zeta_{w_{x,y}}(t) \quad (10)$$

where,

$$\begin{aligned} \frac{d \text{STDP}_{x,y}(t)}{dt} = & - \frac{\text{STDP}_{x,y}(t)}{\tau_{\text{STDP}}} \\ & + \delta(t - t_{y,\text{spike}}) \cdot u_x(t) \\ & - \eta_{\text{negative\_relative}} \cdot \delta(t - t_{x,\text{spike}}) \cdot u_y(t) \\ \frac{d u_x(t)}{dt} = & - \frac{u_x(t)}{\tau_{\text{trace}}} + \delta(t - t_{x,\text{spike}}) \\ \eta_{\text{eff}}(t) = & \eta \cdot \text{Progress}(t) \\ r_{\text{eff}}(t) = & r(t) - c_{\text{average}} \cdot \overline{r(t-0.1s,t)} \\ \zeta_{w_{x,y}}(t) = & \frac{[w_{\text{max}} - w_{x,y}(t)][w_{x,y}(t) - w_{\text{min}}]}{(w_{\text{max}} - w_{\text{min}})^2} \end{aligned} \quad (11)$$

where,  $\text{STDP}_{x,y}(t)$  is the STDP signal from motor neuron pool  $x$  to motor neuron pool  $y$ .  $\tau_{\text{STDP}}$  is the STDP signal decay time constant.  $t_{y,\text{spike}}$  and  $t_{x,\text{spike}}$  are the times of firing of motor neuron pool  $y$  and  $x$ .  $u_x(t)$  and  $u_y(t)$  are the spike traces of motor neuron pool  $x$  and  $y$  at time  $t$ .  $\tau_{\text{trace}}$  is the spike trace decay time constant.  $\eta_{\text{negative\_relative}}$  is the relative negative learning rate, which controls the contribution of negative STDP signal in learning.  $\eta_{\text{eff}}(t)$  is the effective learning rate, which is the product of training progress  $\text{Progress}(t)$  and learning rate  $\eta$ . The definition of training progress can be found in the Experiments section.  $r_{\text{eff}}(t)$  is the effective reward for STDP learning.  $\overline{r(t-0.1s,t)}$  is the sliding window average of the reward function from 0.1 seconds before to the current time.  $c_{\text{average}}$  is the averaged reward coefficient. By subtracting a fraction of  $\overline{r(t-0.1s,t)}$ , the system encourages the concurrent firing of the muscles which provides a higher reward than the average. It also discourages the concurrent firing of the muscles leading to the falling of the robot, at which time the angular velocities of the torso are large.  $\zeta_{w_{x,y}}(t)$  is a weight constraint factor preventing the weight from being out of the range of  $[w_{\text{min}}, w_{\text{max}}]$ .

It is worth noting here that the STDP learning algorithm is usually applied in situations where excitatory synapses play a significant role [49]. The learning rate for post-synaptic firing is usually much larger than its counterpart for pre-synaptic

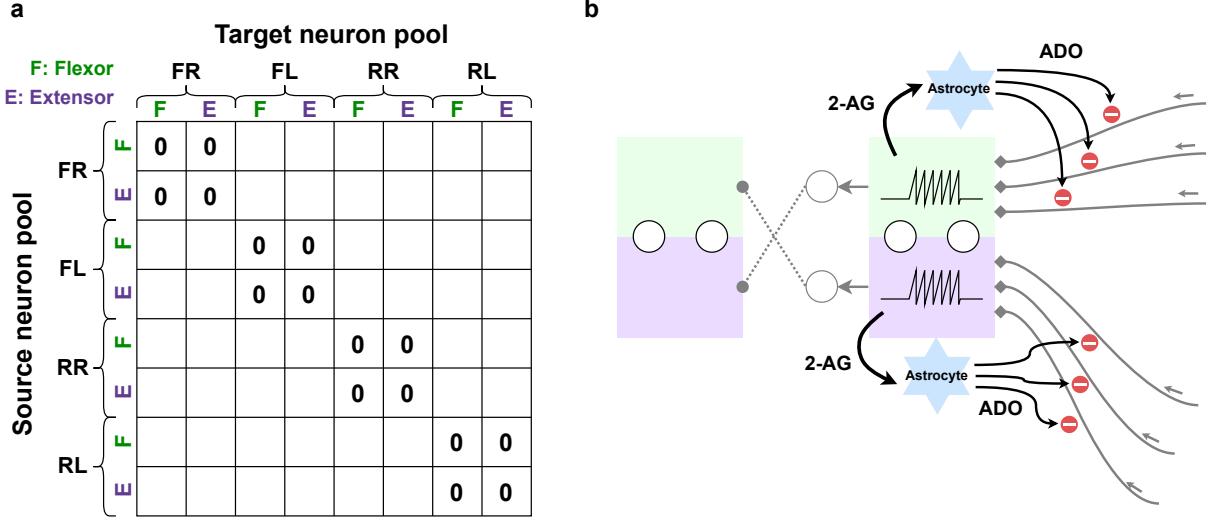


Fig. 2. **Astrocyte regulated gait search.** a, The table containing the synapse weight of inter-limb connections. As there is no self-limb connection, the 2 by 2 sub-tables on the diagonal are all zero. b, The astrocyte receives 2-AG from the motor neuron pool, which is later activated and releases adenosine that decreases the strength of the input synapses to this motor neuron pool. For clarity, only three of the input synapses are displayed.

firing. As a result, the learning of inhibitory synapses, which are critical in many neural systems like ours, is largely unexplored. According to the previous section, adenosine released by the astrocytes is sensed by the neighboring neurons. For simplifying the computational model, the detailed interaction between the adenosine and motor neurons is neglected. It is assumed that the adenosine generates an inhibitory effect on the input synapses of the motor neurons. i.e., the adenosine reduces the weight of the inter-limb synapses from other motor neurons to the corresponding motor neuron pool, as illustrated in Figure 2b. Therefore, at every time step, the astrocyte-induced weight update from motor neuron pool  $x$  to motor neuron pool  $y$  is:

$$\Delta w_{x,y}^{\text{astrocyte}}(t) = -\eta_{\text{ADO}} \cdot \text{Progress}(t) \cdot [\text{ADO}_y] \cdot \zeta_{w_{x,y}}(t) \quad (12)$$

where,  $\eta_{\text{ADO}}$  is the efficacy of adenosine in building inhibitory synapses,  $\text{Progress}(t)$  is the training progress,  $[\text{ADO}_y]$  is the adenosine concentration at motor neuron pool  $y$ ,  $\zeta_{w_{x,y}}(t)$  is the weight constraint factor. The complete weight update of the inter-limb connection is:

$$\Delta w_{x,y}(t) = \Delta w_{x,y}^{\text{reward}}(t) + \Delta w_{x,y}^{\text{astrocyte}}(t) \quad (13)$$

For discussion on the computational modeling of the proposed system, please refer to the section *Computational Implementation of the Dynamical System* in the supplementary material.

