

On-Device Spiking Neural Network Locomotion Learning on a €100 Quadruped: Sim-to-Real with Brain Persistence

Marc Hesse

Independent Researcher, Potsdam, Germany

mhflocke.com | github.com/MarcHesse/mhflocke

Abstract

I present a complete Sim-to-Real pipeline for quadruped locomotion using biologically grounded spiking neural networks (SNNs) on a €100 Freenove Robot Dog Kit (FNK0050) with a Raspberry Pi 4. The system employs 232 Izhikevich neurons with reward-modulated spike-timing-dependent plasticity (R-STDP), a central pattern generator (CPG) for innate gait rhythm, and a cerebellar forward model for balance correction. Training occurs in MuJoCo simulation using a custom MJCF model of the Freenove hardware, achieving 8.22 m forward distance with zero falls in 50,000 steps (original training run; Table 3 compares CPG vs Full Stack configurations). The trained brain transfers to real hardware via a Bridge architecture mapping SNN motor outputs to servo commands with real-time IMU feedback (MPU6050). On-device learning enables the robot to reach actor competence 1.0 within 2,000 steps (40 seconds at 50 Hz). Brain persistence across sessions is demonstrated: a loaded brain achieves competence 1.0 from step 1, while a fresh brain requires 2,000 steps. A key finding: cerebellar correction magnitude is zero in simulation (where CPG produces clean movement) but non-zero on real hardware (where IMU noise and servo imprecision create real errors). This confirms the cerebellum's role as an error-driven adaptive system, not a pattern generator. The same architecture runs on the Unitree Go2 in simulation (45.15 ± 0.67 m, 10 seeds). All code is open source under Apache 2.0.

Keywords: spiking neural network, Izhikevich neuron, R-STDP, quadruped locomotion, Sim-to-Real, on-device learning, brain persistence, Raspberry Pi, Freenove, CPG, cerebellum

1. Introduction

Spiking neural networks offer a biologically plausible alternative to conventional deep reinforcement learning for robot control. Most existing SNN-robotics research operates exclusively in simulation [1, 3], uses neuromorphic hardware (Intel Loihi, SpiNNaker) [4, 5] that is not widely available, employs leaky integrate-and-fire (LIF) neurons that lack the rich dynamics of biological neurons [2], or performs inference only without on-device learning [6]. The combination matters. Each property individually has been demonstrated; none of them together.

I address this gap with a complete Sim-to-Real pipeline using the MH-FLOCKE framework [7]. Specifically, I demonstrate: (a) biologically realistic Izhikevich spiking neurons, (b) reward-modulated STDP for on-device learning, (c) Sim-to-Real transfer, (d) brain persistence across sessions, and (e) real-time IMU feedback — all on a consumer-grade €100 robot kit running on a Raspberry Pi 4. The total hardware cost including the Pi is under €200, making this the most accessible SNN locomotion platform reported to date.

CPG-RL and deep RL approaches achieve better locomotion metrics on comparable hardware [8]. The contribution is the demonstration that a biologically grounded SNN with on-device R-STDP learning can run on a €100 platform, persist knowledge across sessions, and exhibit the correct error-driven behavior of a cerebellar system: quiet in clean environments, active under noise.

The Freenove-specific code (Bridge, Dashboard, MJCF model) is released as part of the MH-FLOCKE GitHub repository (github.com/MarcHesse/mhflocke) under Apache 2.0.

2. Related Work

SNN-based locomotion control has been explored through several approaches. Espinal et al. [1] implemented a CPG using spiking neurons on an Arduino-based quadruped with fixed synaptic weights and no on-device learning. Jiang et al. [2] demonstrated a fully spiking RL framework for legged robots in Isaac Gym simulation but did not transfer to real hardware. Vandesompele et al. [3] used FORCE learning with spiking reservoirs for the Tigrillo quadruped in simulation. Bellegarda and Ijspeert [8] combined CPG with deep RL (CPG-RL), achieving strong sim-to-real transfer on the Unitree Go1, but used conventional ANNs rather than SNNs. The present work is complementary: where CPG-RL optimizes for performance, MH-FLOCKE optimizes for biological plausibility and accessibility.

