# Neurodynamic Sensory-Motor Phase Binding for Multi-Legged Walking Robots

Rudolf Szadkowski
*Faculty of Electrical Engineering*
*Czech Technical University in Prague*
Prague, Czech Republic
orcid:0000-0003-4075-116X
szadkrud@fel.cvut.cz

Jan Faigl
*Faculty of Electrical Engineering*
*Czech Technical University in Prague*
Prague, Czech Republic
orcid:0000-0002-6193-0792
faiglj@fel.cvut.cz

***Abstract*—Motivated by observations of animal behavior, locomotion of multi-legged walking robots can be controlled by the central pattern generators (CPGs) that produce a repetitive motion pattern. A rhythmic pattern, a gait, is defined by phase relations between all leg joints. In a case of an external influence such as terrain irregularity, some actuator phase can shift and thus disrupt the phase relations between the actuators. The actuator phase relations can be maintained only by synchronizing to the sensors, which output can indicate the motion disruption. However, establishing correct sensory-motor phase relations requires not only the motor phase model but also a model of the sensory phase, which is generally unknown. Although both sensory and motor phases can be modeled by single CPG, the capabilities of such CPG-based controllers are limited because they are not flexible and robust. In this paper, we propose to model the phases of each sensor and motor by separate CPGs. The phase relations between the sensor and motor phases are established by radial basis function (RBF) neurons learned with proposed periodic Grossberg rule for which we present the convergence proof. Based on the reported evaluation results using high-fidelity simulation, the proposed locomotion controller demonstrates the desired plasticity, and it is capable of learning multiple gaits with robust synchronization to terrain changes using sensor inputs.**

***Index Terms*—gait control, central pattern generator, phase control, neurodynamics, neural oscillator, multi-legged robots**

## I. Introduction

An important step towards walking robots deployed in a dynamic environment is to capture the animals' adaptability and robustness of their locomotion gait. A gait is a motion pattern that has repetitive motion phases during walking, such as stance and swing phases. The physiological evidence shows that the motion phase is controlled by the *Central Pattern Generator* (CPG), a group of recurrently connected spinal neurons generating oscillatory signals [1]. The CPG oscillations are both self-actuated and entrained by proprioception, providing both feedforward and feedback phase control [2]. The phase control determines the motion frequency and phase offsets between the sensors and motion phases, allowing the motor control to synchronize with the sensed environment. Animals can learn the sensory-motor synchronization as they can learn new motion patterns or tune them in a case of the body changes. Thus, we consider a plastic motion phase controller as an essential part of animal survival as for the longevity of walking multi-legged robot deployment in a long-term mission.

The core of biomimetic phase controllers is a CPG, a dynamic system with a *limit-cycle* attractor: cyclic trajectory to which all close states converge in the limit [2]. The limit-cycle dynamics provide oscillatory behavior, essential for the gait motion control, but it can also synchronize to entraining signal (an external periodic signal). The CPG that is synchronized to an entraining signal has the same frequency as the signal, and the phase offset between the signal and CPG is stable. Thus, if the phase of the entraining signal shifts, the phase of the CPG also shifts. The synchronized CPG provides a continuous model of the phase of the entraining signal, such as the sensory signal [3]. Hence, in a phase controller, the CPG can be used as (i) a phase controller of the motor, or (ii) a phase model of the sensor.

CPG-based controllers might use a single CPG as a motor phase controller and a sensory phase model. Using such shared sensory-motor CPG, we can model the motor phase as a function of the single sensory phase. However, such a solution limits the CPG-based control as it is impossible to synchronize the motor phase to multiple sensors while ensuring that the motor phase has the limit-cycle dynamics.

In this paper, we propose to decouple the shared sensory-motor CPG architecture, and model the sensory and motor phases with separate CPGs. The decoupled architecture allows each motor controller to synchronize all sensory inputs; therefore, it is possible to establish a stable phase offset between each sensory-motor pair. The phase offsets between the sensory and motor CPGs must be plastic to allow learning new gaits or adapting changes to body morphology. We propose to learn the sensory-motor phase offsets with Radial Basis Function (RBF) neurons [4] trained by Grossberg rule [5], which we modify for periodic learning. The proposed controller can produce the target phase control pattern and also robustly synchronize to the environment through sensory inputs.

The paper is organized as follows. A brief review of the

This work was supported by the Czech Science Foundation (GAČR) under research project No. 18-18858S.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

most related existing CPG-based controllers is presented in Section II. The problem of gait phase control is formally described in Section III. The proposed locomotion architecture and its building blocks are detailed in Section IV. The evaluation results on the proposed controller training of three different gaits are reported in Section V together with a comparison of the trained controller with other sensory-motor connectivity architectures to demonstrate advantages of the proposed decoupled architecture. A discussion of the results and proposed approach are presented in Section VI. Finally, the paper is concluded in Section VII.

## II. Related Work

CPG-based controllers are used in various locomotion related domains such is wearable robotics [3], or physiological models of invertebrate [6] [7] and vertebrate [8] locomotion, or legged robot motion control.

In robotic motion control, the *phase controller* is a part of the CPG-based gait controller that generates the phase of controlled variable whereas the *amplitude controller* modulates the phase control. There are various implementations of amplitude controllers using for example inverse kinematics [9] [10], direct mapping [11], [12] or reflexive systems [13]. For a detailed analysis of CPG-based controller architectures and their various use cases, a reader is kindly referred to the existing comprehensive review [14]. However, for the scope of this paper, we briefly highlight the main properties of the existing learnable CPG-based controllers.

Both amplitude and phase control require parametrization for each particular gait but also each particular robot body. Thus, there is a scientific effort to learn parameters using machine learning methods, as it is reviewed in [15]. A learnable coupled CPG network is designed in [16], which is synchronized to post-processed sensory inputs, where the sensory post-processing is learned by particle swarm optimization. This optimization method, however, cannot be used in real-time. In [4], the authors train the amplitude controller to modulate the output of the CPG-RBF network, where the RBF neuron weights are statically set on the CPG limit cycle. The RBF neuron activation is further modulated by the amplitude controller trained by self-supervised reinforcement learning. A real-time training of the CPG frequency is presented in the theoretical work [17], where the authors present a Hebb-like update rule for learning the frequency from the target periodic signal. The frequency training rule can be integrated directly into differential equations describing the CPG. Such an approach is fast and compact because the learning itself is solved by solving the differential equations describing the CPG controller. However, there is little work done on the methods learning phase offset between the modeled sensory phase and the control phase.