## V. EXPERIMENTS

### A. Problem Setup and Simulation Parameters

The experiments are performed using a computational simulation framework containing the quadruped robot physics simulation and the discrete simulation of the SNN-based CPG. MuJoCo (<https://github.com/deepmind/mujoco>) version 2.1.0

is adopted in this paper for the quadruped robot simulation. mujoco-py [50], version 2.1.2.14, is used to call the MuJoCo functions through Python APIs. The MJCF model of the A1 robot is obtained from the manufacturer's repository ([https://github.com/unitreerobotics/unitree\\_mujoco](https://github.com/unitreerobotics/unitree_mujoco)). The simulations are run on a bare metal computer with one Intel(R) Core(TM) i9-13900K CPU, 64GBytes RAM and the Ubuntu 22.04.2 LTS operating system. To simulate the dynamics of the SNN, i.e., the neuron/astrocyte models and the synaptic learning rules, we implemented the model dynamics in Python in a discrete manner, without utilizing any external SNN library. The simulation frequency is 1000 Hz. The entire codebase of the SNN implementation can be found at [https://github.com/NeuroCompLab-psu/Astrocyte\\_Regulated\\_Gait\\_Search.git](https://github.com/NeuroCompLab-psu/Astrocyte_Regulated_Gait_Search.git).

The information about the A1 robot model configuration for simulation, model initialization and control parameters such as the coefficients of the PI controller can be found in the supplementary material.

The structure of the algorithm in a single time step can be split into three distinct steps:

- 1) SNN and astrocyte model status are updated. Motor torque output is calculated from motor neuron activities.
- 2) MuJoCo simulates A1 model for one step
- 3) Reward is calculated from robot sensor data. Inter-limb connection weight is updated.

### B. Training Session and Training Progress

The MuJoCo environment inherently can simulate the robot system for an indefinitely long time. However, based on the locomotion control architecture mentioned before, the robot cannot reset its own status after a fall. Therefore, splitting the entire duration of the simulation into training sessions is necessary. In this paper, a simulation session will terminate when either the time reaches the maximum session length or the “non-alive counter” of the robot has a value higher

than a pre-defined threshold, which implies that the robot falls or the robot is stuck into a non-healthy attitude. The maximum session length is set to 10 seconds. The “non-alive counter” accumulator will increase by one for each time step the “alive indicator” is false. The “alive indicator” is a Boolean expression:

$$\text{ALIVE} = \text{Z\_AXIS}_{\text{torso}} \cdot (0, 0, 1) \geq \text{Thres}_{z\text{-ALIVE}} \quad (14)$$

where,  $\text{Z\_AXIS}_{\text{torso}}$  is the torso’s local Z-axis vector expressed in the world coordinate system.  $\text{Thres}_{z\text{-ALIVE}}$  is the threshold of the torso Z-axis vertical component. In an ideal still-standing attitude, the  $\text{Z\_AXIS}_{\text{torso}}$  has a value  $(0, 0, 1)$ , which is vertically upward.

As the reward-modulated STDP reinforces the synapse weight when concurrent firing provides a positive reward, the synapse weights will continuously change even when a good gait allows the robot to run smoothly. Therefore, the learning rate should be dependent on the learning progress, in order to halt the learning when a well-performing gait has been captured. The training progress is defined as:

$$\text{Progress}(t) = \text{Sigmoid}\left(-\frac{\frac{L_{[\text{curr}-10, \text{curr}-1]}}{L_{\text{max}}} - 0.9}{0.02}\right) \quad (15)$$

where,  $L_{[\text{curr}-10, \text{curr}-1]}$  is the average length of the previous 10 sessions.  $L_{\text{max}}$  is the maximum session length. Notice that the value of  $\text{Progress}(t)$  is in the range of  $[0, 1]$ . The training progress approaches zero when the average session length is approaching the maximum session length.

At the beginning of the entire training, the robot has a high possibility to fall to the ground. The training goal at the beginning, therefore, is learning not to fall. Progressing further, at the stage where the robot can run continuously for 7 to 8 seconds, the learning goal becomes maximizing the rewards. The beginning 2 seconds of the session, at this time, negatively contribute to the learning goal because of the zero resetting velocity. Consequently, we designed a dynamic learning period instead of entire-session learning. The starting time step of the learning period in any session is:

$$\begin{aligned} t_{\text{learning\_start}} \\ = \text{Clip}\left(\left(\frac{L_{[\text{curr}-10, \text{curr}-1]}}{L_{\text{max}}} - 1\right), 0, 2\right) \text{ seconds} \end{aligned} \quad (16)$$

### C. Gait Emergence and Ablation Study

The entire training process can be divided into multiple training sessions. For details about training sessions, please check the Methods section. At the beginning of the training, all the synapses of the inter-limb connection have zero values. The limbs hence swing back and forth individually with random and time-variant phase differences but similar frequencies. In the process of such random phase difference generation, multiple types of collaboration among the limbs will temporally emerge. For example, trotting, pacing or bounding can be seen for short time periods, typically much shorter than the maximum session length, at the beginning stages of learning. Based on our observations in the performed experiments, such random swings usually lead to a fall or non-continuous slow running. Among all the emerged gaits, the optimal gait

produces maximum  $x$ -directional speed and stable attitude of the torso. Nevertheless, the non-optimal gaits, generating slow locomotion, will also result in a positive reward, because of the large reward coefficient of the  $x$ -directional speed. The effect of the optimal and the non-optimal gaits all positively contribute to the inter-limb connections, through the STDP learning algorithm. Such always-positive learning causes saturated inter-limb synapse strengths at the pre-defined strength upper bound. Hence, there is a strong need to build inhibitory inter-limb connection weights, which is performed by the astrocytes through the release of adenosine to the motor neurons.