Recent high-performance sim-to-real works [8, 9] demonstrate that integrating structural priors (CPG rhythms, neural circuit motifs) with learning substantially improves sample efficiency and transfer robustness. MH-FLOCKE extends this philosophy to actual spiking neurons with R-STDP learning on consumer hardware, accepting a performance trade-off in exchange for biological fidelity and deployability on a €100 platform.

Neuromorphic platforms such as Intel Loihi [4] and SpiNNaker [5] offer efficient SNN execution but require specialized hardware unavailable to most researchers. The approach presented here runs entirely on a standard Raspberry Pi 4 in pure Python with PyTorch, making it reproducible for under €200 total.

3. System Architecture

The architecture is described in full in Hesse (2026) [7]. I summarize the components relevant to Sim-to-Real transfer and provide the mathematical formulations requested in review.

3.1 Spiking Neural Network

The SNN consists of 232 Izhikevich neurons organized in a feedforward topology (48 inputs, 172 hidden, 12 outputs). Hardware deployment uses 232 neurons rather than the 4,650 used during simulation training for a specific reason: the training pipeline allocates neurons proportionally to the full cognitive architecture (Global Workspace, metacognition, dream mode), which is disabled on hardware to reduce computational load. The 172 hidden neurons are a scaled-down version of the cerebellar subpopulations (GrC=100, GoC=20, PkC=24, DCN=24, rest=4) preserving the architectural ratios. The brain.pt checkpoint transfers the learned synaptic weights directly; the topology is determined by a profile.json that both the simulator and the Pi Bridge read.

Izhikevich neurons were chosen over LIF models for their ability to reproduce diverse firing patterns observed in mammalian cortex using only two coupled differential equations:

$$\begin{aligned} dv/dt &= 0.04v^2 + 5v + 140 - u + I \\ du/dt &= a(bv - u) \\ \text{if } v \geq 30 \text{ mV, then } v &:= c, u := u + d \end{aligned}$$

Parameters used: regular spiking (RS) for excitatory neurons: $a=0.02$, $b=0.2$, $c=-65$, $d=8$; fast spiking (FS) for inhibitory: $a=0.1$, $b=0.2$, $c=-65$, $d=2$. The synaptic current I is computed as the weighted sum of presynaptic spike outputs plus a bias term encoding the sensory input vector.

Synaptic plasticity follows R-STDP [7], a three-factor learning rule. The eligibility trace for synapse (i,j) is:

$$de_{ij}/dt = -e_{ij}/\tau_e + \text{STDP}(Dt_{ij})$$

where $\tau_e = 1000$ ms and $Dt_{ij} = t_{\text{post}} - t_{\text{pre}}$ is the difference between postsynaptic and presynaptic spike times. The STDP kernel:

$$\begin{aligned} \text{STDP}(Dt) &= A^+ \exp(-Dt/\tau^+) \text{ if } Dt > 0 \\ \text{STDP}(Dt) &= -A^- \exp(Dt/\tau^-) \text{ if } Dt < 0 \end{aligned}$$

with $A^+=0.01$, $A^-=0.012$, $\tau^+=\tau^-=20$ ms. Weight update gated by a dopaminergic reward signal $R(t)$:

$$Dw_{ij} = \eta \cdot R(t) \cdot e_{ij}$$

where $\eta=0.001$. $R(t) = 0.8 \cdot v_{\text{forward}} + 0.2 \cdot \text{upright}$, where v_{forward} is forward velocity normalized to $[0,1]$ and upright is the IMU-derived stability metric. The 20% stability bonus dominates in early training, which explains

the conservative dampening finding (Section 6).