The close approach to the herein addressed sensory-motor phase binding is sensory-motor phase offset learning addressed in [3], where the authors model the human gait phase with the adaptive oscillator [17] in a framework that learns both the phase and sensory-motor phase offset. However, the architecture of the phase controller uses CPG as both the motor and sensor mode, which is not suitable for robust and adaptable locomotion control. In this paper, we propose novel architecture with separate CPGs to model the phase of each sensor and motor to achieve the desired plasticity of the locomotion controller.

## III. Problem Statement

The adaptive gait of a multi-legged robot is a repetitive motion pattern where all $N$ joints move synchronously with the environment sensed by $M$ sensors. The phase controller learning task is twofold: (i) learn to generate the phase control $\mathbf{a}(t) \in [0,1]^N$ that corresponds to $T$-periodic target pattern $\mathbf{d}(t) \in [0,1]^N$; (ii) learn and utilize the phase offsets between the motor phase control $\mathbf{a}(t)$ and sensory inputs $\mathbf{x}(t) \in [0,1]^M$. Assuming that variables $x_i(t)$ and $a_j(t)$ are $T$-periodic, in convergent state, we can define their phases as a variable that grows linearly with time: $\Phi(a_j(t)) = 2\pi T^{-1}t + \phi_{a_j}$, where $\phi_{a_j}$ is initial phase; similarly we define $\Phi(x_i(t))$. Then we formally define the phase offset between $x_i(t)$ and $a_j(t)$ as $\phi_{ij} = \Phi(x_i(t)) - \Phi(a_j(t))$. Generally, the explicit expression of $\Phi(a_j(t))$ and $\Phi(x_i(t))$ is not known. Moreover, in practice, the perception $x_i(t)$ can indicate motion deviation due to external influences, such as uneven terrain or external force. The deviation propagates into the phase $\Phi(x_i(t)) = 2\pi T^{-1}t + \phi_{x_i} + \varepsilon_\phi$ and results into the phase offset deviation $\phi_{ij} + \varepsilon_\phi$. This offset deviation should be corrected by the phase controller in order to maintain the gait pattern.

In this work, we use the CPGs to model the phases of each $x_i$ and $a_j$ variables, and the RBF neurons learn the phase offset as it is described in the following section.

## IV. Training The Gait And Synchronization

In this section, we describe the CPG-based phase controller, which learns the prescribed gait pattern and sensory-motor phase offsets. For $N$ actuators, we train phase controller output $\mathbf{a}^{\mathrm{e}}(t), \mathbf{a}^{\mathrm{f}}(t) \in [0,1]^N$ to imitate target phase control $\mathbf{d}^{\mathrm{e}}(t), \mathbf{d}^{\mathrm{f}}(t) \in [0,1]^N$ and synchronize to $M$ sensory signals $\mathbf{x}(t) \in [0,1]^M$. Superscripts $\mathrm{e}$ and $\mathrm{f}$ denote antagonistic movements extension and flexion; in practice on one joint, they represent opposing movements (positive and negative angular velocity). By coupling to sensory signals, the CPGs indirectly couple between themselves and produce a synchronous phase control. Assuming the sensory and motor variables are coupled in the environment, they have the same period $T$ during undisturbed gait motion.

There are two building blocks of the proposed controller: the CPG and trainable RBF neuron. Let the CPG be modelled by $\dot{\mathbf{y}} = f(\mathbf{y}, c(t))$, where differential function $f$ describes dynamics of the CPG state $\mathbf{y}(t) \in \mathbb{R}^D$ perturbed by CPG input $c(t) \in \mathbb{R}$. Assuming the CPG has the limit-cycle property, in the limit, the state $\mathbf{y}$ converges to the cyclic trajectory that is denoted as $\mathbf{y}_0 \subset \mathbb{R}^D$, where the CPG oscillates with intrinsic period $T_0$: $\mathbf{y}(t) = \mathbf{y}(t + T_0)$ [2]. If the input signal $c(t)$ has

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

period very close to $T_0$ and has small amplitude[1] the CPG synchronizes to $c(t)$ and estimates its phase [3], while keeping the shape of the limit-cycle $\mathbf{y}_0$ unchanged. Therefore, each point of $\mathbf{y}_0$ represents a certain phase of the input $c(t)$.

The closeness to certain phase is defined by a *Radial Basis Function* (RBF) $a = \exp(-\varepsilon||\mathbf{y} - \mathbf{m}||^2) = \varphi(\mathbf{y}, \mathbf{m})$, where $\mathbf{m}$ is the weight that we train to approach certain segment of the limit-cycle $\mathbf{y}_0$, and thus $a \in (0, 1)$ measures closeness to that segment [4]. We propose following periodic Grossberg rule modification to update the weight $\mathbf{m}$

$$\dot{\mathbf{m}} = \eta(t)d(t)(\mathbf{y} - \mathbf{m}) = g(\mathbf{m}, \mathbf{y}, d(t), \eta(t)), \qquad (1)$$

where the added term $d(t) \in [0, 1]$ is a target signal with the period $T = T_0$ that selects the limit-cycle segment to which the weight $\mathbf{m}$ converge (visualised in Fig. 5). For the discrete-time, we provide proof of convergence in the Appendix. Henceforth, we refer to trained weight convergence to limit-cycle as RBF neuron *binding* to CPG.