Figure 3 shows the homeostatic contribution of the astrocytes. Removing the astrocyte regulation implies that the second term on the right-hand side of Equation 13 is removed. In Figure 3a, the left panel shows the trained inter-limb connection weights without the astrocyte model. All the synapses are trained into excitatory connections with saturated weight  $w_{\text{max}}$ . In the right panel, it can be observed that the connection from one limb to another limb always contains two excitatory and two inhibitory synapses. For example, the connections from FR limb to FL limb have two inhibitory synapses in the diagonal positions and two excitatory connections in the anti-diagonal positions, which implies that the extensor of FR inhibits the extensor of FL (similar discussions are valid for the flexors). The extensor of FR excites the flexor of FL and vice versa. Such a combination produces a complementary firing pattern of the FR and the FL. In fact, the right panel of Figure 3a indicates a diagonal-synchronizing gait: FR-RL synchronized and FL-RR synchronized. The two groups are complementary to each other. This diagonal-synchronizing gait is usually called trotting. In summary, the astrocyte can filter out the effect of non-optimal gaits. Figure 3b more obviously illustrates the trotting gait. In the upper panel, the all-excitatory inter-limb connection brings unstable locomotion where the phase difference among the limbs is unstructured. Under current thigh and calf limit positions,  $I_{\text{background}}$ ,  $I_{\text{thigh}}$  and  $I_{\text{calf}}$ , only trotting gait will emerge. For a real quadruped animal, for example, a cat, the muscle tension and CPG rhythmic pattern frequency increase along with the descending modulating signal from the brainstem. Therefore, for the emergence of other gaits, the above-mentioned parameters need to be specifically tuned for our robotic application scenario. Furthermore, this trotting gait is currently only applicable to flat ground. Adaptability to more complex terrain will be the scope of future work. In Figure 3c, the average rewards in each session, the final  $x$  coordinate and the session lengths are plotted. The final  $x$  coordinate evaluates the distance the robot has moved as the reset heading direction is positive  $x$  direction. The reward and moving distance in the case without astrocyte modulation fluctuate around a constant and do not significantly improve after training. In contrast, including astrocyte regulation improves the average reward and locomotion speed, because of the emergence of the optimal gait. More specifically, the average  $x$ -directional speed of the last 20 sessions is 1.17 m/s, which is close to the trotting gate speed manually set in prior literature [20]. Such a speed emerges naturally, and it is the result of specific CPG burst duration and

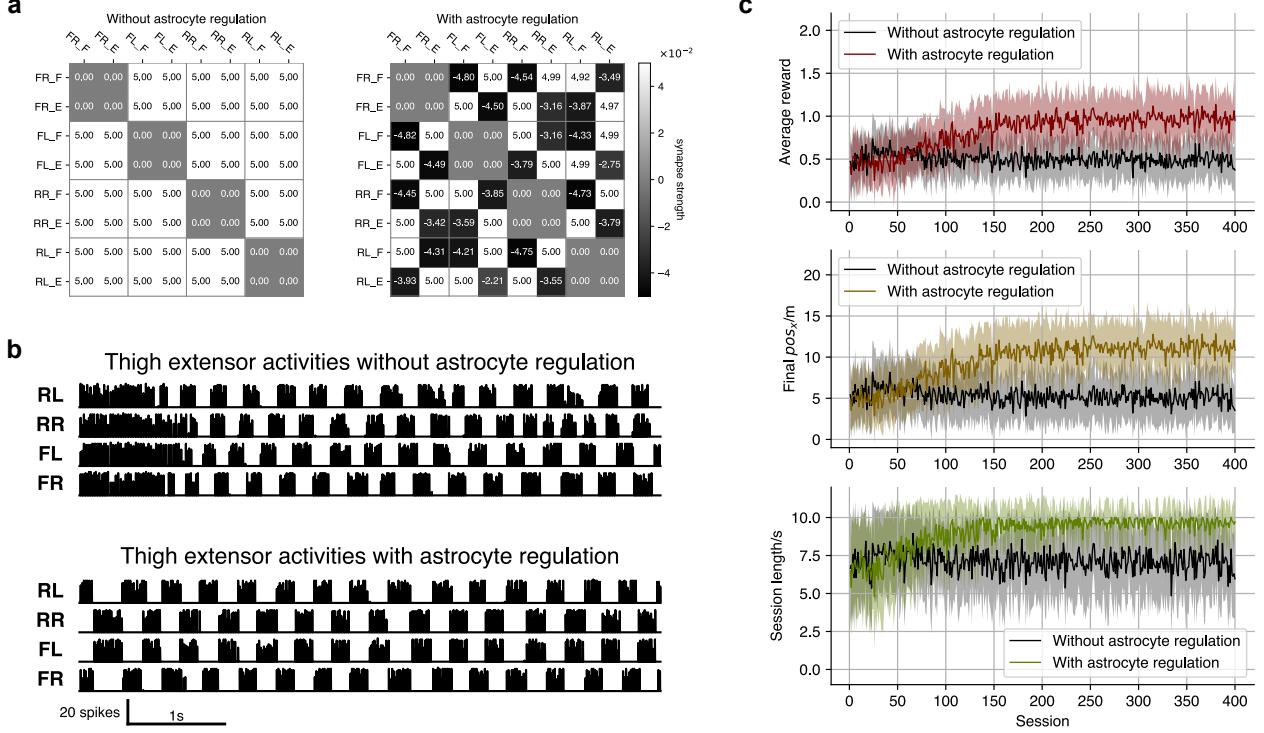


Fig. 3. **Ablation studies.** **a**, Typical trained inter-limb connection weights. **b**, Typical thigh extensor activities of the four limbs after training. Upper panel: limb collaboration is absent when astrocyte regulation is removed. Lower panel: diagonal-synchronization emerges under astrocyte regulation. **c**, The statistics of the average reward, the displacement in  $x$  direction at the final time step of each session, and session length in the training process, where the number of simulation trails is 20. The dark line represents the mean value, and the shaded region represents the interval of mean  $\pm$  standard deviation.

motor torque hyper-parameters. A session length shorter than 10 seconds implies a fall of the robot in the session. Involving astrocyte regulation significantly decreased the probability of falling during running.