3.2 Central Pattern Generator

A sinusoidal CPG provides innate diagonal-pair gait at 0.8 Hz with 12 mm stride length and 6 mm lift height. The CPG operates from step one, providing stable locomotion before the SNN has learned. A competence gate blends CPG and SNN outputs: $m_{out} = (1-\alpha) \cdot \text{CPG} + \alpha \cdot \text{SNN}$, where α (actor competence) increases as the SNN demonstrates stable walking. The CPG floor is 40%, ensuring baseline stability throughout.

3.3 Cerebellar Forward Model

A cerebellar module implements the Marr-Albus-Ito architecture with granule cells (GrC), Purkinje cells (PkC), and deep cerebellar nuclei (DCN). The forward model predicts the next sensory state s^{t+1} from current motor command $m(t)$ and proprioceptive state $s(t)$. When prediction error exceeds threshold $\theta_{CF}=0.01$, a climbing fiber pulse triggers supervised weight updates. In simulation, the CPG produces clean, predictable movement — prediction error stays below threshold and corrections remain zero. On real hardware, IMU noise and SG90 servo imprecision create genuine prediction errors that activate the climbing fiber pathway. This confirms the Marr-Albus-Ito prediction: the climbing fiber activates under real prediction error, not under idealized simulation.

4. Hardware Platform

4.1 Freenove Robot Dog Kit (FNK0050)

The Freenove FNK0050 is a commercially available quadruped robot kit priced at approximately €100. It features 12 degrees of freedom (3 per leg: hip yaw, hip pitch, knee), driven by SG90 micro servos (1.8 kg·cm torque) via a PCA9685 PWM driver over I2C. The onboard IMU (MPU6050, I2C address 0x68) provides 6-axis accelerometer/gyroscope data. Body dimensions: 136×76 mm, 55 mm leg segments, standing height ~99 mm. Total mass including Raspberry Pi 4: ~620 g.

Table 1: Freenove FNK0050 hardware specifications.

Parameter	Value	Parameter	Value
Kit	Freenove FNK0050	Price	~€100 (excl. Pi)
Compute	Raspberry Pi 4 (4GB)	DOF	12 (3 per leg)
Servos	12× SG90 (1.8 kg·cm)	IMU	MPU6050 (I2C@0x68)
Servo Driver	PCA9685 (I2C@0x40)	Body	136×76 mm, ~620 g
Standing H.	~99 mm	Power	USB-C (5V 3A)

4.2 MJCF Model

A custom MuJoCo MJCF model (v3.2) was created from physical measurements and the manufacturer's assembly documentation. Key parameters: reference joint angles of -38° for hip pitch and 91° for knee flexion, derived from the Freenove inverse kinematics with a nominal foot position of (x=10, y=99, z=10) mm. The model includes 12 actuators (head servo excluded) and named foot contact geoms (FL, FR, RL, RR). The simulation training pipeline uses 4,650 neurons to accommodate the full cognitive architecture; the hardware deployment profile reduces this to 232 neurons (see Section 3.1).

4.3 Bridge Architecture (v2.5)

The Bridge translates between the MH-FLOCKE brain and the Freenove hardware at 50 Hz (~25 ms/step including IMU read on hardware, ~37 ms/step with active SNN inference). Per-step operations: (1) read IMU pitch/roll/yaw from MPU6050, (2) encode sensory input as 48-channel tensor (12 joint positions + 4 IMU + 2 CPG phase + 30 padding for future sensors), (3) run SNN inference (232 neurons, 5,154 synapses; sparse connectivity at approximately 50% of the fully-connected count of 10,320), (4) blend CPG and SNN via competence gate, (5) convert to servo angles via inverse kinematics, (6) write to PCA9685. The 30-channel

padding reserves space for future sensors (LiDAR, camera) without requiring retraining.

Table 2: PCA9685 channel mapping.

