Central Pattern Generators Locomotion Control for Low-Cost Quadruped Robots

Alex Pitoiset Department of Electrical Engineering, Eindhoven University of Technology

a.pitoiset@student.tue.nl

June 11, 2025

***Abstract*—This paper explores the implementation of biologically inspired central pattern generators (CPGs) to control the locomotion of a quadrupedal robot. The controller is based on coupled nonlinear Hopf oscillators that produce rhythmic and coordinated joint trajectories, making them suitable for real-time control on low-power hardware. A key feature of the Hopf model is its ability to independently modulate the descending and ascending phases of the oscillation cycle, enabling separate control of the swing and stance phase. The CPG was tested on the physical robot. The most stable locomotion was achieved with the trotting gait, where the robot can recover from small disturbances with a speed of 0.68 body lengths per second (BL/s). To further improve the system, a Pybullet simulation environment has been used to optimize the CPG parameters for forward velocity through a grid parameter space search. The optimized configuration reaches a speed of 1.94 BL/s with high stability. Ultimately, the results show that biologically inspired controllers can be applied to simple robotic platforms with great potential for education and low-cost robotic research.**

I. Introduction

Efficient locomotion is crucial for animal survival, allowing predator evasion, foraging, and mating. Over millions of years, natural selection has refined both the morphologies and neural control systems of animals, allowing them to navigate in complex and unpredictable conditions with notable efficiency [1].

In robotics, achieving the same level of adaptability is crucial for deploying legged machines in constantly changing real world environments. To bridge the gap between engineering and nature, engineers have frequently turned to biology for ideas, replicating animal morphologies. For example, snake-like robots for constrained environments or quadruped platforms for irregular terrain [2], [3]. These morphologies inherit the mechanical advantages of their biological counterparts. However, their control systems often lack in adaptability. For example, the Boston Dynamics quadrupedal robot, Spot, mimics animal legs. However, it relies on centralized and computationally intensive controllers that require multiple sensors [3]. Although this approach performs well, it remains complex and expensive, limiting its scalability for wider deployment and access.

To address these challenges, a promising approach to improve robotic locomotion is to use Central Pattern Generators (CPGs), which are neural circuits that generate high-dimensional rhythmic signals. In vertebrates, CPGs are located in the spinal cord and produce periodic coordinated movements such as walking and breathing [4]. By automating these repetitive physical patterns, CPGs allow the brain to redirect its resources to other cognitive tasks, such as selecting safe paths or responding to unexpected stimuli. These systems are triggered by simple inputs from higher brain centers. The brain can modulate their rhythmic output to control parameters such as locomotion speed and gait. Moreover, CPGs remain coupled to the body and environment through sensory feedback [1]. This decentralized architecture reduces the delays for commands and the required bandwidth while simplifying high-level control [5]. These features make CPG-based controllers an attractive solution as they are less energy intensive, simpler, and more affordable.

A. Related Work

Inspired by biological principles, various CPG models have been implemented in robotics. Among them, coupled nonlinear oscillators have emerged as biologically plausible and computationally efficient for generating rhythmic movement [5], [6].

The power of coupled nonlinear oscillators lies in their ability to inhibit each other. This coupling behavior enables synchronization between joints, producing coordinated gaits and smooth transitions between them [6]–[8]. Such mechanisms mimic biological inter-neuronal

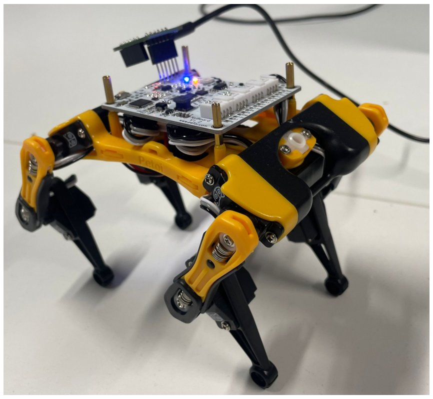


Figure 1: The Petoi Bittle robot used for implementing and testing the CPG-based controller.

inhibition and allow the generation of out-of-phase rhythmic signals. They also exhibit a limit-cycle behavior, producing periodic and self-sustaining oscillations that quickly recover from transient perturbations [7]. In addition, their compact dynamics, governed by a small set of parameters (e.g. frequency and amplitude), makes them appropriate for embedded systems with limited resources. Despite their promise, most existing implementations target high-end robotic platforms. Their applicability to compact, affordable systems remains underexplored, which underpins the present study.