#### D. Power Consumption Benefits

Recent literature on conventional deep learning approaches for quadruped robot gait learning primarily involves reinforcement learning, where the action at each control time step is output from a policy network. For example, [20] introduces an energy-related reward function in a reinforcement learning setting for quadruped (Unitree A1 robot) locomotion. The policy network is a 3-layer architecture where each hidden layer contains 128 neurons and the output layer contains 12 neurons, providing the target angular positions of the 12 joints in the limbs. The input to the policy network is a 42-dimensional state vector containing the current angular position and velocity of the joints in the limbs, the roll and pitch of the torso, the foot contact indicators and the action produced in the previous control time step. Therefore, the power consumption of the control system can be regarded as the product of the energy consumption of a single feed-forward inference computation and the control frequency. For the feed-forward inference, the number of multiplication operations of

a single layer  $j$  with the bias of an artificial neural network is:

$$\text{Num}_{j,\text{MULT}} = d_i \cdot d_o \quad (17)$$

where,  $d_i$  is the input dimension of this layer and  $d_o$  is the output dimension of this layer. The number of addition operations of layer  $j$  is:

$$\text{Num}_{j,\text{ADD}} = [(d_i - 1) + 1] \cdot d_o = d_i \cdot d_o \quad (18)$$

where  $[(d_i - 1) + 1]$  represents the number of additions, including bias, for a single output neuron. Therefore, the power consumption of the policy network based control system can be calculated as,

$$P_{\text{Policy}} = f_{\text{control}} \cdot \sum_{j=1,2,3} (\text{Num}_{j,\text{MULT}} \cdot \epsilon_{\text{MULT}} + \text{Num}_{j,\text{ADD}} \cdot \epsilon_{\text{ADD}}) \quad (19)$$

where,  $f_{\text{control}} = 100\text{Hz}$  is the control frequency [20],  $\epsilon_{\text{MULT}}$  and  $\epsilon_{\text{ADD}}$  are the energy consumption of a single multiplication and a single addition operation. In this paper, we consider the multiplication and addition operations are implemented

in 45nm CMOS process [51], where  $\epsilon_{\text{MULT}} = 3.7\text{pJ}$  and  $\epsilon_{\text{ADD}} = 0.9\text{pJ}$ . Consequently,

$$\begin{aligned} P_{\text{Policy}} &= 100\text{Hz} \cdot [(42 \cdot 128) + (128 \cdot 128) + (128 \cdot 12)] \\ &\quad \times (3.7\text{pJ} + 0.9\text{pJ}) \\ &= 1.07 \times 10^{-5} \text{ Watt} \end{aligned} \quad (20)$$

The SNN-based CPG proposed in this paper operates by means of transmitting and receiving spikes. Instead of the multiplication and addition operation that takes place for each synaptic connection in a non-spiking network, only an ADD operation takes place per synaptic operation in an SNN upon the receipt of an incoming spike [52]. The energy consumption of the SNN-based control system therefore can be calculated by counting the total number of accumulation operations. The discrete simulation of an SNN implies that the simulation frequency does not impact the firing frequencies of the spiking neurons. Thus, the power consumption of the SNN-based CPG is independent of the SNN simulation frequency, excluding the sensory input portion in this paper, which will be discussed in detail below. As elaborated in Section III, the neurons composing the CPG can be categorized into motor neurons and inhibitory interneurons, including the V1/V2b interneurons maintaining the reciprocal inhibition and the IINs between the thigh and the calf muscles in the same limb. These two types of inhibitory interneurons both individually inhibit 20 corresponding motor neurons, and hence they can be treated uniformly: one spike sent from an inhibitory interneuron results in 20 accumulation operations. One motor neuron in the calf flexor and extensor motor neuron pool excites 19 other motor neurons in the same motor neuron pool and excites a V1/V2b interneuron, which implies that one spike from a calf muscle motor neuron also leads to 20 accumulation operations. The case is slightly different for the thigh motor neurons. In addition to exciting 19 motor neurons in the same motor neuron pool and a V1/V2b interneuron, one thigh motor neuron also excites an IIN to inhibit the antagonistic muscle in the calf. Additionally, one thigh motor neuron in one limb sends 60 axons in total to the thigh motor neuron pools in all the other limbs. Thus, one spike from a thigh muscle motor neuron results in 81 accumulation operations. Also, when a thigh joint is approaching the limit positions, the 20 corresponding thigh muscle motor neurons receive limit position inhibitory spikes. The limit position spiking frequency here is the simulation frequency of the SNN, which is 1000Hz. Increasing the SNN simulation frequency will not influence the strength of limit position inhibition onto the motor neurons, but it will slightly raise the total power consumption. The total power consumption of the SNN-based controller is:

$$\begin{aligned} P_{\text{SNN-CPG}} &= (f_{\text{all\_inhi}} \times 20 \\ &\quad + f_{\text{all\_calf}} \times 20 \\ &\quad + f_{\text{all\_thigh}} \times 81 \\ &\quad + f_{\text{all\_limit\_posistion}} \times 20) \times \epsilon_{\text{ADD}} \end{aligned} \quad (21)$$

where,  $f_{\text{all\_inhi}}$ ,  $f_{\text{all\_calf}}$ ,  $f_{\text{all\_thigh}}$  and  $f_{\text{all\_limit\_posistion}}$  are the total firing frequencies of all the inhibitory interneurons, calf muscle motor neurons, thigh muscle motor neurons and limit

position inhibition. The total firing frequencies of the neurons are estimated by taking the mean of the session-wise average firing frequencies of the neurons. Specifically, the CPG runs for a session whose maximum session length is 10s to produce one session-wise average firing frequency. The estimated total firing frequency is the average of 10 session-wise average firing frequencies. The estimated average firing frequencies are  $f_{\text{all\_inhi}} = 6.69 \times 10^2\text{Hz}$ ,  $f_{\text{all\_calf}} = 4.34 \times 10^3\text{Hz}$ ,  $f_{\text{all\_thigh}} = 4.52 \times 10^3\text{Hz}$  and  $f_{\text{all\_limit\_posistion}} = 2.22 \times 10^3\text{Hz}$ . Consequently,  $P_{\text{SNN-CPG}} = 4.60 \times 10^{-7} \text{ Watt}$ , which is  $23.3 \times$  power efficient than the reinforcement learning based approach.

## VI. DISCUSSION

In summary, the work proposes a bio-inspired, SNN-based control architecture for quadruped robotic systems regulated by the homeostasis property of astrocytes. The neuromorphic CPG reveals the high energy efficiency of a spike-based locomotion control system. Also, the inhibitory astrocytic regulation causing the gait emergence underscores the significant role of modeling astrocytes for neuromorphic robotic control deployed in edge robotic applications.