### A. The Trainable Phase Controller

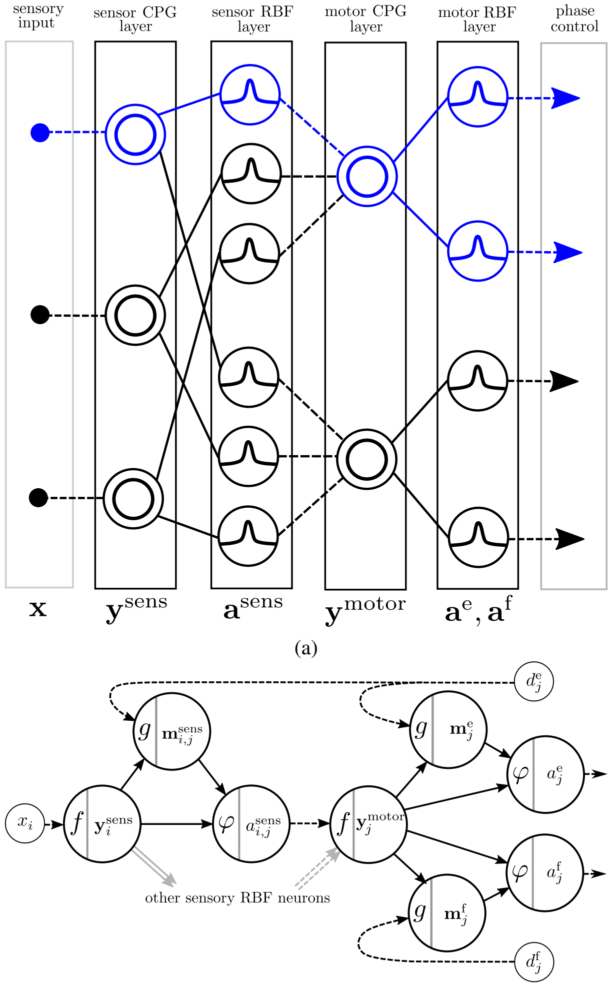
The building blocks presented in the previous section are utilized in the proposed architecture containing the following four layers: sensory CPG, sensory RBF, motor CPG, and motor RBF layers, depicted in Fig. 1a. In this work, we assume that the intrinsic CPG period $T_0$ is already very close to the target gait period $T$, and rely on the inherent property of CPG to synchronize to the external period that is close to its intrinsic period.

In *the sensory CPG layer*, we let each $i$-th sensory signal $x_i$ entrain the $i$-th sensory CPG: $\dot{\mathbf{y}}_i^{\text{sens}} = f(\mathbf{y}_i^{\text{sens}}, x_i(t))$; thus, after the convergence, the CPG state $\mathbf{y}_i^{\text{sens}}$ continuously represents the phase of $x_i$. The *sensory RBF layer* binds to sensory CPGs. Each $i, j$-th sensory RBF neuron weight $\mathbf{m}_{i,j}^{\text{sens}}$ is trained by (1) to bind the $i$-th limit-cycle segment selected by the target $d_j^{\text{e}}$. The activation of $i, j$-th sensory RBF neuron is then $a_{i,j}^{\text{sens}} = \varphi(\mathbf{y}_i^{\text{sens}}, \mathbf{m}_{i,j}^{\text{sens}})$. Since weights $\mathbf{m}_{\bullet,j}^{\text{sens}}$ are trained by the same target $d_j^{\text{e}}(t)$ the activation $\mathbf{a}_{\bullet,j}^{\text{sens}}$ peaks are close to each other. The average of sensory RBF neuron activations is then fed into *the motor CPG layer*, where the activations entrain the motor CPGs

$$a_j^{\text{sens}} = M^{-1} \sum_{i}^{M} a_{i,j}^{\text{sens}}, \qquad (2)$$

$$\dot{\mathbf{y}}_j^{\text{motor}} = f(\mathbf{y}_j^{\text{motor}}, a_j^{\text{sens}}). \qquad (3)$$

The limit-cycle $\mathbf{y}_{0,j}^{\text{motor}}$ represents the phases of the $j$-th actuator (muscle or servo), therefore it contains two phase segments during which the actuator flexes and extends (body-coxa joint retracts and protracts the limb while coxa-femur joint ascends and descends the limb; see Fig. 2b). The extensor and flexor phase segments of the $j$-th actuator are estimated by two RBF neurons in *the motor RBF layer*. The $j$-th extensor RBF neuron activation is then $a_j^{\text{e}} = \varphi(\mathbf{y}_j^{\text{motor}}, \mathbf{m}_j^{\text{e}})$, where weight $\mathbf{m}_j^{\text{e}}$ approaches the extension phase segment selected by $d_j^{\text{e}}$: $\dot{\mathbf{m}}_j^{\text{e}} = g(\mathbf{m}_j^{\text{e}}, \mathbf{y}_j^{\text{motor}}, d_j^{\text{e}}(t), \eta(t))$; in the same manner, we train the flexor RBF neuron.


sensory input
sensor CPG layer
sensor RBF layer
motor CPG layer
motor RBF layer
phase control
$\mathbf{x}$
$\mathbf{y}^{\text{sens}}$
$\mathbf{a}^{\text{sens}}$
$\mathbf{y}^{\text{motor}}$
$\mathbf{a}^{\text{e}}, \mathbf{a}^{\text{f}}$
$d_j^{\text{e}}$
$g$ $\mathbf{m}_{i,j}^{\text{sens}}$
$g$ $\mathbf{m}_j^{\text{e}}$
$\varphi$ $a_j^{\text{e}}$
$x_i$
$f$ $\mathbf{y}_i^{\text{sens}}$
$\varphi$ $a_{i,j}^{\text{sens}}$
$f$ $\mathbf{y}_j^{\text{motor}}$
$\varphi$ $a_j^{\text{f}}$
other sensory RBF neurons
$g$ $\mathbf{m}_j^{\text{f}}$
$d_j^{\text{f}}$

(a)

(b)

Fig. 1. The architecture of the proposed controller composed of four layers. (a) Example of connectivity between the layers for $M = 3$ sensory inputs and $N = 2$ motor outputs. The dashed lines represent a scalar connection, while the full lines represent a $D$-dimensional vector connection. The highlighted part (blue) is shown in detail at the bottom subfigure (b).

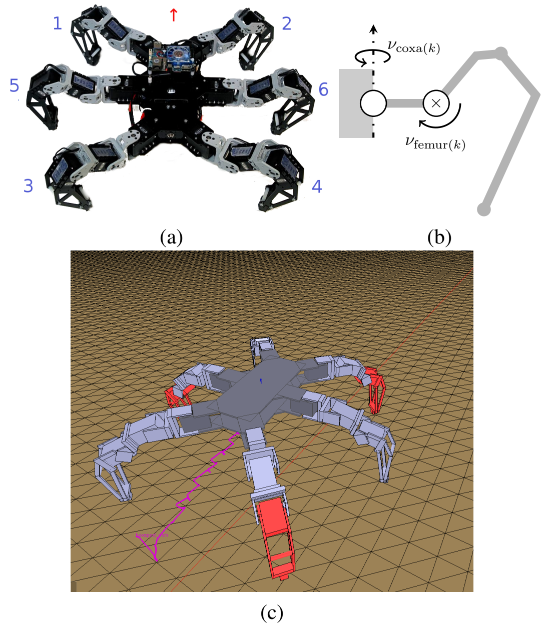
After converging, the activations $\mathbf{a}^{\text{e}}, \mathbf{a}^{\text{f}}$ not only approximate the target phase control $\mathbf{d}^{\text{e}}(t), \mathbf{d}^{\text{f}}(t)$ but are also synchronized to sensory signals $\mathbf{x}$. The synchronization ensures that the sensory-motor phase offsets are stable: if the sensory phase shifts, the controller continuously shifts the motor phase to the learned phase offset. The gait training and sensory-motor synchronization are demonstrated in the following section.