B. *Contributions*

In this work, the main contributions are as follows:

- Development of a CPG-based locomotion controller using coupled nonlinear oscillators to generate rhythmic coordinated gait patterns (Sec. III-A).
- Integration of the controller on the Petoi Bittle, a compact and low-cost quadrupedal robot shown in Figure 1 (Sec. IV-A).
- Implementation of a PyBullet simulation environment for testing and refinement of the CPGs and mapping parameters (Sec. IV-B).
- Parameter space exploration in simulation to optimize forward velocity using grid-based search (Sec. IV-C).
- Discussion of current limitations and suggestions for future improvements (Sec. V).

II. Problem Statement

The Petoi Bittle quadruped robot should execute a stable forward walk without falling. The robot has four legs, each with two joints controlled by servomotors: a hip joint $v_i(t)$ and a knee joint $w_i(t)$ for leg $i = 1, \dots, 4$. At each discrete time step k, with fixed sampling interval Δt, the pose of the robot's base is described by its linear position $\mathbf{q}_k = (q_k^x, q_k^y, q_k^z)$ and its orientation in Euler angles $\boldsymbol{\gamma}_k = (\gamma_k^p, \gamma_k^r, \gamma_k^y)$, where q_k^x is the forward displacement along the x–axis, q_k^y and q_k^z represent lateral and vertical displacements, and γ_k^p, γ_k^r, and γ_k^y denote pitch, roll, and yaw angles, respectively. Joint trajectories $v_i(t)$, $w_i(t)$ are subject to two performance criteria. First, forward progression must meet a minimum target distance D_{target} over a time $T = \frac{N}{\Delta t}$, where N is the number of time steps, such that $q_T^x \geq D_{\text{target}}$. Simultaneously, the robot must maintain stability, which means that lateral and vertical deviations, as well as orientation errors, remain negligible: $|q_k^y| \approx 0$, $|q_k^z - H| \approx 0$, and $|\gamma_k^p| \approx 0$, $|\gamma_k^r| \approx 0$, $|\gamma_k^y| \approx 0$, where H is the nominal body height.

Second, the CPG must exhibit stable limit–cycle behavior so that transient perturbations are rapidly attenuated and do not lead to falls. A fall is declared if $q_k^z < h_{\min}$ or $|\gamma_k^r| > \gamma_{\max}$ at any time step, where $h_{\min}$ and $\gamma_{\max}$ are safety thresholds.

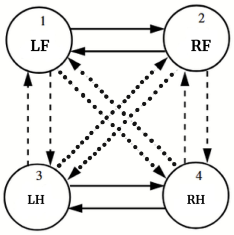
From a control perspective, the Hopf oscillator network should allow independent modulation of the descending (stance) and ascending (swing) phases, reflecting the asymmetric nature of animal locomotion [1]. In fact, in stable quadrupedal gaits, the stance phase typically occupies a longer duration than the swing phase. It should also allow smooth adjustment of amplitude and frequency via simple parameter changes and strong coupling with the robot's mechanics. The overall architecture must remain generic enough to accommodate different quadruped robots and to allow future integration of higher–level command signals.

III. Solution

A. *Central Pattern Generator Model*

The CPGs are modeled as a network of four mutually coupled Hopf oscillators, each driving one limb: left front (LF), right front (RF) left hind (LH) and right hind (RH). The coordination between limbs arises from inhibitory coupling. The coupling network is shown in Figure 2, where each oscillator modulates the phase of its neighbors to generate specific gaits [7], [8]. For instance, for the trot gait, oscillator 1 (LF) is excited by 4 (RH) $k_{14} = +1$, while 2 (RF) and 3 (LH) inhibit 1 (LF), as $k_{12} = -1$ and $k_{13} = -1$.

Each oscillator is governed by a modified Hopf model. It allows phase-dependent frequency modulation and enables independent control over stance and swing durations. The state of each oscillator (x_i, y_i) depends on



(a) Trot Pace Bound Net.