Future work should consider further strengthening of the neuroscience connection in CPG modeling to develop adaptive and energy-efficient robotic locomotion. For instance, a more bio-plausible CPG modeling can consider the torque applied to a joint to have a complex relationship with the background stimulation. Multiple types of interneurons in the spinal cord, in fact, are involved in the neural circuit at different locomotion speeds [41], which could be a topic of further research. Descending command signals from the brainstem and sensory inputs are critical for animal locomotion. Also, the muscles in the torso are indispensable in multiple types of action. At first glance of animal locomotion, it seems that the legs are contributing most to the emergence of gaits. However, the limb muscles are only a fraction of the actuators of animal movement. The feed-forward plus feedback organization of all the muscles and the sensory apparatus as an entity is the true nature of animal locomotion and underlies the locomotion efficiency of biological systems.

## REFERENCES

- [1] S. L. Allam, V. S. Ghaderi, J.-M. C. Bouteiller, A. Legendre, N. Ambert, R. Greget, S. Bischoff, M. Baudry, and T. W. Berger, “A computational model to investigate astrocytic glutamate uptake influence on synaptic transmission and neuronal spiking,” *Frontiers in computational neuroscience*, vol. 6, p. 70, 2012.
- [2] J. D. Monaco, K. Rajan, and G. M. Hwang, “A brain basis of dynamical intelligence for ai and computational neuroscience,” *arXiv preprint arXiv:2105.07284*, 2021.
- [3] D. Acton and G. B. Miles, “Gliotransmission and adenosinergic modulation: insights from mammalian spinal motor networks,” *Journal of Neurophysiology*, vol. 118, no. 6, pp. 3311–3327, 2017.
- [4] D. Acton, M. J. Broadhead, and G. B. Miles, “Modulation of spinal motor networks by astrocyte-derived adenosine is dependent on d1-like dopamine receptor signaling,” *Journal of Neurophysiology*, vol. 120, no. 3, pp. 998–1009, 2018.
- [5] Y. Mu, D. V. Bennett, M. Rubinov, S. Narayan, C.-T. Yang, M. Tanimoto, B. D. Mensh, L. L. Looger, and M. B. Ahrens, “Glia accumulate evidence that actions are futile and suppress unsuccessful behavior,” *Cell*, vol. 178, no. 1, pp. 27–43, 2019.

[6] A. B. Chen, M. Duque, A. Rymbek, M. Dhanasekar, V. M. Wang, X. Mi, L. Tocquer, S. Narayan, E. M. Legorreta, M. Eddison *et al.*, “Norepinephrine changes behavioral state through astroglial purinergic signaling,” *Science*, vol. 388, no. 6748, pp. 769–775, 2025.

[7] A. Verkhratsky and M. Nedergaard, “Physiology of astroglia,” *Physiological reviews*, vol. 98, no. 1, pp. 239–389, 2018.

[8] 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.

[9] I. Polykretis, G. Tang, and K. P. Michmizos, “An astrocyte-modulated neuromorphic central pattern generator for hexapod robot locomotion on intel’s loihi,” in *International Conference on Neuromorphic Systems 2020*, 2020, pp. 1–9.

[10] 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, 2020.

[11] S. Grillner and A. Kozlov, “The cpgs for limbed locomotion—facts and fiction,” *International Journal of Molecular Sciences*, vol. 22, no. 11, p. 5882, 2021.

[12] J. Ausborn, N. A. Shevtsova, and S. M. Danner, “Computational modeling of spinal locomotor circuitry in the age of molecular genetics,” *International Journal of Molecular Sciences*, vol. 22, no. 13, p. 6835, 2021.

[13] L. Righetti, J. Buchli, and A. J. Ijspeert, “Dynamic hebbian learning in adaptive frequency oscillators,” *Physica D: Nonlinear Phenomena*, vol. 216, no. 2, pp. 269–281, 2006.

[14] M. Jouaiti, L. Caron, and P. Hénaff, “Hebbian plasticity in CPG controllers facilitates self-synchronization for human-robot handshaking,” *Frontiers in neuorobotics*, vol. 12, p. 29, 2018.

[15] Y. Sandamirskaya, “Rethinking computing hardware for robots,” *Science Robotics*, vol. 7, no. 67, p. eabq3909, 2022.

[16] A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, and K. Roy, “Magnetic tunnel junction mimics stochastic cortical spiking neurons,” *Scientific reports*, vol. 6, no. 1, p. 30039, 2016.

[17] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, “Advancing neuromorphic computing with loihi: A survey of results and outlook,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 911–934, 2021.

[18] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, “Learning to walk via deep reinforcement learning,” *arXiv preprint arXiv:1812.11103*, 2018.

[19] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, “Sim-to-real: Learning agile locomotion for quadruped robots,” *arXiv preprint arXiv:1804.10332*, 2018.

[20] Z. Fu, A. Kumar, J. Malik, and D. Pathak, “Minimizing energy consumption leads to the emergence of gaits in legged robots,” *arXiv preprint arXiv:2111.01674*, 2021.

[21] M. A. Lewis and G. A. Bekey, “Gait adaptation in a quadruped robot,” *Autonomous robots*, vol. 12, pp. 301–312, 2002.

[22] C. Liu, Y. Chen, J. Zhang, and Q. Chen, “Cpg driven locomotion control of quadruped robot,” in *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009, pp. 2368–2373.

[23] J. Zhang, F. Gao, X. Han, X. Chen, and X. Han, “Trot gait design and cpg method for a quadruped robot,” *Journal of Bionic Engineering*, vol. 11, no. 1, pp. 18–25, 2014.

[24] J. Wade, L. J. McDaid, J. Harkin, V. Crunelli, and S. Kelso, “Self-repair in a bidirectionally coupled astrocyte-neuron (an) system based on retrograde signaling,” *Frontiers in computational neuroscience*, p. 76, 2012.

[25] J. Liu, J. Harkin, L. P. Maguire, L. J. McDaid, and J. J. Wade, “Spanner: A self-repairing spiking neural network hardware architecture,” *IEEE transactions on neural networks and learning systems*, vol. 29, no. 4, pp. 1287–1300, 2017.

[26] J. Liu, L. J. McDaid, J. Harkin, S. Karim, A. P. Johnson, A. G. Millard, J. Hilder, D. M. Halliday, A. M. Tyrrell, and J. Timmis, “Exploring self-repair in a coupled spiking astrocyte neural network,” *IEEE transactions on neural networks and learning systems*, vol. 30, no. 3, pp. 865–875, 2018.

[27] M. Rastogi, S. Lu, N. Islam, and A. Sengupta, “On the self-repair role of astrocytes in stdp enabled unsupervised snns,” *Frontiers in Neuroscience*, vol. 14, p. 603796, 2021.

[28] Z. Han, A. N. Islam, and A. Sengupta, “Astromorphic self-repair of neuromorphic hardware systems,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, 2023, pp. 7821–7829.