## V. Results

This section reports on the proposed controller deployed on the hexapod robot modeled in high-fidelity simulation framework V-REP [18], visualized in Fig 2c. The report has two parts: Section V-B where we show that the proposed controller can learn different gaits such as ripple, caterpillar, and tripod gait; Section V-C where we use the trained tripod gait in two scenarios requiring the phase adaptation and compare

[1] The region of phase closeness and required amplitude is defined by Arnold tongue that is characteristic for each particular CPG model.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE



(a) (b)

(c)

Fig. 2. (a) The modeled robot and its leg numbering. (b) $k$-th leg detail with controlled joints. (c) The experimental setup in high-fidelity simulation software V-REP, the hexapod model [18] is walking tripod gait on a flat terrain.

the multi-sensory (proposed) and single-sensory entrainment architectures. In the following subsection, we describe the setup of the gait controller, which is used in the presented experiments.

### A. Setup

The hexapod robot, depicted in Fig. 2a, has six legs; on each leg, there are three joints connecting body-coxa, coxa-femur, and femur-tibia. In this work, we train the phase controller to control body-coxa and coxa-femur joints of the $k$-th leg, denoted as $\mathrm{coxa}(k)$ and $\mathrm{femur}(k)$, respectively, see Fig. 2b. Therefore, the phase controller learns to control twelve joint phases; $N = 12$. Each $j$-th joint has its angle $p_j$ controlled by integrated velocity controll $\nu_j$. The velocity control is generated by the phase control modulation that inhibits and combines the phase control output

$$a_{\mathrm{coxa}(k)} = \begin{bmatrix} a^{\mathrm{e}}_{\mathrm{coxa}(k)} \\ a^{\mathrm{f}}_{\mathrm{coxa}(k)} \end{bmatrix}^T \begin{bmatrix} \sigma^{\mathrm{e}}_{\mathrm{coxa}(k)} \\ \sigma^{\mathrm{f}}_{\mathrm{coxa}(k)} \end{bmatrix}, \tag{4}$$

$$a_{\mathrm{femur}(k)} = \begin{bmatrix} h(a^{\mathrm{e}}_{\mathrm{femur}(k)} - x_{\mathrm{touch}(k)}) \\ a^{\mathrm{f}}_{\mathrm{femur}(k)} \end{bmatrix}^T \begin{bmatrix} \sigma^{\mathrm{e}}_{\mathrm{femur}(k)} \\ \sigma^{\mathrm{f}}_{\mathrm{femur}(k)} \end{bmatrix}, \tag{5}$$

where $h(q) = \max(q, 0)$ and $x_{\mathrm{touch}(k)} = 1$ if the $k$-th leg touches the ground, and thus the further descend of the leg is stopped. The hyperparameters are set to $\sigma^{\mathrm{e}}_{\mathrm{coxa}(k)} = 1.5$ and $\sigma^{\mathrm{e}}_{\mathrm{femur}(k)} = 3$. The combined phase control of the $j$-th joint $a_j$ forces oscillations to angular velocity $\nu_j$ while keeping the position $p_j$ close to resting position $p^{\mathrm{rest}}_j = 0$.

$$\nu_j = a_j[1 - \min(\vartheta_1|p^{\mathrm{rest}}_j - p_j(t)|, 1)] + \vartheta_2(p^{\mathrm{rest}}_j - p_j(t)). \tag{6}$$

The stabilization parameters were set to $\vartheta_1 = 150^{-1}$ and $\vartheta_2 = 0.008$. The hyperparameters of phase modulation were manually tuned to the hexapod robot model walking the tripod gait.

For each of six legs, we observed the following three events on $k$-th leg:

1) $x_{\mathrm{touch}(k)} = 1$ if the leg touches the ground;
2) $x_{\mathrm{desc}(k)} = 1$ if the coxa-femur joint has positive angular speed;
3) $x_{\mathrm{back}(k)} = 1$ if the body-coxa joint angle is behind its resting position.

Therefore, there are $M = 18$ sensory inputs.

Matsuoka neural oscillator [19] is used as CPG $\mathbf{y} = (u_1, u_2, v_1, v_2)$ with the following dynamics

$$\tau\dot{v}_1 = h(u_1) - v_1, \tag{7}$$

$$\tau\dot{v}_2 = h(u_2) - v_2, \tag{8}$$

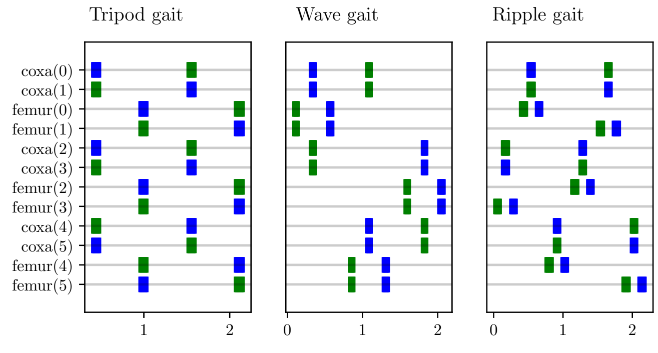
$$\gamma\dot{u}_1 = -u_1 - h(u_2)\alpha - v_1\beta + 1, \tag{9}$$

$$\gamma\dot{u}_2 = -u_2 - h(u_1)\alpha - v_2\beta + 1 + h(c(t) - \lambda_1)\lambda_2, \tag{10}$$

where $c(t)$ is the periodic input signal which entrains the CPG. The CPG hyperparameters are set to $\tau = 0.5, \gamma = 0.25, \alpha = \beta = 2.5, \lambda_1 = 0.5, \lambda_2 = 0.05$. The $\varepsilon$ hyperparameter in the RBF neuron is set to 12 and 4 for sensory and motor layers, respectively. All network hyperparameters were found empirically. The numerical integration is done by the Euler method with a step value of 0.01.