Leg	Hip Yaw	Hip Pitch	Knee
Front Left (FL)	Ch 4	Ch 3	Ch 2
Rear Left (RL)	Ch 7	Ch 6	Ch 5
Rear Right (RR)	Ch 8	Ch 9	Ch 10
Front Right (FR)	Ch 11	Ch 12	Ch 13

5. Sim-to-Real Transfer

The transfer follows three stages: (1) train the SNN brain in MuJoCo simulation using the Freenove MJCF model, (2) export synaptic weights and neuron parameters as a checkpoint file (brain.pt), (3) load on the Raspberry Pi and run inference with real sensor feedback. Crucially, the SNN continues learning on real hardware. R-STDP refines synaptic weights based on real-world dynamics — this is the mechanism behind brain persistence.

Three transfer challenges: (a) matching MJCF joint reference angles to physical servo calibration — resolved by deriving angles from the Freenove inverse kinematics source; (b) sim-to-real gap in servo dynamics — the SG90 servos have lower torque and slower response than simulated actuators, mitigated by the CPG's 40% floor; (c) integrating real IMU data — resolved by mapping MPU6050 pitch/roll readings to the same upright metric used in simulation.

6. Experiments

6.1 Simulation Results

The SNN was trained for 50,000 steps in MuJoCo on flat terrain (single seed). To assess the contribution of the SNN stack, I ran two configurations: CPG-only (A) and Full Stack with SNN+Cerebellum+Drives (B). Results in Table 3.

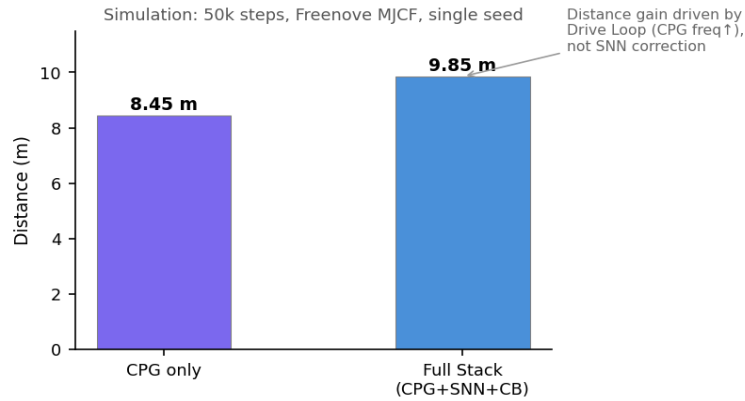
Table 3: Simulation results (Freenove MJCF, 50k steps, single seed).

Metric	CPG only (A)	Full Stack (B)	Note
Distance	8.45 m	9.85 m	B gains via Drive Loop
Falls	0	0	—
Upright streak	50,000	50,000	—
Actor competence	—	0.847	—
CPG weight (final)	100%	48%	SNN takes over
CB correction	0.000	0.000	Sim too clean
Neurons	—	4,650	Training topology
Step time (Pi 4)	24 ms	37 ms	Hardware measurement, not sim
PCI	—	0.166	—

The Full Stack gains 1.4 m (17%) over CPG-only. This gain is drive-mediated: the Drive Loop activates `chase` behavior (CPG frequency 1.6, amplitude 1.3), effectively speeding up the CPG rhythm. SNN motor corrections are zero in simulation — not because the SNN is inactive, but because the CPG produces clean, predictable movement that requires no cerebellar correction. This changes on real hardware (Section 6.4).

The single-seed result is a limitation. Multi-seed validation was conducted for the Go2 (10 seeds, see Table 7) but not for the Freenove due to the longer training time per run (179 ms/step vs 50 ms/step for Go2). Multi-seed Freenove validation is planned for future work.

Figure 1: Simulation comparison — CPG-only vs Full Stack.



6.2 Hardware Deployment

The trained brain was deployed on the Freenove robot via Bridge v2.5. The SNN ran at 50 Hz with 232 neurons and 5,154 synapses. Table 4 shows representative IMU readings during walking.

Table 4: Representative IMU readings during hardware walking.