[29] L. Kozachkov, K. V. Kastanenka, and D. Krotov, “Building transformers from neurons and astrocytes,” *bioRxiv*, pp. 2022–10, 2022.

[30] Y. Tsybina, I. Kastalskiy, M. Krivonosov, A. Zaikin, V. Kazantsev, A. N. Gorban, and S. Gordleeva, “Astrocytes mediate analogous memory in a multi-layer neuron–astrocyte network,” *Neural Computing and Applications*, vol. 34, no. 11, pp. 9147–9160, 2022.

[31] A. Fountain and C. Merkel, “Effect of biologically-motivated energy constraints on liquid state machine dynamics and classification performance,” *Neuromorphic Computing and Engineering*, vol. 2, no. 2, p. 024005, 2022.

[32] M. Han, L. Pan, and X. Liu, “Astronet: When astrocyte meets artificial neural network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 20258–20268.

[33] A. W. Monster and H. Chan, “Isometric force production by motor units of extensor digitorum communis muscle in man,” *Journal of neurophysiology*, vol. 40, no. 6, pp. 1432–1443, 1977.

[34] Dale Purves, George J. Augustine, David Fitzpatrick, William C. Hall, Anthony-Samuel LaMantia, Richard D. Mooney, Michael L. Platt, and Leonard E. White, Eds., *Neuroscience*, 6th ed. Oxford University Press, 2018. [Online]. Available: <https://global.oup.com/ushe/product/neuroscience-9781605353807?q=9781605353807&lang=en&cc=us>

[35] S. Grillner and A. El Manira, “Current principles of motor control, with special reference to vertebrate locomotion,” *Physiological reviews*, 2019.

[36] P. J. Whelan, “Control of locomotion in the decerebrate cat,” *Progress in neurobiology*, vol. 49, no. 5, pp. 481–515, 1996.

[37] X. Wang, “Unitree a1,” <https://www.unitree.com/>, 2020.

[38] E. Todorov, T. Erez, and Y. Tassa, “Mujoco: A physics engine for model-based control,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[39] S. Grillner and P. Zanger, “How detailed is the central pattern generation for locomotion?” *Brain research*, vol. 88, no. 2, pp. 367–371, 1975.

[40] J. J. Collins and I. N. Stewart, “Coupled nonlinear oscillators and the symmetries of animal gaits,” *Journal of Nonlinear science*, vol. 3, pp. 349–392, 1993.

[41] O. Kiehn, “Decoding the organization of spinal circuits that control locomotion,” *Nature Reviews Neuroscience*, vol. 17, no. 4, pp. 224–238, 2016.

[42] J. Zhang, G. M. Lanuza, O. Britz, Z. Wang, V. C. Siembab, Y. Zhang, T. Velasquez, F. J. Alvarez, E. Frank, and M. Goulding, “V1 and v2b interneurons secure the alternating flexor-extensor motor activity mice require for limbed locomotion,” *Neuron*, vol. 82, no. 1, pp. 138–150, 2014.

[43] C. Salio, S. Doly, J. Fischer, M. F. Franzoni, and M. Conrath, “Neuronal and astrocytic localization of the cannabinoid receptor-1 in the dorsal horn of the rat spinal cord,” *Neuroscience letters*, vol. 329, no. 1, pp. 13–16, 2002.

[44] W. G. Gibson, L. Farnell, and M. R. Bennett, “A computational model relating changes in cerebral blood volume to synaptic activity in neurons,” *Neurocomputing*, vol. 70, no. 10–12, pp. 1674–1679, 2007.

[45] T. Höfer, L. Venance, and C. Giaume, “Control and plasticity of intercellular calcium waves in astrocytes: a modeling approach,” *Journal of Neuroscience*, vol. 22, no. 12, pp. 4850–4859, 2002.

[46] Y.-X. Li and J. Rinzel, “Equations for insp3 receptor-mediated [Ca<sup>2+</sup>]i oscillations derived from a detailed kinetic model: a Hodgkin-Huxley like formalism,” *Journal of theoretical Biology*, vol. 166, no. 4, pp. 461–473, 1994.

[47] T. Manninen, R. Havela, and M.-L. Linne, “Computational models for calcium-mediated astrocyte functions,” *Frontiers in computational neuroscience*, vol. 12, p. 14, 2018.

[48] E. M. Izhikevich, “Solving the distal reward problem through linkage of stdp and dopamine signaling,” *Cerebral cortex*, vol. 17, no. 10, pp. 2443–2452, 2007.

[49] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.

[50] A. Ray, B. McGrew, J. Schneider, J. Ho, P. Welinder, W. Zaremba, and J. Tworek, “mujoco-py,” <https://github.com/openai/mujoco-py>, 2021.

[51] S. Han, J. Pool, J. Tran, and W. Dally, “Learning both weights and connections for efficient neural network,” *Advances in neural information processing systems*, vol. 28, 2015.

[52] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, “Going deeper in spiking neural networks: VGG and residual architectures,” *Frontiers in neuroscience*, vol. 13, 2019.

## ACKNOWLEDGMENTS

We are thankful to T. Iwasaki from the Department of Mechanical and Aerospace Engineering, UCLA and R. Dzakpasu from the Department of Physics, Georgetown University for their valuable suggestions regarding the work.

Research was sponsored primarily by the Army Research Office and was accomplished under Grant Number W911NF-24-1-0127. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. This work was also supported partially by the National Science Foundation under award No. EFRI BRAID #2318101.



**Zhuangyu Han** received the BEng from Sun Yat-sen University, Guangzhou, China, and Hong Kong Polytechnic University, Hong Kong, China, in 2017. He completed his Master's degree in Computer Science at Washington University in St. Louis in the Summer of 2020. Following graduation, he worked as a Software Engineer at Microsoft, Redmond, Washington before joining the Ph.D. program in Electrical Engineering at The Pennsylvania State University, University Park, PA, USA in Fall 2021. Zhuangyu is a Graduate Research Assistant in the Neuromorphic Computing Lab at Penn State and his research focus is on neuromorphic algorithms and astromorphic computing.



**Abhronil Sengupta** received the B.E. degree from Jadavpur University, Kolkata, India, in 2013, and the Ph.D. degree in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA in 2018.