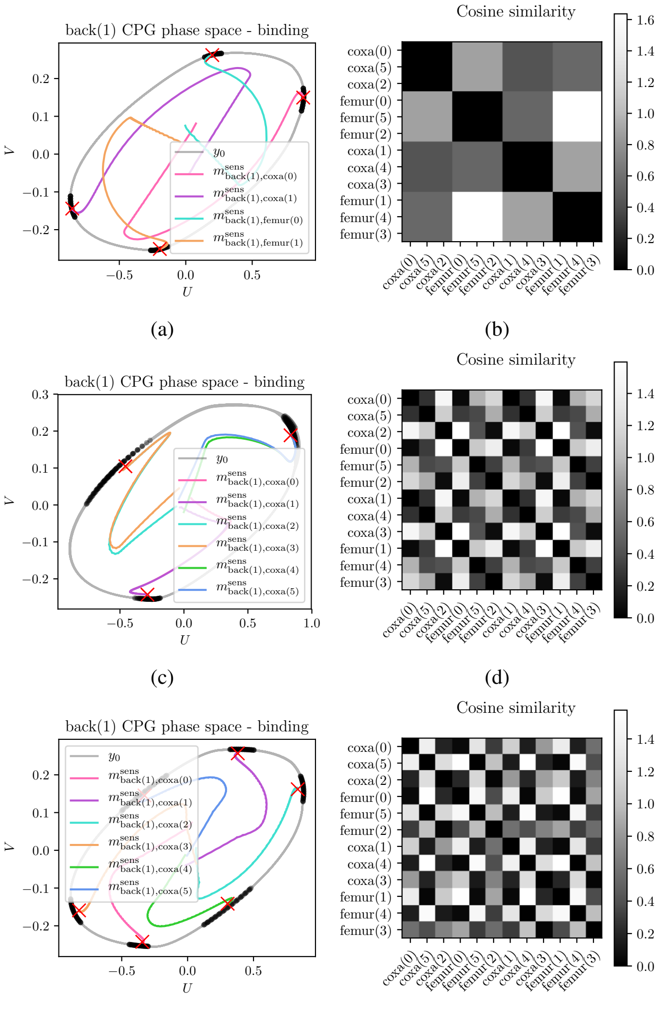
### B. Training the Gait Phase Control

The controller has been trained to control different gaits: ripple, caterpillar, and tripod gaits depicted in Fig. 3. For each gait the control signals $\mathbf{d}^e(t)$ and $\mathbf{d}^f(t)$ are generated as repeating patterns depicted in Fig. 1b, where the period of the gait cycle $T = 2.23$ is close to the estimated CPG period $T_0 \approx 2.23$. Each gait phase control is trained to $t = 400$ (40,000 Euler method steps) with the learning rate $\eta(t)$ set to one to show that the RBF weights converge anyways.



(a) (b) (c)

Fig. 3. The target gait patterns. The blue and green rectangles mark extensor $\mathbf{d}^{\mathrm{e}} = 1$ and flexor $\mathbf{d}^{\mathrm{f}} = 1$ events, respectively. (a) During the tripod gait, the two groups of legs $\{1, 6, 3\}$ and $\{2, 5, 4\}$ are alternating between the stance and swing. (b) During the caterpillar gait, one pair of contralateral legs swing at time, where the pairs swing in the order $\{3, 4\}$, $\{1, 2\}$, and $\{5, 6\}$. (c) During the ripple gait, one leg swings at time in the order 3, 2, 6, 4, 1, and 5.



(a) (b)

(c) (d)

(e) (f)

Fig. 4. Sensory RBF neurons binding to sensory CPG modelling phase of $x_{\text{back}(1)}$ perception for multiple trained gaits. (a) Front legs weights $\mathbf{m}^{\text{sens}}_{\text{back}(1)}$ during training the tripod gait. The weights evolution and the limit-cycle trajectories are projected into $UV$ space; $U = u_2 - u_1$ and $V = v_2 - v_1$. The weights are attracted to their respective limit-cycle segments (black points). The weight trajectories are twisted as the CPG shifts towards its synchronous position. The final state of each weight (red mark) is close to each segment. In (b), we compare the cosine similarity of the final weights for each leg. For a comparison we show more complex gaits such as the caterpillar gait (c),(d), and the ripple gait (e),(f).

At the beginning of the tripod gait training, the robot starts randomly but periodically moving with its limbs, which in turn causes a periodic sensory signal. The sensory signals entrain their respective sensory CPGs. In Fig. 4a, we can see how the sensory RBF neurons bind a limit-cycle segment that corresponds to their target phase $d^{\text{e}}_{j}$. By comparing the learned weights $m^{\text{sens}}_{i,j}$ of each $j$-th joint with cosine similarity, we can see which legs move concurrently and which move in anti-phase; see Fig. 4b. For example, the front left coxa-body joint has very similar weights to hind left joint while it is dissimilar with the front right joint; which is the expected behavior of the tripod gait. The sensory RBF neurons entrain motor CPGs.

Similiarly to sensory RBF binding, it can be seen in Fig. 5, how the motor RBF neurons bind their respective limit-cycle segments selected by the target $d^{\text{e}}_{j}$ or $d^{\text{f}}_{j}$. As the weight of the RBF neuron gets closer to the limit-cycle segment, its activation amplitude gets closer to one, and the activation peaks during $d(t) = 1$, see Fig 5. The activity of each trained RBF neuron during positive value of its respective target $d(t) = 1$ is tracked in Fig. 6, where we can see that the tracked activity converges to the maximal activity. At the end the peak-target, overlap amplitude is bounded by intervals $[0.917, 0.999]$, $[0.995, 0.999]$, and $[0.995, 0.999]$ for sensory, extensor, and flexor RBF neurons, respectively. At the end of the training, the robot is able to walk the tripod gait.