Figure 2: Coupling topology for the 4-cell CPG for trot, pace and bound. Arrows of the same type indicate identical coupling strengths and signs. The arrangement encodes the desired gait symmetries. These topologies are directly translated into the corresponding coupling matrices shown in Figure 3.

$$\begin{bmatrix} 0 & -1 & -1 & 1 \\ -1 & 0 & 1 & -1 \\ -1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{bmatrix}$$

(a) Trot

$$\begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ 1 & -1 & 0 & -1 \\ -1 & 1 & -1 & 0 \end{bmatrix}$$

(b) Pace

$$\begin{bmatrix} 0 & 1 & -1 & -1 \\ 1 & 0 & -1 & -1 \\ -1 & -1 & 0 & 1 \\ -1 & -1 & 1 & 0 \end{bmatrix}$$

(c) Bounce

$$\begin{bmatrix} 0 & -1 & 1 & -1 \\ -1 & 0 & -1 & 1 \\ -1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 0 \end{bmatrix}$$

(d) Walk

Figure 3: Coupling matrices for four quadruped gaits: (a) Trot, (b) Pace, (c) Bounce, and (d) Walk. Each matrix entry k_ij defines the influence of oscillator j on oscillator i. 1 is excitatory coupling, -1 is inhibitory coupling and 0 is no direct coupling

the following set of equations:

$$\dot{x}_i = \alpha\,(u - r_i^2)\,x_i - \omega_i\,y_i, \tag{1}$$

$$\dot{y}_i = \beta\,(u - r_i^2)\,y_i + \omega_i\,x_i + \gamma \sum_{j=1}^{4} k_{ij}\,y_j, \tag{2}$$

$$\omega_i = \frac{\omega_{\text{stance}}}{1 + e^{-b\,y_i}} + \frac{\omega_{\text{swing}}}{1 + e^{b\,y_i}}, \tag{3}$$

where $r_i = \sqrt{x_i^2 + y_i^2}$ is the instantaneous amplitude, u defines the squared amplitude of the limit cycle, and $\alpha, \beta > 0$ are convergence rates (typically $\beta \gg \alpha$ to enforce stronger convergence along the y-axis). The term ω_i transitions between the stance and swing frequencies depending on the phase y_i, and γ is the coupling gain added to have stronger coupling and less dependency on the initial conditions settings. The sigmoid functions in Equation 3 ensure smooth and continuous transition between the stance and swing frequencies. A sigmoid is an S-shaped mathematical function that smoothly interpolates between two values; in this context, it gradually shifts the oscillator's frequency based on the sign and magnitude of y_i. the parameter b sets the sharpness between this transition. This allows the oscillator to gradually change between the swing and the stance phases according to the sign and magnitude of y_i. The coupling matrix $K = [k_{ij}]$ is gait-specific.

Different gaits emerge from the choice of the coupling matrix. Figure 3 illustrates the matrices for trot, pace, bound, and walk, each defining a distinct set of stable phase relationships. Figure 4 shows the time evolution of the outputs from the Hopf CPG model and the joint angles for the walking gait simulation. The CPGs demonstrate the expected sinusoidal movement for each leg. Figure 5 shows the limit cycle behavior of the CPG, where the state variables x_i and y_i converge to a circular trajectory of radius $\sqrt{\mu}$ and closely follow this limit cycle over time. This demonstrates the oscillator's stability and periodicity, which are critical for generating smooth, consistent joint trajectories necessary for coordinated legged locomotion.

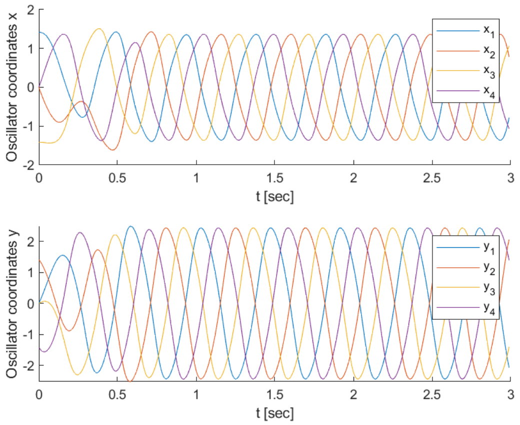


Figure 4: Time series x_i and y_i for a walking pattern.