He worked as a DAAD (German Academic Exchange Service) Fellow at the University of Hamburg, Hamburg, Germany in 2012, and as a graduate research intern at Circuit Research Labs, Intel Labs, Hillsboro, OR, USA in 2016 and Facebook Reality Labs, Redmond, WA, USA in 2017. In 2018, he joined the School of Electrical Engineering and Computer Science at The Pennsylvania State University, University Park, PA, USA where he is an Associate Professor and holds the Joseph R. and Janice M. Monkowski Career Development Professorship. He is also affiliated with the Department of Materials Science and Engineering and the Materials Research Institute (MRI). The ultimate goal of Dr. Sengupta's research is to bridge the gap between Nanoelectronics, Neuroscience and Machine Learning. He is pursuing an inter-disciplinary research agenda at the intersection of hardware and software across the stack of sensors, devices, circuits, systems and algorithms for enabling low-power event-driven cognitive intelligence. Dr. Sengupta has published over 100 articles in referred journals and conferences and holds 3 US patents.

He has been awarded the National Science Foundation CAREER Award, Army Research Office Early Career Award, Purdue Engineering 38 by 38 Award, IEEE Electron Devices Society (EDS) Early Career Award, IEEE Circuits and Systems Society (CASS) Outstanding Young Author Award, Facebook Faculty Award, IEEE SIPS Best Paper Award, Schmidt Science Fellows Award nominee, Purdue Bilsland Dissertation Fellowship, CSPIN Student Presenter Award, Birck Fellowship from Purdue University, and the DAAD WISE Fellowship. He serves on the IEEE Circuits and Systems Society Technical Committee on Neural Systems and Applications, Editorial Board of IEEE Transactions on Cognitive and Developmental Systems, Scientific Reports, Neuromorphic Computing and Engineering, Frontiers in Neuroscience journals and the Technical Program Committee of several international conferences. Dr. Sengupta is a Senior Member of the IEEE, Senior Member of the Association for Computing Machinery (ACM), and member of the American Physical Society (APS) and the Association for the Advancement of Artificial Intelligence (AAAI). He currently serves as an ACM Distinguished Speaker (2024-2027) and IEEE CASS Distinguished Lecturer (2025-2026).

## SUPPLEMENTARY MATERIAL

## Robot Simulation Model and Control Parameters

A single limb of the A1 robot contains three joints which are called hip, thigh and calf, which can be noticed in Figure 1d. The torque outputs from the central pattern generator are directly connected to the thigh and calf joint motor actuators. The terrain is flat ground. The robot's initial heading direction is the positive  $x$  direction. The thigh and calf joint limit positions are modified from the manufacturer's setting for better gait formation, and the hip joint limit position is not changed. The position range for the front and rear thigh joints are correspondingly  $[pos_{thigh,front}^{limit,-}, pos_{thigh,front}^{limit,+}] = [0.6, 1.4]$  and  $[pos_{thigh,rear}^{limit,-}, pos_{thigh,rear}^{limit,+}] = [0.7, 1.5]$ . The position range for the calf joint is  $[pos_{calf}^{limit,-}, pos_{calf}^{limit,+}] = [-1.6, -1.0]$ , all position values are in unit rad. As described in the Locomotion Unit section in the main text, the inhibition region for the thigh joint is the region within 0.05 from the limit positions. The "frictionloss" attribute of the joints is also increased to prevent fast swing angular velocity. The "frictionloss" parameter for hip and calf joints is set to 10 while that for the thigh joint is set to 25. The hip joints allow the limbs to move medially or laterally. Therefore, they do not significantly contribute to the forward driving force onto the torso. Therefore, a simple PI controller is designed to stabilize the hip joints at a constant position, where  $K_P = 30$  and  $K_I = 10$ . The designed stable position of the hip joints is laterally 0.1, i.e. 0.1 rad away from the center line of the torso. The expression of the hip joint torque as a function of the current and cumulative position error therefore is

$$\begin{aligned} \text{torque}_{\text{hip}}(t) = & K_P \cdot [pos_{\text{hip}}^* - pos_{\text{hip}}(t)] \\ & + K_I \cdot \int_0^t [pos_{\text{hip}}^* - pos_{\text{hip}}(t)] dt \end{aligned} \quad (22)$$

where,  $pos_{\text{hip}}^*$  is the target position of hip joints, i.e. 0.1.  $pos_{\text{hip}}(t)$  is the position of hip joint at time  $t$ . The resetting torso altitude is modified from the default value 0.3 to 0.35 to avoid conflict between the feet and the ground.

Before the onset of each training session,  $t_{\text{learning\_start}}$  and  $\text{Progress}(t)$  are calculated. The hip PI controller, motor traces, MuJoCo system status and all the SNN neuron status are reset. The astrocyte status is not reset before the beginning of a new session since the  $\text{Ca}^{2+}$  dynamics is a slow fluctuating process that will not have an abrupt change because of the change of robot status and is a reflection of the average motor neuron activities over several seconds. The thigh and calf joint positions are reset to a weighted sum of the limit positions of the joints. Accurately:

$$\begin{aligned} pos_{\text{thigh,front}}^{\text{reset}} &= 0.7 \cdot pos_{\text{thigh,front}}^{\text{limit},-} + 0.3 \cdot pos_{\text{thigh,front}}^{\text{limit},+} \\ pos_{\text{rear}}^{\text{reset}} &= 0.7 \cdot pos_{\text{thigh,rear}}^{\text{limit},-} + 0.3 \cdot pos_{\text{thigh,rear}}^{\text{limit},+} \\ pos_{\text{calf}}^{\text{reset}} &= 0.7 \cdot pos_{\text{calf}}^{\text{limit},-} + 0.3 \cdot pos_{\text{calf}}^{\text{limit},+} \end{aligned} \quad (23)$$

## Computational Implementation of the Dynamical System

The STDP signal, astrocyte calcium concentration, neuron membrane potential, and the other system dynamics mentioned in this work are mostly described through a set of differential equations which have a common form. The finite difference method is adopted to implement the evolution of such dynamical systems. Figure 4 illustrates the derivation of an implementable model state updating method. For uniformity, the time constants in all the implemented dynamic systems are regarded as much greater than the simulation time step  $\Delta t$ .

**Differential equation in form:**

$$\frac{dx}{dt} = -\frac{x}{\tau_x} + f(t)$$

**Finite difference:**

$$\Delta x = -\frac{\Delta t}{\tau} x_t + f(t) \Delta t$$

**Reorganize:**

$$x_{t+\Delta t} = x_t \left(1 - \frac{\Delta t}{\tau_x}\right) + f(t) \Delta t$$

**When  $\Delta t \ll \tau_x$ :**

$$x_{t+\Delta t} = x_t \cdot e^{-\frac{\Delta t}{\tau_x}} + f(t) \Delta t$$

Fig. 4. Computational implementation of model dynamics. The order of the dynamics of the models used in the paper are not higher than one. Here,  $x$  represents the state variable of the system. The Euler method is adopted to simulate all the dynamics. A flowchart of derivation is shown here to illustrate the approximation used in the computation.