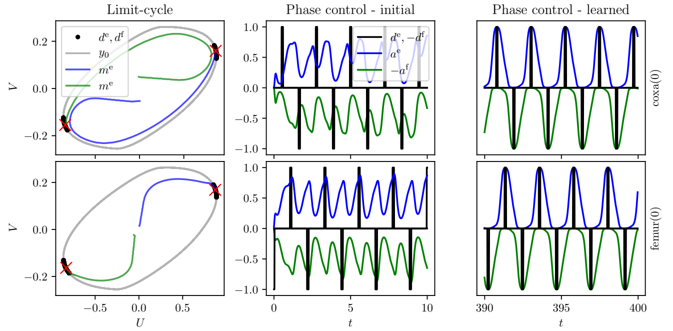


Fig. 5. On the left, the motor RBF neurons (green and blue trajectories) binding to their respective target limit-cycle segments (black). The red cross marks their final weights. On the right, the activations (green and blue evolutions) of RBF neurons as they learn the binding, where at the start the activations are weak and not uniform, but after the training, they peak during the target activations $d(t) = 1$ (black pulse).

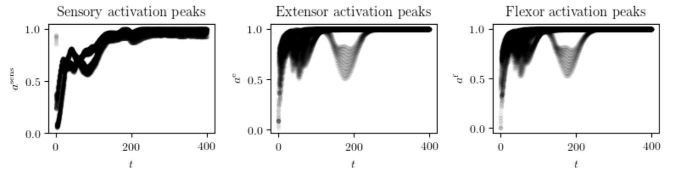


Fig. 6. Visualisation of the target-activation overlap convergence. The visualized points are activations of all RBF neurons that are concurrent with their respective targets $d^{\text{e}}(t) = 1$ or $d^{\text{f}}(t) = 1$. The target and RBF activations have the same period; thus, high values indicate that the target is concurrent with the activation peak, which is the desired behavior.

Also, for the tripod gait training, we trained the ripple and caterpillar gaits. As can be seen in Fig. 4, each gait has different sensory RBF binding due to different gait patterns. The results are further discussed in Section VI.

The output of the phase control, such as in Fig. 5, can be generated as an open-loop control; however, the CPG-based phase control also provides the ability to synchronize to the environment. The synchronization has been further examined, and the results are reported in the following section.

## C. Synchronization Evaluation

The advantages of the proposed architecture are demonstrated in two scenarios that require synchronization. We also compare the proposed architecture, where one motor phase synchronizes all sensory inputs, to two *single entraintment* architectures, where each motor phase synchronizes to just a single selected sensory input.

The sensory input selection requires prior knowledge. Here, we assume two strong motor-sensory couplings (sensory signals that are caused by particular joint): (i) $x_{\text{touch}(k)}$ and $\nu_{\text{femur}(k)}$, and (ii) $x_{\text{back}(k)}$ and $\nu_{\text{coxa}(k)}$. In the *self-sense* architecture, each motor CPG is entrained by a strongly coupled sensory signal. In the *circ-sense* architecture, each motor CPG is entrained by a sensory signal strongly coupled to the clockwise neighboring joint. The connectivity of all three used architectures is depicted in Fig. 7. Both architectures are implemented by modifying the combination of sensory activity in (2) to relaying just one selected sensory activation. Despite the similarity of wave-shapes of RBF neuron activations, the little differences can slightly change the timing in motor CPGs. Therefore, each architecture was trained separately, so each has weights tuned to its connectivity. Then we let each trained phase controller walk with the robot and recorded its position control $\mathbf{p}^{\text{force}}$, which we use in the following two scenarios.

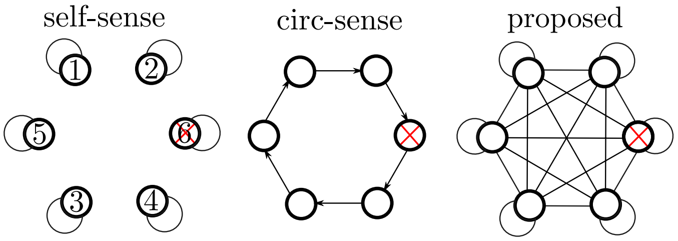
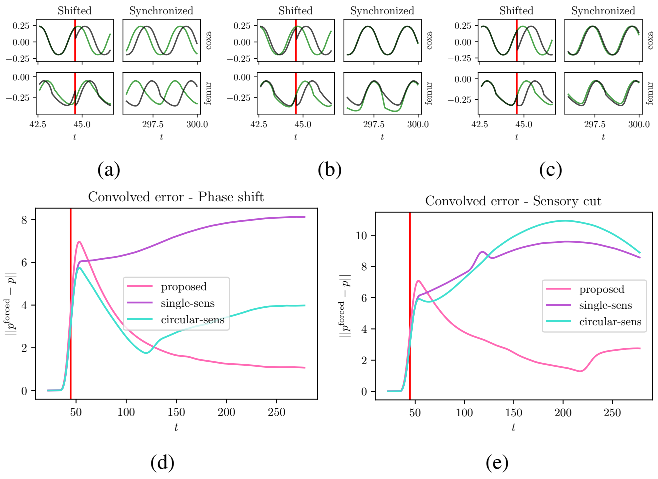


Fig. 7. The connectivity of the sensor RBF layer that binds the motor and sensory phase. Each node represents a coxa-femur joint (body-coxa joints are omitted in this depiction). The self-sense architecture connects sensory inputs to its strongly coupled joint. The circ-sense architecture connects sensory inputs neighboring to their strongly coupled joint. The proposed architecture connects each joint to all sensory inputs. The red cross marks the leg, which is not providing sensory output during the sensory-cut scenario.

In the *phase-shift scenario*, the robot is controlled by the trained controller until $t = 44.6$ when both front legs are externally forced to walk with shifted phase relative to the other four legs, visualized in Fig. 8. The controller should adapt the phase shift by continuously aligning the phase of the hind and middle legs with the front legs. Every compared controller architecture has its own forcing position control $\mathbf{p}^{\text{force}}$. The experiment shows that the sensory-motor binding connectivity (visualized in Fig. 7) plays a crucial role in the phase adaptation and can fail if designed poorly, as we can see in the case of self-sense in Fig. 8a.

In the *sensory-cut scenario*, we evaluate robustness against the sensory cut, where at $t = 44.6$, the sensory inputs strongly coupled with the sixth leg are turned off ($x_{\text{touch}(6)} = x_{\text{back}(6)} = x_{\text{desc}(6)} = 0$), depicted by the red cross in Fig. 7. We also apply the same phase shift from the previous experiment, so the phase controllers have to utilize the sensory input and adapt the shift. The joints of the sixth leg in self-sense and joints of the fourth leg in circ-sense are left without sensory entrainment. These unbound joints cannot synchronize and also have a slightly different frequency, and thus they are constantly phase-shifting in relation to the force control, which is visible in Fig. 8e. Both experiments demonstrate that relying on just one sensory input is not a robust solution for phase control.