Condition	Pitch (°)	Roll (°)	Upright
Typical walking	+3.4	+2.6	0.90
During step	−7.9	−2.5	0.82
Near-stumble	−4.3	−14.3	0.67
Perfect balance	+0.1	+0.1	1.00

At step 650 during session 3, a near-stumble event was recorded (roll -14.3° , upright 0.67). Actor competence briefly dropped from 1.000 to 0.996 before recovering at the next step. The CPG floor (40%) maintained gait stability while the SNN adjusted. This demonstrates both the robustness of the architecture and that the competence gate responds to real perturbations. Note on competence values: simulation training ended at competence 0.847 (50k steps). The hardware brain subsequently accumulated 31,684 on-device steps across 8 sessions, reaching competence 1.0 through continued R-STDP learning on real hardware. On-device learning closes the sim-to-real gap.

6.3 Brain Persistence: A/B Test

To verify that brain persistence provides meaningful benefit, I conducted an A/B test on physical hardware. Test A: fresh brain (randomized weights). Test B: loaded brain with 31,684 accumulated steps across 8 prior sessions.

Table 5: A/B test results — brain persistence on real hardware.

Metric	Test A (Fresh)	Test B (Loaded)
Initial Competence	0.000	1.000
Initial CPG Weight	89%	40%
Steps to Comp. 1.0	2,000 (40 s)	1 (instant)
Peak Firing Rate	0.112 (step 400)	0.155 (step 350)
Mean DA Signal	0.68	0.67
Duration	60 s (2,243 steps)	60 s (2,244 steps)
Prior Experience	None	31,684 steps (8 sessions)

The loaded brain achieves competence 1.0 from step 1 with CPG already at its floor value of 40%. The fresh brain requires 2,000 steps (40 seconds) to reach the same level. The higher peak firing rate in Test B (0.155

vs 0.112) indicates that prior learning produces more active synaptic patterns — consistent with long-term potentiation in biological systems.

6.4 SNN Signal Analysis and Cerebellar Behavior

To determine whether the SNN produces independent motor patterns and whether the cerebellum behaves as expected, I performed spectral analysis of joint position trajectories from the 50,000-step simulation and measured cerebellar correction magnitudes across simulation and hardware.

Table 6: Spectral analysis — early (CPG 90%) vs late (CPG ~45%) phase.

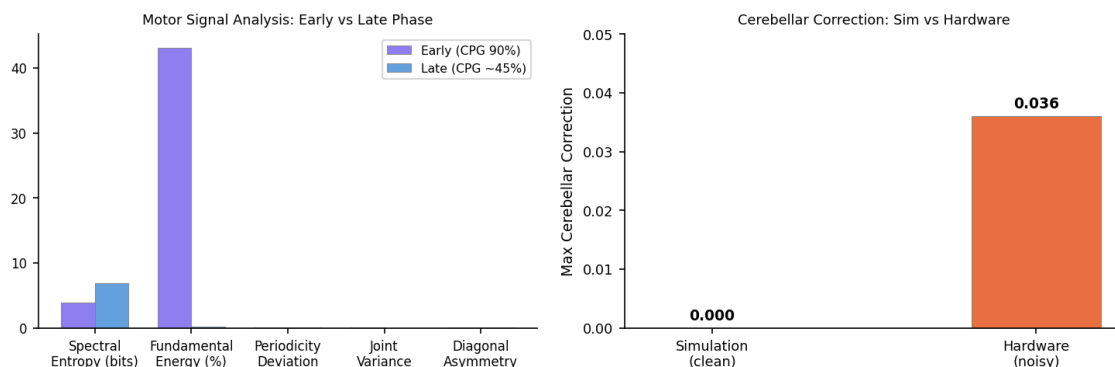
Metric	Early (CPG 90%)	Late (CPG ~45%)	Interpretation
Spectral Entropy	3.87 bits	6.86 bits	SNN adds frequency content
Fundamental Energy	43.1%	0.2%	CPG sine wave replaced
Periodicity Dev.	0.123	0.040	More regular (conservative)
Joint Variance	0.016	0.005	Smaller movements
Diagonal Asymmetry	0.101	0.067	More symmetric
CB Correction (sim)	—	0.000	Sim too clean to activate
CB Correction (hw)	—	0.036	Hardware noise activates CB