Parameter	Description	Value
<b>SNN-based CPG</b>		
<i>Synapse generation</i>		
$w_{0\text{dist}}$	Intra-pool zero distance synapse strength	4
$c_{\text{dist}}$	Synapse strength spatial decay coefficient	0.3
$w_{V1/V2b \rightarrow \text{motor}}$	Uniform synapse strength from V1/V2b to motor neuron	-50
$w_{\text{motor} \rightarrow V1/V2b}$	Uniform synapse strength from motor neuron to V1/V2b	2
<i>Motor neuron</i>		
$\mathbf{Size}_{\text{pool}}$	Number of motor neurons in a motor neuron pool	20
$v_{\text{motor\_rest}}$	Motor neuron resting and resetting potential	0 mV
$v_{\text{motor\_th}}$	Motor neuron threshold potential	10 mV
$\tau_{\text{motor}}$	Motor neuron membrane potential decay time constant	9 ms
$\text{refractory}_{\text{motor}}$	Motor neuron refractory period	5 ms
$S_{\text{motor\_spike}}$	Motor neuron stochastic firing transition width	0.2
$r_{\text{Ca}^{2+}}$	$[\text{Ca}^{2+}]$ unit increment per action potential	1
$\tau_{\text{Ca}^{2+}}$	$[\text{Ca}^{2+}]$ decay time constant	250 ms
$\text{Thres}_{\text{Ca}^{2+}}$	$[\text{Ca}^{2+}]$ threshold for opening the $\text{K}^+$ channel	10
$S_{\text{K\_chan}}$	$[\text{Ca}^{2+}]$ sensitivity in $\text{K}^+$ channel	10
$C_{\text{K\_chan}}$	$\text{K}^+$ channel induced potential drop rate	8000 mV/s
$C_{\text{limit\_position}}$	Limit position inhibitory potential drop rate	400 mV/s
$I_{\text{background},0}$	Zero-velocity background stimulation strength	1380 mV/s
$k_{\text{background},v}$	Velocity-dependent background stimulation increment coefficient	40 mV/(s·m/s)
$A_{\text{random}}$	Randomness amplitude of background stimulation	0.5
<i>V1/V2b</i>		
$v_{V1/V2b\_rest}$	V1/V2b resting and resetting potential	0 mV
$v_{V1/V2b\_th}$	V1/V2b threshold potential	10 mV
$\tau_{V1/V2b}$	V1/V2b membrane potential decay time constant	9 ms
$\text{refractory}_{V1/V2b}$	V1/V2b refractory period	3 ms
$S_{V1/V2b\_spike}$	V1/V2b stochastic firing transition width	0.2
<i>Torque output</i>		
$\tau_h$	Motor torque trace decay time constant	0.1 s
$I_{\text{thigh}}$	Torque increment for a single spike at thigh joint	0.7 N · m
$I_{\text{calf}}$	Torque increment for a single spike at calf joint	1.1 N · m
$\text{pos}_{\text{hip}}^*$	Hip joints' target position	lateral 0.1 rad
$K_P$	Hip controller P coefficient	30
$K_I$	Hip controller I coefficient	10

TABLE I  
PARAMETERS

Parameter	Description	Value
<b>Reward-modulated STDP</b>		
$[w_{\min}, w_{\max}]$	Weight constraint factor	$[-0.05, 0.05]$
$\tau_{\text{trace}}$	Motor neuron spike trace decay time constant	0.01 s
$\tau_{\text{STDP}}$	STDP signal decay time constant	2 s
$\eta_{\text{negative\_relative}}$	Relative negative learning rate in STDP signal	0.3
$\eta$	Learning rate	$5 \times 10^{-10}$
$c_{\text{average}}$	Averaged reward coefficient	0.5
$\alpha_{\omega_x}$	$x$ -directional velocity reward coefficient	1
$\alpha_{\omega_{\text{roll}}}$	Roll angular velocity reward coefficient	-0.1
$\alpha_{\omega_{\text{pitch}}}$	Pitch angular velocity reward coefficient	-0.1
$\alpha_{\omega_{\text{yaw}}}$	Yaw angular velocity reward coefficient	-0.1
<b>Astrocyte dynamics</b>		
$r_{\text{AG}}$	2-AG production rate	$1 \times 10^{-3}$
$\tau_{\text{AG}}$	[2-AG] decay time constant	1 s
Thres_CaADO	$[\text{Ca}^{2+}]$ threshold for adenosine release	0.3
$r_{\text{ADO}}$	Adenosine production rate	0.01
$\tau_{\text{AG}}$	Adenosine decay time constant	1 s
refractory_ADO	Adenosine release refractory period	0.3 s
$\eta_{\text{ADO}}$	Adenosine inhibition efficacy	$1.8 \times 10^{-5}$
<b>MuJoCo model</b>		
$f_{\text{simulation}}$	MuJoCo simulation frequency	1000 Hz
$L_{\text{max}}$	Maximum session length	10 s
$L_{\text{non\_alive\_threshold}}$	Non-alive counter threshold	0.5 s
Thres <sub>z</sub> _ALIVE	Alive torso vertical-ness threshold	0.5
$[\text{pos}_{\text{thigh,front}}^{\text{limit},-}, \text{pos}_{\text{thigh,front}}^{\text{limit},+}]$	Front thigh joint limit position	$[0.6, 1.4]$ rad
$[\text{pos}_{\text{thigh,rear}}^{\text{limit},-}, \text{pos}_{\text{thigh,rear}}^{\text{limit},+}]$	Rear thigh joint limit position	$[0.7, 1.5]$ rad
$[\text{pos}_{\text{calf}}^{\text{limit},-}, \text{pos}_{\text{calf}}^{\text{limit},+}]$	Calf joint limit position	$[-1.6, -1.0]$ rad
$Z_{\text{torso,initial}}$	Initial torso altitude	0.35 m
frictionloss <sub>thigh</sub>	Thigh joint frictionloss	25
frictionloss <sub>calf</sub>	Calf joint frictionloss	10
frictionloss <sub>hip</sub>	Hip joint frictionloss	10

TABLE II  
PARAMETERS CONT.