B. Mapping to Joint Angles

The output from the Hopf CPG model, represented by the variables x_i and y_i, is mapped to the robot's joint angles, determining how the simulated CPG behavior translates into actual limb movement. Specifically, the hip and knee joint angles for each leg are computed as follows:

$$\theta_{\text{hip}} = \theta_{\text{hip base}} + \theta_{\text{hip range}} \cdot x_i \tag{4}$$

$$\theta_{\text{knee}} = \theta_{\text{knee base}} - \theta_{\text{knee range}} \cdot \max(0, y_i) \tag{5}$$

Here, x_i modulates the hip motion, corresponding to the forward-backward swing of the leg, while y_i controls knee flexion during the swing phase. Negative values of

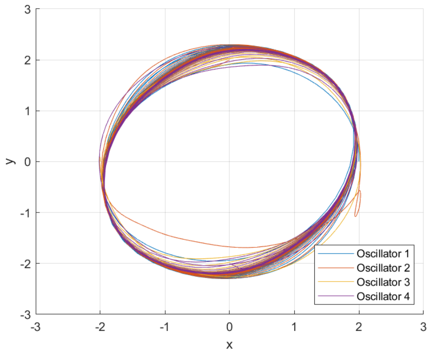


Figure 5: Limit cycle behavior of Hopf oscillators.

y_i are suppressed to prevent unnatural hyperextension. This mapping transforms the original circular limit cycle into a biologically inspired trajectory for the foot, more closely resembling the natural gait cycle observed in animals. It allows the robot to generate plausible leg trajectories that contribute to effective and natural locomotion.

IV. Experiments

A. The Robot

video results can be found at this link.

1) Physical Platform and Control Architecture

The quadruped robot used for testing is the Petoi Bittle, a compact and affordable robot dog designed for education and research [9]. It features a lightweight plastic chassis that supports four legs, each driven by two servo motors, one for the hip and one for the knee. Damping springs are integrated into each leg to absorb shocks during locomotion, adding some compliance. To improve grip and prevent slippage, the robot's feet feature custom 3D printed rubber shoes.

The Bittle is controlled via its onboard microcontroller, the NyBoard, which runs custom firmware capable of interpreting serial commands. Communication between the robot and a host computer is established via a USB interface. All CPG computations are performed externally on the host computer. The resulting target joint angles are transmitted to the Bittle as low-level servo commands through a serial communication protocol. This setup enables open-loop control of the robot joints, as no on-board sensors are used in the control loop. A Python API facilitates communication with the NyBoard, supporting both high-level posture commands (e.g., ”sit” or ”stand”) and low-level control over individual joint angles. For CPG-based locomotion, the latter is essential, as it allows each oscillator output to be directly mapped to a servo target position in real time.

2) Evaluation on the Physical Robot

Two gaits were tested on the physical Petoi Bittle robot, the walk and the trot. Each gait was generated using the same set of CPG and joint mapping parameters. These tests aimed at evaluating the effectiveness of open-loop CPG-based control.

The walk gait resulted in limited forward progression and showed reduced stability. In contrast, the trot gait achieved more consistent locomotion and covered a significantly greater distance per cycle. Measured over several trials, the trot achieved a performance of approximately 0.68 body lengths per second (BL/s). Furthermore, the trot gait demonstrated remarkable adaptability and robustness when tested on unpredictable terrains, including outdoor environments with debris and uneven surfaces. Despite these challenging conditions, the trot gait maintained stability and effective locomotion, showcasing its potential for real-world applications.

These experiments highlight the strong sensitivity of locomotion performance to the parameterization of the CPG and its associated joint mappings. Since the mapping function depends directly on the CPG's internal state variables, even minor changes to the Hopf oscillator parameters can lead to considerable variations in joint trajectories. Consequently, any adjustment to the CPG model requires a recalibration of the joint mapping parameters to maintain effective and stable locomotion.

B. Simulation environment

To evaluate the performance of the CPG model prior to deployment on the physical robot, a simulated environment was created using Python-based tools. The PyBullet physics engine [1] was employed to simulate the body dynamics and locomotion behavior of a quadruped robot under CPG control.