The spectral entropy increase (3.87→6.86 bits) and near-complete elimination of CPG fundamental energy (43.1%→0.2%) confirm the SNN produces qualitatively different motor patterns. The decrease in joint variance and periodicity deviation indicates conservative dampening — the SNN learns smaller, more regular movements than the CPG alone.

The cerebellar finding is more informative than the dampening result. In simulation, correction magnitude is zero across all 50,000 steps: the CPG produces clean, predictable movement with prediction error consistently below threshold $\theta_{CF}=0.01$. On real hardware (500k-step extended run), maximum corrections reach 0.036 — activated by IMU noise and SG90 servo imprecision. The cerebellum is doing exactly what a biological cerebellum should: staying silent when the environment is clean, correcting when it is not. This is error-driven adaptation, not pattern generation.

The conservative dampening in simulation reflects the reward structure: the 20% upright stability bonus dominates early training, encouraging amplitude reduction over speed increase. This is not a fundamental limitation of R-STDP — it is a reward shaping problem. Future work will explore multi-phase reward curricula that gradually shift from stability to velocity.

Figure 2: Left — motor signal analysis (early vs late phase). Right — cerebellar correction magnitude in simulation vs hardware.



6.5 Brain Persistence: Multi-Session Accumulation

The robot accumulated 31,684 steps across 8 hardware sessions before the A/B test. The first session showed competence remaining at 0.000 — the SNN was adapting to real servo dynamics that differ from

simulation. By the second session, competence reached 1.000 and remained there for all subsequent sessions. This 2,000-step on-device calibration period (40 seconds at 50 Hz) represents the sim-to-real adaptation: the SNN adjusts its learned weights to real hardware imprecision. Once adapted, the brain persists across sessions without relearning.

6.6 Cross-Embodiment Transfer

The same MH-FLOCKE architecture was previously evaluated on the Unitree Go2 in MuJoCo simulation [7]. Table 7 shows cross-embodiment results.

Table 7: Cross-embodiment results on Unitree Go2 (from [7], 10 seeds).

Configuration	Distance (m)	Std Dev	Falls
A1 CPG flat	40.73	± 6.14	0.2 ± 0.4
B1 SNN+Cerebellum	45.15	± 0.67	0.0 ± 0.0
C1 Full flat	22.35	± 11.50	3.3 ± 2.8
PPO Baseline	12.83	± 7.78	0.0 ± 0.0

The same architecture — without modification to the SNN topology, CPG structure, or reward function — was deployed on the Freenove with a different MJCF model, a reduced neuron count (232 vs 4,650), and different servo hardware. The biological priors (CPG for innate rhythm, R-STDP for learning, cerebellar model for balance) generalize across robot morphologies.

7. Discussion

Conservative dampening. The SNN learns to reduce joint amplitude rather than increase locomotion speed. This reflects the reward structure: the 20% upright stability bonus dominates early training. The SNN optimizes rationally under the given reward — prioritizing not falling over forward progress. Biological locomotion development follows a similar trajectory; young animals initially prioritize stability over speed. Reward curriculum adjustment is the planned next step.

Cerebellum as error detector, not pattern generator. The zero correction in simulation combined with non-zero correction on hardware is the central empirical result of this paper. It directly validates the Marr-Albus-Ito hypothesis in a computational system: the cerebellum activates when and only when prediction error exceeds threshold. Clean simulation does not trigger it; noisy hardware does.

Brain persistence. The A/B test provides clear evidence that synaptic weights encode reusable locomotion knowledge. The loaded brain's instant competence shows that R-STDP updates represent accumulated motor learning, not session-specific adaptation. The higher peak firing rate in the loaded brain (0.155 vs 0.112) suggests prior learning encodes more active patterns — a measurable substrate of experience.