(a) (b) (c)

(d) (e)

Fig. 8. On top, a pose evolution of (a) self-sense, (b) circ-sense, and (c) the proposed architecture. The pose evolution show forced left front leg (black) and left hind leg controlled by the phase controller (green). At $t = 44.6$ (red), the forced control shifts the phase of the front legs. (d) and (e) shows the pose error for the phase shift and sensory cut scenarios, respectively. The error is calculated as the distance of the observed phase $\mathbf{p}$ to the forced control $\mathbf{p}^{\text{forced}}$. The proposed controller starts with the highest error, but at the end of the experiment, it has the lowest error.

## VI. Discussion

The proposed phase controller can learn multiple gaits by binding the sensor and motor phases with the periodic Grossberg rule (1). The learned phase offset of the tripod, caterpillar and ripple gait, can be checked in Fig. 4, where the body-coxa weights $\mathbf{m}^{\text{sens}}_{\text{back}(2),\text{coxa}(k)}$ converge on two, three and six limit-cycle segments, respectively. The number of segments corresponds to the number of the concurrent coxa return strokes during one gait cycle; see blue body-coxa joint events in Fig. 3.

The more detailed view on learned phase dependencies is provided by the cosine similarity between $\mathbf{m}^{\text{sens}}_{\text{back}(2),j}$ weights visualised in Fig. 4. For example, in the ripple gait, the extension of $\text{femur}(6)$ and $\text{coxa}(6)$ joints is almost concurrent, see Fig 3c, which is encoded in the similarity matrix in Fig. 4f, where both joints have similar weights.

As the weights converge to their respective segments, the RBF neurons activations gain amplitude, and their maxima are concurrent with the target $d(t) = 1$; see Fig. 5. The activation-target overlap is tracked for the tripod gait in Fig. 6, where we can see that the periodic activations converge to the upper bound two times before and after $t \approx 200$. The first convergence of motoric RBF neurons at $t \approx 120$ caused the change in the period of sensory signals generated by the environment, and thus "rotating" the sensory CPGs. Such a behavior can be observed in Fig. 4, where the weights are twisted due to tracking the rotating limit-cycle segments. The change in the sensory CPG layer is propagated into the motor

CPG layer and caused the divergence of the motor RBF activations at $t \approx 200$. After the RBF motor neurons adapted to the change, the RBF activation peaks and target overlap converged to the upper bound again. Based on the observation, we can conclude that the training converged to weights that correspond to the required gait.

The trained tripod gait controller is used in synchronization experiments, where we demonstrate that the proposed architecture, where the motor CPG synchronizes to multiple sensory signals, is more robust than the single sensory entrainment architectures (depicted in Fig. 7). The self-sense architecture fails to synchronize to shifted sensory input, as can be seen in Fig. 8. Indirect connections maintain the synchronization between joints through the environment. In single sensory entrainment architectures, such an indirect connection requires prior knowledge about the robot's morphology and its sensory-motor dynamics. The circ-sense controller is such a designed controller, and it performs well in the phase shift scenario. However, the circ-sense controller is not robust against sensory failures since each joint relies on just one sensor; see Fig. 8. In the sensory cut scenario, the proposed controller overperforms both single sensory entrainment architectures. The proposed controller does not require prior knowledge about sensory-motor dynamics, and we demonstrate that it can robustly adapt to phase changes in the environment.

The periodic Grossberg rule (1) works under the assumption that the period of the target signal and CPG are very close. The assumption of equal periods poses a limitation to the proposed controller as the frequency of the gait must be either tuned to the CPG or vice versa. There are methods of CPG frequency learning, such as [17], which we propose to integrate in the future. Another limitation of the proposed controller is that it works well with gaits where power and return stroke periods are similar, e.g., tripod gait; however, it performs poorly during gaits with different power/return stroke periods, e.g., caterpillar and ripple gaits. We trained the phase control of both caterpillar and ripple gaits, but the result motion was unbalanced and inefficient. The phase control modulation (6) is designed and tuned to tripod gait as the phase control modulation is not in the scope of this paper. Integration with learnable phase modulation (amplitude control) is a natural future step to obtain a fully learnable gait controller.

## VII. Conclusion

A plastic CPG-based controller that robustly synchronizes the motor control to the sensed environment has been presented. The controller contains four layers, the sensory CPG layer which models the phase of multiple sensory signals; the sensory RBF layer which binds the phase of sensory CPGs and entrains the next layer; the motor CPG layer which models the phase of multiple actuators; the motor phases are then bound by the last layer, the motor RBF layer which generates the phase control. We propose to use the periodic Grossberg training rule to learn the binding between RBF neurons and CPG. The proposed controller learned to control multiple different gaits. The learned tripod gait was evaluated against two single sensory entrainment architectures in scenarios requiring phase adaptation. Unlike the single sensory entrainment architectures, the proposed multi-sensory entrainment architecture can utilize multiple sensory signals and is robust against sensory failures, without the need for prior knowledge about the sensory-motor dynamics. In future work, we aim to integrate frequency and phase modulation learning to develop a fully learnable model-free gait controller.

## Appendix
## Periodic Grossberg Rule Convergence on Limit-Cycle

Let $\mathbf{y}(t) \in \mathbb{R}^D$ is a CPG state that converged into a limit-cycle $\mathbf{y}(t) \in \mathbf{y}_0 \subset \mathbb{R}^D$ with intristic period $T_0$, thus $\mathbf{y}(t + nT_0) = \mathbf{y}(t), n \in \mathbb{N}$. Let $d(t) = [\![\exists n \in \mathbb{N} : t \in [a,b] + nT]\!]$ be a $T$-periodic target signal that is 1 during $[a+nT, b+nT], n \in \mathbb{N}$ intervals; and 0 otherwise. Assume that the intrinsic CPG period $T_0$, and the target signal period $T$ are the same; $T_0 = T$. For the time interval $[q,r]$, we denote *limit-cycle segment* $\{\mathbf{y}(t) | t \in [q,r]\} \subset \mathbf{y}_0$ as $\mathbf{y}([q,r])$. We apply convergence theorem presented in [5] to show that the periodic Grossberg rule (1) also converges. However, to use the theorem [5], we must relax to discrete time steps $\tau_i = \epsilon i; i \in \mathbb{N}$, where $0 < \epsilon \ll (b-a)$, and define the discrete time periodic Grossberg rule

$$\Delta \mathbf{m}(\tau_i) = \eta(\tau_i) d(\tau_i)(\mathbf{y}(\tau_i) - \mathbf{m}). \tag{11}$$

By relaxing to discrete time steps, we are not losing much, since we already use the Euler method (see Section V-A), which solves the differential equations in discrete time steps. If both $\epsilon \ll T_0$ are rational numbers (in practice they usually are), then the $\epsilon$ and $T_0$ are commensurable; thus, the set of limit-cycle samples $Y_0 = \{\mathbf{y}(\tau_i) | i \in \mathbb{N}\}$ is finite and so is the sampled limit-cycle segment $\mathbf{y}([a,b]) \cap Y_0$.