The Laikago robot model, available within PyBullet, was selected for the simulation due to its structural similarity to the Petoi Bittle. Although the Laikago is larger and more advanced, it features a comparable leg configuration with eight actuated joints, allowing for a relevant approximation of the real system. This resemblance enabled direct testing of the CPG's ability to coordinate leg motion, with only minor parameter tuning required to adapt the model. The physical structure of the robot in the simulation is defined using a URDF file (Unified Robot Description Format), which specifies the geometry, joint configuration, and inertial properties. The Hopf oscillator-based CPG, described in Section III-A, was used to generate joint trajectories in real time. These trajectories were applied to the joints of the simulated

[1] Erwin Coumans and Yunfei Bai, *PyBullet Quickstart Guide*, 2021. https://docs.google.com/document/u/1/d, accessed: 11/04/2025.

robot, allowing visualization of walking behavior and assessment of gait stability and coordination in a 3D environment. In addition to validating locomotion, this simulation environment was also used to perform a grid parameter space search. By systematically varying key CPG parameters, the resulting gait patterns and stability could be observed and compared efficiently. This approach facilitated the identification of optimised parameters for effective locomotion.

C. *Results of Simulation in Software*

To evaluate how different gait dynamics influence locomotion performance, a grid search was conducted over the parameters ω_{swing} and ω_{stance}, which respectively control the angular velocities of the swing and stance phases in the Hopf CPG. The goal was to identify configurations that enhance gait stability and forward progression. This parameter space search was performed using a fixed set of CPG and joint mapping parameters, as summarized in Table I. These parameters were chosen to provide a sufficiently good baseline for the search, although they are not necessarily optimal. In practice, these fixed values should ideally be adapted for each combination of ω_{stance} and ω_{swing}.

Table I: Fixed parameters used during the parameter space search

Parameter	Value	Parameter	Value
α	20	ω_{stance}	$\omega_{\text{stance}} \cdot \text{eff}_{\text{req}}$
β	20	ω_{swing}	$\omega_{\text{swing}} \cdot \text{eff}_{\text{req}}$
b	100	$\theta_{\text{hip base}}$	0.3
γ	5	$\theta_{\text{hip range}}$	0.25
eff_{req}	6	$\theta_{\text{knee base}}$	-0.8
		$\theta_{\text{knee range}}$	1.0

The results of this grid search are visualized in Figure 6, where darker colors indicate greater distances traveled, and red crosses indicate falls. We observe that the best locomotion performance, in terms of both distance and stability, occurs when $\omega_{\text{stance}} \approx 2 \cdot \omega_{\text{swing}}$. This result is consistent with [7], which shows that a longer stance phase improves passive stability, and is also biologically plausible based on gait observations in animals.

Moreover, the results demonstrate the high sensitivity of the system to parameter changes: even minor adjustments in ω_{stance} or ω_{swing} can significantly degrade stability. This underlines the importance of carefully tuning both the oscillator parameters and the joint mapping functions together. While this study held the mapping parameters fixed, optimal performance would likely require co-adapting all parameters jointly.

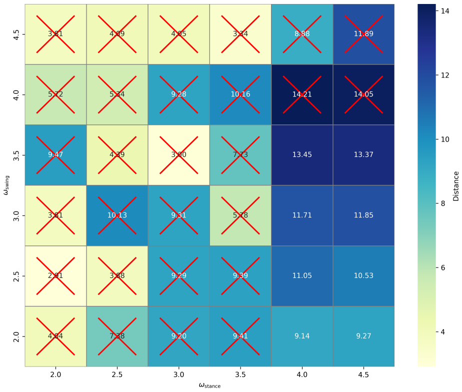


Figure 6: Heatmap of distance traveled for different values of ω_{stance} and ω_{swing} from 2 to 4.5 with step of 0.5. Red crosses indicate parameter combinations where the robot fell during the simulation.

V. Discussion

A. *Limitations of the CPG Model and Mapping Function*

The Hopf oscillator-based CPG model exhibits several limitations, primarily due to the interdependence of its parameters. Parameters such as u, ω_{stance} and ω_{swing}, are tightly coupled with the mapping gains for hip and knee joint amplitudes. Changes in the CPG parameters directly affect the amplitude of oscillations x_i and y_i, which in turn require retuning the mapping parameters $\theta_{\text{hip range}}$ and $\theta_{\text{knee range}}$ to maintain stable and biologically plausible joint trajectories. This strong coupling complicates the tuning process and may be mitigated by normalizing the oscillator outputs before applying the mapping function.

Another limitation arises from the initial conditions of the CPG states and their rapid convergence to the limit cycle. The oscillators quickly reach their maximum amplitude, which is inconvenient when starting locomotion from rest. In practical experiments with the physical robot, a waiting period is required to allow the CPG to settle into the limit cycle before initiating movement. Introducing a parameter to gradually increase the amplitude of the oscillator outputs could facilitate smoother gait initiation.