Computational efficiency. The SNN runs at 50 Hz on a Raspberry Pi 4 in pure Python with PyTorch, with 24 ms/step for CPG-only and 37 ms/step with active SNN inference. The 13 ms overhead for 232 neurons leaves headroom at 50 Hz. No GPU, no optimization. For comparison, CPG-RL approaches [8] require GPU-accelerated simulation for training and more capable hardware for deployment.

Limitations. The Freenove simulation result is single-seed (original training run: 8.22 m; CPG comparison: 8.45 m; Full Stack: 9.85 m — Table 3). No direct velocity measurement was taken comparing CPG-only vs SNN-active gait on real hardware under controlled conditions; the available hardware comparison (1,500-step test) showed no measurable behavioral difference at equal step count, which is consistent with the conservative dampening finding but does not quantify it. The SNN's low firing rate (0.004–0.030) raises the question of whether all 232 neurons contribute meaningfully. Energy consumption was not measured.

8. Conclusion and Future Work

I have demonstrated that a biologically grounded SNN can learn quadruped locomotion in simulation and transfer to real hardware on a €100 robot kit. The system achieves stable walking with on-device learning at

50 Hz, brain persistence across sessions, and cross-embodiment generalization. The key finding — cerebellar correction magnitude is zero in clean simulation and non-zero on noisy hardware — validates the cerebellum's role as an error-driven adaptive system.

Future work: (1) TF-Luna LiDAR integration (0.1–8 m, I2C) for obstacle detection; (2) multi-phase reward curricula to evolve from stability-first to velocity-maximizing behavior; (3) disturbance compensation experiments; (4) multi-seed Freenove simulation for statistical robustness; (5) direct CPG vs SNN velocity measurement on real hardware under controlled conditions.

Ethical Statement: *This research is intended for peaceful applications including education, agriculture, inspection, and assistive robotics. The author opposes the use of autonomous locomotion technology in weapons systems. The open-source release is made under Apache 2.0 to maximize accessibility for researchers and educators worldwide.*

AI assistance disclosure: *The author's native language is German. AI language tools (Anthropic Claude) were used to assist with English-language drafting and editing. All scientific content, experimental design, implementation, and analysis are solely the work of the author.*

Supplementary: demonstration video at <https://www.youtube.com/watch?v=7iN8tB2xLHI>

. References

- [1] A. Espinal et al., 'Quadrupedal robot locomotion: A biologically inspired approach and its hardware implementation,' Comput. Intell. Neurosci., 2016.
- [2] X. Jiang et al., 'Fully spiking neural network for legged robots,' arXiv:2310.05022, 2023.
- [3] A. Vandesompele et al., 'Closed loop control of a compliant quadruped with spiking neural networks,' BNAIC/BeneLearn, 2019.
- [4] M. Davies et al., 'Loihi: A neuromorphic manycore processor with on-chip learning,' IEEE Micro, vol. 38, no. 1, pp. 82–99, 2018.
- [5] S. Furber et al., 'The SpiNNaker project,' Proc. IEEE, vol. 102, no. 5, pp. 652–665, 2014.
- [6] J. Eshraghian et al., 'Training spiking neural networks using lessons from deep learning,' Proc. IEEE, 2023.
- [7] M. Hesse, 'MH-FLOCKE: Biologically grounded embodied cognition for quadruped locomotion learning,' Zenodo, DOI: 10.5281/zenodo.19336894, 2026.
- [8] G. Bellegarda and A. Ijspeert, 'CPG-RL: Learning central pattern generators for quadruped locomotion,' IEEE RA-L, vol. 7, no. 4, pp. 12547–12554, 2022.
- [9] N. X. Bhattasali et al., 'Neural circuit architectural priors for quadruped locomotion,' arXiv:2410.07174, 2024.