**Theorem 1.** *Let the learning rate $\eta(\tau_i)$ converge to 0 and its series $\sum \eta(\tau_i)$ diverge. The weight $\mathbf{m}(\tau_i)$ updated by (11), converges to the centroid of the sampled limit-cycle segment $\mathbf{y}([a,b]) \cap Y_0$.*

*Proof:* If $d(\tau_i) = 0$, then there is no update $\Delta\mathbf{m}(\tau_i) = 0$. Thus, let consider only such time steps $\tau_i'$ when $d(\tau_i') = 1$. Then we can rewrite (11) to $\Delta\mathbf{m}(\tau_i') = \eta(\tau_i')(\mathbf{y}(\tau_i') - m)$, where $\tau_i' \in [a + nT, b + nT]$ for some $n$. The CPG state $\mathbf{y}(\tau_i')$ is then in a segment $\mathbf{y}([a + nT, b + nT])$ for some $n$. By the assumption $T_0 = T$, and recalling that $\mathbf{y}(t)$ is $T_0$-periodic, there is, in fact, just a signle segment $\mathbf{y}([a+nT_0, b+nT_0]) = \mathbf{y}([a,b] + nT_0) = \mathbf{y}([a,b])$ from which $\mathbf{y}(\tau_i')$ can be drawn. The set of all $\mathbf{y}(\tau_i')$ points is then the sampled limit-cycle segment $\mathbf{y}([a,b]) \cap Y_0$ that is finite. By transforming the periodic Grossberg rule to standard Grossberg rule, and showing $\mathbf{y}([a,b]) \cap Y_0$ is finite, we meet the requirements of [5]. ■

## References

[1] T. G. Brown, "The factors in rhythmic activity of the nervous system," *Proceedings of the Royal Society of London. Series B, Containing Papers of a Biological Character*, vol. 85, no. 579, pp. 278–289, 1912.

[2] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization without formulae*, ser. Cambridge Nonlinear Science Series. Cambridge University Press, 2001, pp. 25–172.

[3] T. Yan, A. Parri, V. Ruiz Garate, M. Cempini, R. Ronsse, and N. Vitiello, "An oscillator-based smooth real-time estimate of gait phase for wearable robotics," *Autonomous Robots*, vol. 41, no. 3, pp. 759—774, 2017.

[4] M. Pitchai, X. Xiong, M. Thor, P. Billeschou, P. L. Mailänder, B. Leung, T. Kulvicius, and P. Manoonpong, "Cpg driven rbf network control with reinforcement learning for gait optimization of a dung beetle-like robot," in *Artificial Neural Networks and Machine Learning – ICANN 2019: Theoretical Neural Computation*, 2019, pp. 698–710.

[5] D. M. Clark and K. Ravishankar, "A convergence theorem for grossberg learning," *Neural Networks*, vol. 3, no. 1, pp. 87 – 92, 1990.

[6] T. I. Tóth, S. A. Knops, and S. Daun-Gruhn, "A neuro-mechanical model explaining forward and backward stepping in the stick insect," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 107, no. 12, pp. 3267–3280, 2012.

[7] A. Büschges and J. Schmidt, "Neuronal control of walking: studies on insects," *e-Neuroforum*, vol. 21, no. 4, pp. 105–112, 2015.

[8] A. Frigon and S. Rossignol, "Experiments and models of sensorimotor interactions during locomotion," *Biological cybernetics*, vol. 95, no. 6, pp. 607–627, 2006.

[9] G. Sartoretti, S. Shaw, K. Lam, N. Fan, M. Travers, and H. Choset, "Central pattern generator with inertial feedback for stable locomotion and climbing in unstructured terrain," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 5769–5775.

[10] W. Chen, T. Liu, W. Li, J. Wang, X. Wu, and D. Liu, "Locomotion control with sensor-driven reflex for a hexapod robot walking on uneven terrain," *Transactions of the Institute of Measurement and Control*, vol. 38, no. 8, pp. 956–970, 2016.

[11] A. J. Ijspeert, A. Crespi, and J.-M. Cabelguen, "Simulation and robotics studies of salamander locomotion," *Neuroinformatics*, vol. 3, no. 3, pp. 171–195, 2005.

[12] W. Chen, G. Ren, J. Zhang, and J. Wang, "Smooth transition between different gaits of a hexapod robot via a central pattern generators algorithm," *Journal of Intelligent & Robotic Systems*, vol. 67, no. 3, pp. 255–270, Sep 2012.

[13] Y. Fukuoka, H. Kimura, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187–202, 2003.

[14] S. Aoi, P. Manoonpong, Y. Ambe, F. Matsuno, and F. Wörgötter, "Adaptive control strategies for interlimb coordination in legged robots: a review," *Frontiers in Neurorobotics*, vol. 11, p. 39, 2017.

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

[16] S. Gay, J. Santos-Victor, and A. J. Ijspeert, "Learning robot gait stability using neural networks as sensory feedback function for central pattern generators," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 194–201.

[17] 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.

[18] M. T. Nguyenová, P. Čížek, and J. Faigl, "Modeling proprioceptive sensing for locomotion control of hexapod crawling robot in robotic simulator," in *2018 Modelling and Simulation for Autonomous Systems (MESAS)*, 2019, pp. 215–225.

[19] K. Matsuoka, "Mechanisms of frequency and pattern control in the neural rhythm generators," *Biological Cybernetics*, vol. 56, no. 5, pp. 345–353, 1987.