While the CPG model allows straightforward implementation of walking or trotting patterns, the interdependency of parameters makes achieving stable, straight-line locomotion challenging. Furthermore, the current open-loop control scheme, lacking sensory feedback, limits adaptability and robustness. Incorporating sensory feedback, such as through touch sensors influencing the oscillator states as described in [8], or via joint position measurements, could enhance gait stability.

B. System Limitations

From a mechanical and structural perspective, stability improvements have been obtained using 3D-printed rubber feet; however, further enhancements might be realized by integrating compliant elements into the robot's legs, which could improve shock absorption and adaptability to uneven terrain.

A significant limitation is the mismatch between the simulation model and the physical robot. The simulation uses the Laikago robot model, which differs from the Petoi Bittle in size and dynamics. This discrepancy complicates parameter transfer and reduces the power of simulation-based tuning. Future work would benefit from developing a simulation environment replicating the physical robot, facilitating more effective parameter optimization and real-world validation.

C. Future Work

Future research directions include the integration of sensory feedback into the CPG framework to allow adaptive and reactive control. Sensory inputs such as touch sensors or joint encoders could modulate oscillator dynamics, improving gait robustness, especially in unstructured environments [8].

Moreover, reinforcement learning could be implemented to automatically optimize the interdependent CPG parameters, overcoming the challenges of manual tuning [10]. Realistic simulation environments matching the physical robot should be implemented for training and evaluating such learning algorithms.

Finally, implementation and testing of these advances on the physical Petoi Bittle robot will be crucial for closing the gap between simulation and real-world performance and for validating the efficacy of the proposed control strategies.

VI. Conclusion

This work successfully demonstrates the implementation of CPGs based on the Hopf model on a low-cost quadrupedal robot. The control architecture, utilizing coupled nonlinear oscillators, generates stable and biologically plausible gaits in both simulation and for a real robot, enabling it to adapt to various terrains and external perturbations. The simulation and experimental results validate the effectiveness of the CPG in achieving robust and adaptive locomotion without the need for high-level feedback or complex planning strategies.

The integration of the CPG model into an affordable robotic platform highlights the potential of biologically inspired control mechanisms in enhancing the accessibility and practicality of quadrupedal robots. This approach not only retains the biological advantages of CPG-based control but also provides a more accessible solution for research and educational applications. Future work may explore further optimization of the CPG parameters and the integration of additional sensory feedback to enhance the robot's adaptability and performance in dynamic environments.

References

[1] M. H. Dickinson, C. T. Farley, R. J. Full, M. A. R. Koehl, R. Kram, and S. Lehman, "How animals move: An integrative view," *Science*, vol. 288, no. 5463, pp. 100–106, 2000.

[2] K. Y. Pettersen, "Snake robots," *Annual Reviews in Control*, vol. 44, pp. 19–44, 2017.

[3] R. Dwivedula, S. Modak, A. Akella, J. Biswas, D. Kim, and C. J. Rossbach, "Configbot: Adaptive resource allocation for robot applications in dynamic environments," 2025, arXiv preprint arXiv:2501.10513.

[4] M. Goulding, "Circuits controlling vertebrate locomotion: moving in a new direction," *Nature Reviews Neuroscience*, vol. 10, no. 7, pp. 507–518, 2009.

[5] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, vol. 21, no. 4, pp. 642–653, June 2008, source: PubMed.

[6] S. A. Collins and A. J. Richmond, "Hard-wired central pattern generators for quadrupedal locomotion," *Biological Cybernetics*, vol. 71, no. 5, pp. 375–385, September 1994, source: DBLP.

[7] L. R. A. Badri-Spröwitz and A. J. Ijspeert, "Passive compliant quadruped robot using central pattern generators for locomotion control," in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on*, November 2008, source: IEEE Xplore.

[8] L. Righetti and A. J. Ijspeert, "Pattern generators with sensory feedback for the control of quadruped locomotion," in *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 819–824.

[9] Camp, "OpenCat: Open-source Robotic Quadruped Project," https://github.com/PetoiCamp/OpenCat, 2025, accessed: 2025-06-10.

[10] G. Bellegarda and A. Ijspeert, "CPG-RL: Learning Central Pattern Generators for Quadruped Locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 547–12 554, Oct. 2022.